RITA Intelligent Transportation Systems
Joint Program Office

Cooperative Intersection Collision Avoidance System Limited to Stop Sign and Traffic Signal Violations (CICAS-V)

Task 10 Final Report

Integration of Subsystems, Building of Prototype Vehicles and Outfitting Intersections

(Appendix G)

September 30, 2008

Crash Avoidance Metrics Partnership (CAMP) Produced
In conjunction with Virginia Tech Transportation Institute for
ITS Joint Program Office
Research and Innovative Technology Administration
U.S. Department of Transportation

CAMP Members:

Mercedes-Benz
General Motors (GM)
Toyota
Honda
Ford

Photos Credits
Photos and Illustration's courtesy of CAMP

Notice

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Cooperative Intersection Collision Avoidance System Limited to Stop Sign and Traffic Signal Violations (CICAS-V) Task 10 Final Report: Integration of Subsystems, Building of Prototype Vehicles, Outfitting Intersections		5. Report Date 6. Performing Organization Code
7. Author(s) Maile, M., Ahmed-Zaid, F., Basnyake, C., Caminiti, L., Kass, S., Losh, M., Lundberg, J., Masselink, D., McGlohon, E., Mudalige, P., Pall C., Peredo, M., Popovic, Z., Stinnett, J., and VanSickle, S.		8. Performing Organization Report No.
9. Performing Organization Name and Address Crash Avoidance Metrics Partnership on behalf of the Vehicle Safety Communications 2 Consortium 39255 Country Club Drive Suite B-40 Farmington Hills, MI 48331		10. Work Unit No. (TRAIS) 11. Contract or Grant No. DTFH61-01-X-00014
12. Sponsoring Agency Name and Address Federal Highway Administration 1200 New Jersey Avenue S.E. Washington, DC 20590		13. Type of Report and Period Covered Final Task Report May 1, 2006 – September 30, 2008 14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

The CICAS V project was a four-year project to develop a cooperative intersection collision avoidance system to assist drivers in avoiding crashes in the intersection by warning the driver of an impending violation of a traffic signal or a stop sign. The Vehicle Safety Communications 2 Consortium (VSC2) executed the project. Members of VSC2 are Ford Motor Company, General Motors Corporation, Honda R & D Americas, Inc., Mercedes-Benz Research and Development North America, Inc., and Toyota Motor Engineering & Manufacturing North America, Inc. Task 10 of the CICAS-V project contained the development of the final Phase I prototype that included Hardware and Software for both the Vehicle and Intersection part of the CICAS-V system. The Task also included the development of test procedures on the system and the component level to test and verify the correct functioning of all the components.

17. Key Word	18. Distribution Statement
DSRC, intersection crashes, traffic safety, traffic signal violation, stop sign violation, signal phase and timing, intersection map, positioning corrections, map matching, driver warning algorithm, message sets, intersection selection, local safety system, system architecture	No restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	412	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

List of Acronyms

AGID	Area Geometric Intersection Description or Area GID
API	Application Programming Interface
ATP	Acceptance Test Plan
CAN	Controller Area Network
CCH	Control Channel
CDIP	CICAS-V DSRC Information Process
CICAS-V	Cooperative Intersection Collision Avoidance System for Violations
CLOGP	CICAS-V Logging Process
ConOps	Concept of Operations
CVIP	CICAS-V Vehicle Information Process
CVLP	CICAS-V Vehicle Location Process
CWA	CICAS-V Warning Application
DAS	Data Acquisition System
DOP	Dilution of Precision
DOT	Department of Transportation
DSRC	Dedicated Short Range Communication
DVI	Driver-Vehicle Interface
DVIN	Driver Vehicle Interface Notifier
FHWA	Federal Highway Administration
FOT	Field Operational Test
GID	Geometric Intersection Description
GPS	Global Positioning System
GPSC	Global Positioning System Correction
HW	Hardware
ID	Identification or Identifier
IEEE	Institute of Electrical and Electronics Engineers
MCNU	Multiband Configurable Network Unit
NEMA	National Electrical Manufacturer's Association
NHTSA	National Highway Traffic Safety Administration
NMEA	National Marine Electronics Association
OBE	On-Board Equipment
OEM	Original Equipment Manufacturer
OTA	Over the Air
PC	Personal Computer
POC	Proof-of-Concept
PSC	Provider Service Content
PSID	Provider Service Identifier
RCMD	Remote Command

RTCM	Radio Technical Committee for Maritime Applications
RTK	Real-Time Kinematic
RSE	Road-Side Equipment
SCH	Service Channel
SPA	Service Provider Application
SPaT	Signal Phase and Timing
SW	Software
TMT	Technical Management Team
TOM	Transportation Object Message
TOMC	Transportation Object Message Compiler
TSVWG	Traffic Signal Violation Warning Given
UDP	User Datagram Protocol
USDOT	United States Department of Transportation
UTC	Coordinated Universal Time
VII	Vehicle-Infrastructure Integration
VSC2	Vehicle Safety Communications 2
VTTI	Virginia Tech Transportation Institute
WAAS	Wide Area Augmentation System
WAVE	Wireless Access in Vehicular Environment
WBSS	WAVE Basic Service Set
WSA	WAVE Service Advertisement
WSM	WAVE Short Message
WSU	Wireless Safety Unit
XML	Extensible Markup Language

Table of Contents

List of Figures

List of Tables

Executive Summary

This report presents the Task 10 – 'Integration of Subsystems, Building of Prototype Vehicles and Outfitting Intersections' Final Report for the Cooperative Intersection Collision Avoidance System Limited to Stop Sign and Traffic Signal Violations (CICAS-V) project.

The CICAS-V project's objective was to develop a cooperative intersection collision avoidance system to assist drivers in avoiding crashes in the intersection by warning the driver of an impending violation of a traffic signal or a stop sign. The Vehicle Safety Communications 2 Consortium (VSC2) conducted the project under Federal Highway Administration (FHWA) Cooperative Agreement No. DTFH61-01-X-00014, Work Order W-05-001. Members of VSC2 are Ford Motor Company, General Motors Corporation, Honda R & D Americas, Inc., Mercedes-Benz Research and Development North America, Inc., and Toyota Motor Engineering & Manufacturing North America, Inc.

The project was divided into two phases. Phase I featured the development and testing of a prototype CICAS-V system that was used for testing with naïve users. Phase II was to feature a full-scale Field Operational Test (FOT) of the system. While Phase I showed that the system was FOT-ready (i.e. the prototype was at a functional level that would support operation by naïve drivers) the U.S. Department of Transportation (USDOT) decided not to continue with the FOT in Phase II.

Task 10 involved the development, testing, and refinement of the CICAS-V system installed in both the vehicle and intersection for Phase I. The primary activities that took place as part of Task 10 were:

- Migrating the in-vehicle systems to the final FOT prototype Hardware and Software
- Migrating the intersections to the final FOT prototype Hardware and Software
- Verifying that all system components were operational, integrated, and working properly

Task 10 leveraged the work done in Task 3 – 'Human Factors', Task 8 – 'Phase I Prototype Build and Testing,' and Task 12 – 'Data Acquisition System' with an especially high reliance on the Task 8 material. It ran in parallel with those tasks since some of the necessary development work for the vehicle On-Board Equipment (OBE) and intersection Road-Side Equipment (RSE) had significant lead-time.

In-Vehicle FOT Ready Prototype Development

The result of the in-vehicle FOT ready prototype development was the final vehicle build used by each of the automotive Original Equipment Manufacturers (OEMs) for the component and system level testing run as a part of this task prior to executing the Task 11 – 'Vehicle and Intersection Objective Testing' activities. This involved the following:

- OBE Software (SW) Development and Test – The OBE SW development was based in large part on the prototype specifications developed as part of the

Task 8.7 activities. In Task 8, the details in these specifications were individually implemented and tested but not formally combined into a complete prototype. Task 10 consolidated the information in these and other documents by:

1. Developing an OBE Software Specification which detailed requirements for each of the software application modules and their interactions
2. Developing an OBE Software Design to describe the structure and design of the CICAS-V OBE software
3. Implementing and releasing the SW in several stages with the 'Final' release being used in the Task 11 objective testing
4. Developing an Acceptance Test Plan (ATP) and executing the tests contained in the plan

- In-Vehicle Hardware (HW) Integration – This involved identifying, acquiring, and integrating the necessary HW components into an integrated system which comprised the complete in-vehicle prototype FOT platform for the CICAS-V system. It was composed of both a set of CICAS-V hardware system components and a set of Data Acquisition System (DAS) hardware system components. The DAS hardware components are discussed in detail in the Task 12 Final Report: *Infrastructure and Vehicle DAS Functional Designs*. The Wireless Safety Unit (WSU) from DENSO was chosen as the primary component of the OBE hardware.
- Hardware and Over the Air (OTA) Interface Definition – For the CICAS-V application SW running on the WSU to communicate and work with the other in-vehicle HW components connected to the WSU, the content of the interfaces to these HW components was defined.

Intersection FOT Ready Prototype Development

The majority of the groundwork for the Task 10 intersection development work took place as part of the Task 8 activities. With this groundwork in place the Task 10 intersection, activities primarily focused on:

- Updating the RSE SW applications for compatibility with the Dedicated Short Range Communication (DSRC) standards in addition to making changes to support a change in the RSE HW platform. The planned hardware for the RSE was not going to be available, so a change was made to use the DENSO WSU
- Making the final HW selection along with installing the equipment in the remainder of the intersections
- Refining the OTA message formats to meet the final FOT CICAS-V system design

System Testing

The purpose of the Task 10 system testing effort was to verify that all of the system components were operational, integrated, and working properly through a combination of component level and intersection level system testing efforts. The execution of these tests was seen as a precursor to the execution of the Task 11 objective test procedures.

Test procedures were created to test not only the OBE as a whole but also the individual SW components that comprised the OBE. The component test case procedures were based on the SW specification, while the whole OBE test case procedures consisted of a set of intersection driving scenarios intended to verify that the OBE was operating as intended.

1 Introduction

1.1 Project Background

Each year about 5,000 fatal crashes occur in intersections with traffic signals or stop signs (National Highway Traffic Safety Administration, 2005 [1]. About 44% occur at traffic signals and 56% at stop signs. About 400,000 injury crashes occur at those intersections each year. About 600,000 property damage crashes also occur at those intersections annually. An initial analysis of relevant National Highway Traffic Safety Administration (NHTSA) crash databases shows that violation crashes have a variety of causal factors. These factors include driver distraction (a frequent factor cited by Campbell, Smith and Najm, 2004, p. 65 [2]); obstructed/limited visibility due to weather, intersection geometry or other vehicles; the presence of a new control device not previously known to the driver; and driver judgment errors.

The CICAS-V project was a four-year project to develop a cooperative intersection collision avoidance system to assist drivers in avoiding crashes in the intersection by warning the driver of an impending violation of a traffic signal or a stop sign. Cooperative means that the system involves both infrastructure and in-vehicle elements working together. Driver warnings, such as those planned for CICAS-V, may prevent many violation-related crashes by alerting the distracted or inattentive driver, thus increasing the likelihood that the driver will stop the vehicle and avoid violating the intersection.

The project was divided into two phases. Phase I featured the development and testing of a prototype of a CICAS-V system that is now ready for testing with naïve users. Phase II was to feature a full-scale FOT of the system. However, at the end of Phase I, the U.S. DOT decided not to continue with the FOT in Phase II even though the system was shown to be FOT-ready. Specific goals of CICAS-V include the establishment of:

- A warning system that will be effective at reducing the number of fatal crashes, the severity of injuries and property damage at CICAS-V intersections
- A warning system that is acceptable to users
- A vehicle-infrastructure cooperative system that helps vehicle drivers avoid crashes due to violations of a traffic signal or stop sign
- A system that is deployable throughout the United States

The basic concept of CICAS-V is illustrated at a high level in Figure 1 for a signalized intersection. In the figure, a CICAS-V equipped vehicle approaching a CICAS-V equipped intersection receives messages about the intersection geometry, status of the traffic signal and position correction information from the RSE. The driver is issued a warning if the OBE in the vehicle determines that, given current operating conditions, the driver is predicted to violate the signal in a manner which is likely to result in the vehicle entering the intersection. While the system may not prevent all crashes through such warnings, it is expected that, with an effective warning, the number of traffic control device violations will decrease, and result in a decrease in the number and severity of crashes at controlled intersections.

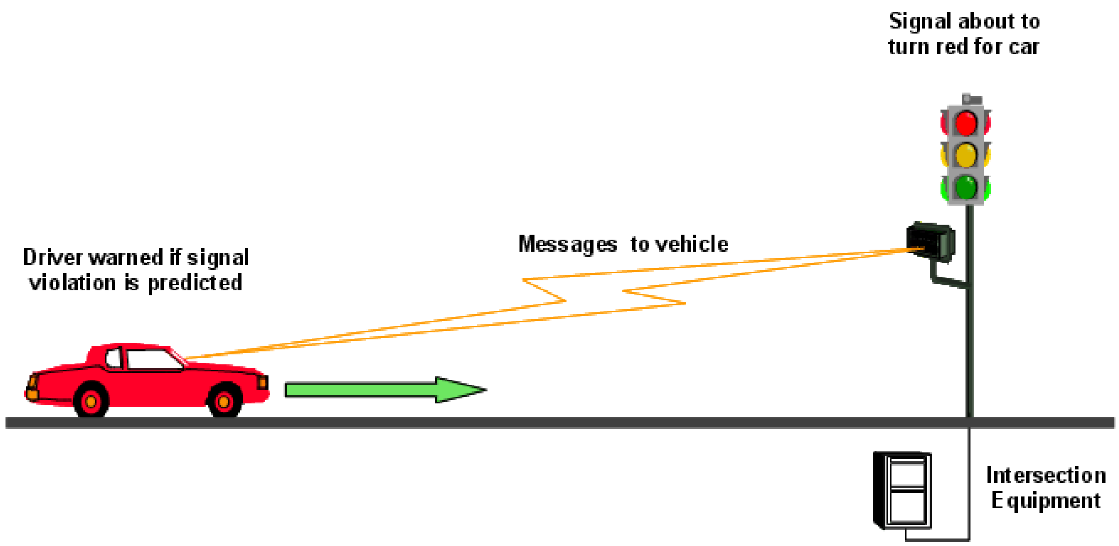

Figure 1: Basic Concept of the CICAS-V System at a Signalized Intersection

1.2 Task Overview

Task 10 – 'Integration of Subsystems, Building of Prototype Vehicles and Outfitting Intersections' involved the development, testing, and refinement of the CICAS-V system installed in both the vehicle and intersection. The result of this work was intended to be an FOT-ready system for a future FOT in Phase II of the project. This task leveraged the work done in Task 3 – 'Human Factors Research' [3], Task 8 – 'Phase I Prototype Build and Testing' [5], and Task 12 – 'Data Acquisition System' [7] with an especially high reliance on the Task 8 material. It ran in parallel with those tasks since some of the necessary development work for the OBE and RSE had significant lead-time.

The primary activities that took place as part of Task 10 were:

- Migrating the in-vehicle systems to the final FOT prototype HW and SW which involved:
    - The OBE SW development which required developing an integrated SW specification, design, implementation, and acceptance test plan
    - Definition of the physical HW architecture comprising the complete in-vehicle system along with the installation of this HW
    - Definition of HW and OTA interface message sets
- Migrating the intersections to the final FOT prototype HW and SW which involved:
    - The RSE SW development
    - Definition of the physical HW architecture comprising the complete intersection system along with the installation of this HW
    - Definition of HW and OTA interfaces

- Verifying that all system components were operational, integrated, and working properly through a combination of subsystem, system, and component level testing.

1.3 Report Organization

The remainder of this report is devoted to discussing the activities as outlined in Section 1.2 – 'Task Overview' above. Due to the size of this report and the detailed nature of much of its material the primary sections of the document will essentially be an overview of the task's activities with the specifics of these activities being discussed in the attached appendices. Section 2 discusses the in-vehicle FOT ready prototype development which included the OBE SW development and test, the in-vehicle HW integration, and the OBE HW and OTA interfaces. Section 3 discusses the Intersection RSE equivalent of the Section 2 material. Section 4 discusses the component and system level testing activities that took place by the CICAS-V Technical Management Team (TMT) as part of Task 10.

2 In-Vehicle FOT Ready Prototype Development

After the system tests in Task 8 showed that the system design was realistic and sound, the vehicles identified and purchased as a part of the Task 8 activities were upgraded to the final FOT-ready hardware and software (Prototype functioning at a level suitable to be operated by naïve drivers on actual roads). The result was the final vehicle build used by each of the OEMs for the component and system level testing run as a part of this task prior to executing the Task 11 – ‘Vehicle and Intersection Objective Testing’ [6] activities. This involved developing and testing the consolidated CICAS-V OBE SW, selection and installation of the required in-vehicle HW, and defining the necessary HW and SW interfaces to allow the system to work as intended.

2.1 OBE Software Development and Test

The OBE SW development was based, in large part, on the prototype specifications developed as part of the Task 8.7 activities which included the following:

- Map Matching
- CICAS-V Global Positioning System (GPS) Corrections (GPSC) Generation and Vehicle Positioning System
- CICAS-V Warning Algorithm
- CICAS-V Message Sets

The details in these specifications were individually implemented and tested in Task 8 but not formally combined into a complete prototype. They are documented in the Task 8 Final Report. The Task 10 activities consolidated the information in these and other documents into a concise CICAS-V OBE specification, design, software implementation, and software acceptance test plan.

2.1.1 OBE Software Specification

Using the Task 8.7 specifications as the primary input, the initial step of the Task 10 SW specification process involved the development of a CICAS-V OBE Software Architecture (Figure 2). The architecture developed consisted of two sets of modules:

1. CICAS-V Application Modules – Modules specific to the CICAS-V OBE SW Application
2. WSU Software Services Modules – Generic modules supplied with the DENSO WSU HW that provided services and an Application Programming Interface (API) to enable applications to interface to the Controller Area Network (CAN) buses, GPS receiver, and the Wireless Access in Vehicular Environments (WAVE) Radio

The scope of the SW specification was limited to the CICAS-V Application modules only.

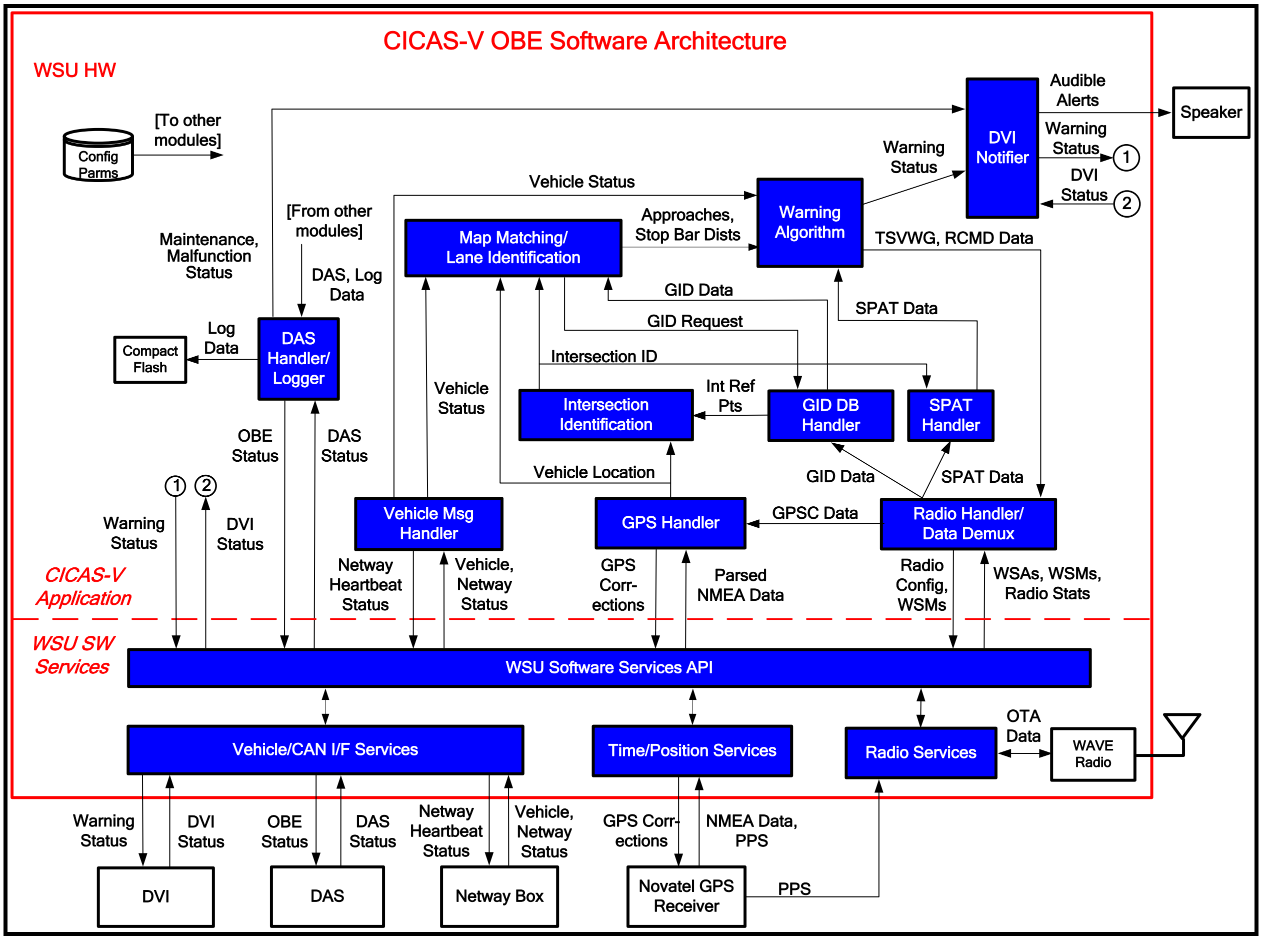

Figure 2: CICAS-V OBE Software Architecture

The CICAS-V Application modules were grouped and divided into two categories:

1. Interface/Message Handling Modules – Interface to external devices and/or perform message handling and parsing functions
2. Violation Detection Modules – Process the latest vehicle, GPS, Geometric Intersection Description (GID), and Signal Phase and Timing (SPaT) data to determine whether an intersection violation is likely to occur

The modules assigned to each sub-category are listed in Table 1 below along with a brief description of the module's intent followed by the module grouping in Figure 3.

Table 1: CICAS-V OBE Application Module Summary

Module	Description
Interface/Message Handling Modules	
Vehicle Message Handler	• Interfaced to the Netway device (through the WSU Vehicle/CAN Interface Services) to receive generic CAN messages with vehicle status • Transmitted and received heartbeat status information with the Netway

Module	Description
Radio Handler/Data Demux	• Interfaced to the WAVE Radio (through the WSU Radio Services) • Configured the radio, and polled the radio driver for statistics • Transmitted and received WAVE Short Messages (WSMs) • Processed received WAVE Service Advertisement (WSA) indications
GPS Handler	• Interfaced to the NovAtel® OEMV® GPS receiver (through the WSU Time/Position Services) • Output GPSC data • Input GPS time and position data
GID Database Handler (Vehicle Based)	• Maintained the GID database • Upon receipt of GID data, added a record to the database, or updated an existing record if the data was of a different version • Deleted expired GID records • Performed WAVE Basic Service Set (WBSS) selection if the GID or GPSC data was being broadcast on the Service Channel
SPaT Handler	• Received and parsed the SPaT data • Converted the data to a format usable by other modules
DAS Handler/Logger	• Interfaced to the DAS (through the WSU Vehicle/CAN Interface Services) • Output OBE status and input DAS status • Performed hardware/software watchdog processing and determined whether a maintenance or malfunction condition exists
Violation Detection Modules	
Intersection Identification	• Identified the intersection the vehicle was approaching based on the vehicle location and direction and the GID intersection reference points
Map Matching/Lane Identification	• Calculated the most likely lane(s) and approach(es) of the vehicle, and the distance to the stop bar(s) based on the vehicle location and GID data
Warning Algorithm	• Determined if an intersection violation was likely to occur • Generated Traffic Signal Violation Warning Given (TSVWG) and Remote Command (RCMD) messages to be transmitted to the RSE
DVI Notifier	• Interfaced to the Driver-Vehicle Interface (DVI) (through the WSU Vehicle/CAN Interface Services) • Controlled the DVI icon and flexible warning outputs • Transmitted and received heartbeat status information with the visual DVI device • Generated audible DVI alerts

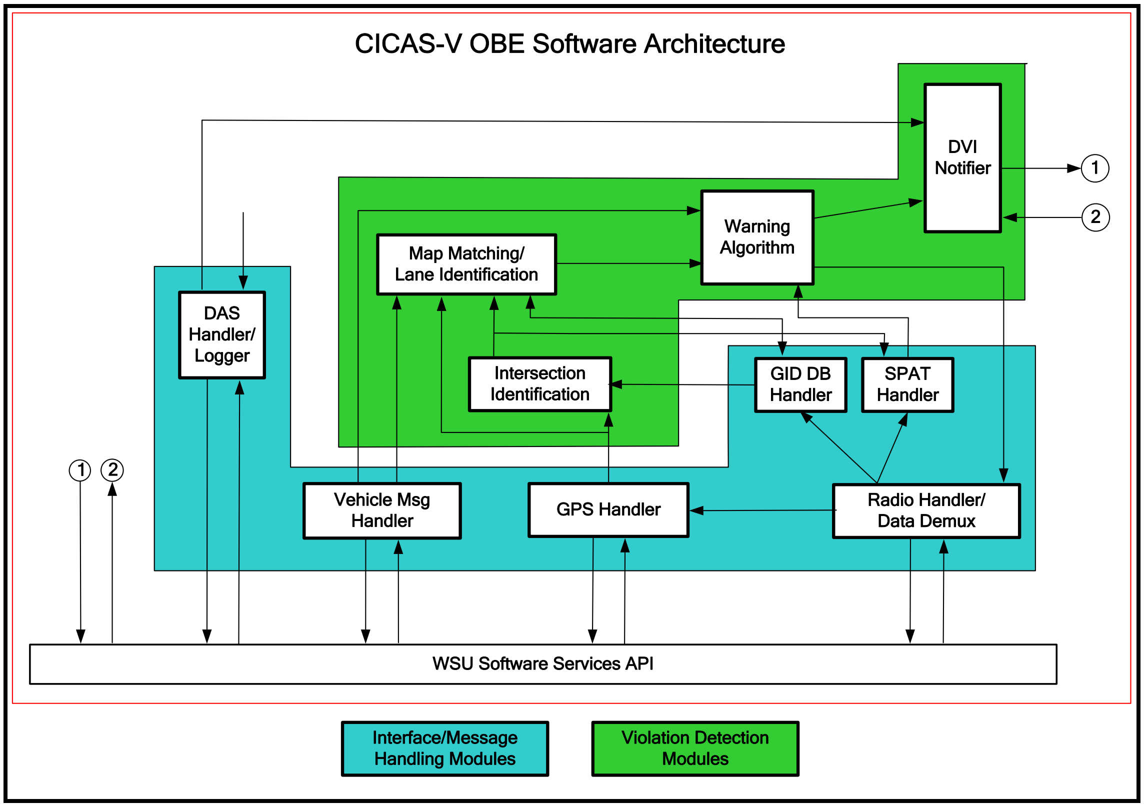

Figure 3: CICAS-V OBE Software Module Categories

Following the Application Module identification and grouping, detailed SW Specifications for each of the modules and their interaction were developed. The specifications included the following information for each module:

- Module Requirements
- Inputs to the Module
- Processing Logic for the Module
- Outputs from the Module
- Configuration parameters for each Module

0 contains the specification details for each of the modules in the order in which they are listed in Table 1. This order is based on the primary data flows, and, is intended to present the information in a logical sequence.

2.1.2 OBE Software Design

The Software Design describes the structure and design of the CICAS-V OBE SW. The design was intended to satisfy the requirements of the CICAS-V OBE Software Specification discussed in Section 2.1.1 and detailed in 0. Like the SW specification, the scope of the SW design was limited to the CICAS-V application modules as listed in Table 1 above.

The application modules were grouped into six processes for implementation in addition to a CICAS-V main process. These processes are listed in Table 2 below along with a brief description of each process. Following the process description table is Figure 4, which illustrates the process breakdown and the assignment of the application modules (see Table 1) to each of the processes.

Table 2: CICAS-V OBE Process Description

Process	Description
CICAS-V Main	• Parsed the configuration file • Started up the other processes
CICAS-V DSRC Information Process (CDIP)	• Interfaced to the WAVE Radio • Configured the radio, and polled the radio driver for statistics • Processed received WSA indications and WSMs • Transmitted WSMs
CICAS-V Vehicle Location Process (CVLP)	• Interfaced to the NovAtel OEMV GPS receiver • Output GPSC data • Input GPS time and position data
CICAS-V Vehicle Information Process (CVIP)	• Interfaced to the Netway box to receive generic CAN messages with vehicle status data • Transmitted and Received Netway heartbeat status
CICAS-V Warning Application (CWA)	• Processed the latest vehicle status, vehicle location, GID, and SPaT data to determine whether an intersection violation was likely to occur
Driver Vehicle Interface Notifier (DVIN)	• Interfaced to the DVI to control the DVI icon • Controlled the flexible warning outputs • Generated audible DVI alerts • Transmitted and Received DVI icon heartbeat status
CICAS-V Logging Process (CLOGP)	• Interfaced to the DAS • Output OBE status and input DAS status • Performed HW/SW watchdog processing and determined whether a maintenance or malfunction condition existed

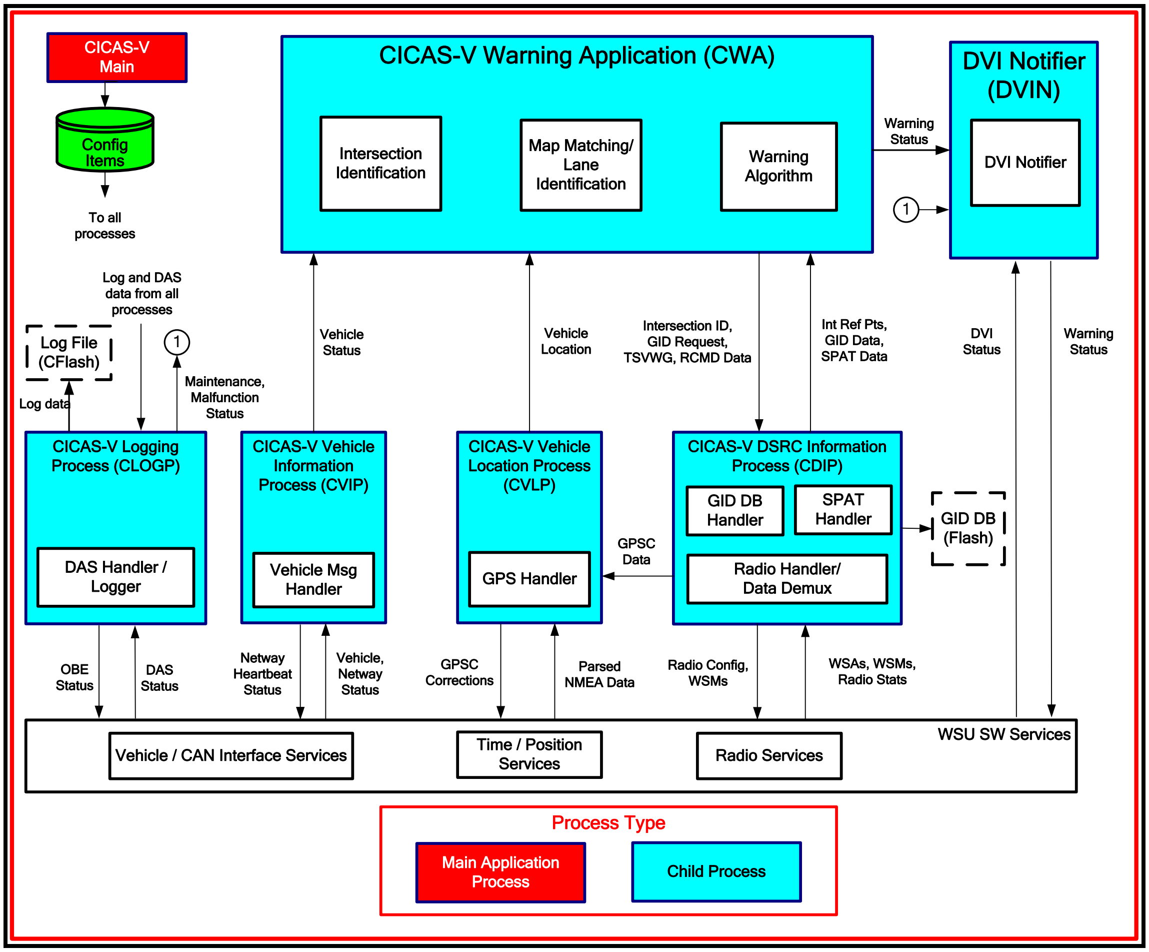

Figure 4: CICAS OBE Software Design

Following the process identification and corresponding application grouping, the SW Design details for each process and its applications modules were developed. The design included the following information for each process:

- Process Overview
- Interface to the Process
- Process Structure
- Process Thread Functional Description(s)

0 contains the design details for each of the processes in the order in which they are listed in Table 2, which is intended to present the information in a logical sequence.

2.1.3 Software Implementation and Release

The vehicle component of the software development in Task 10 was accomplished in several stages where increased functionality was added and tested in a tight feedback loop. The following SW release types were identified for the program:

- Interface – Supported the external interface requirements to the OBE. This release of the software allowed issues with the interfaces to be worked out prior to receiving application level software.
- Alpha – Supported the primary functionality of the high priority application components of the CICAS-V system as identified by the Task 8.7 activities. This release of the software was suitable for engineering testing of the main components of the CICAS-V system such as lane-level positioning, map matching, GPS position correction processing, and initial warning algorithm functionality.
- Baseline – Supported the full functionality of the high priority application components of the CICAS-V system as identified by the Task 8.7 activities. This release of the software incorporated the error handling paths of the software that was not present in the Alpha release.
- Pilot FOT – Supported refinements to the components in the Baseline release based on further development and testing by the OEMs. In addition this release included the remaining lower priority modules as identified by the Task 8.7 activities. This release of the software was used for the Pilot FOT testing with naïve drivers.
- Final – Supported refinements to the components in the Pilot FOT release based on the Task 10 System and Pilot FOT testing efforts. This release of the software was used for the Task 11 objective testing effort.

Each of the release types identified above had multiple releases packages with each package containing a WSU Software Services release and a CICAS-V Application release. The table below shows the final release revision and timing for each of the release types.

Table 3: CICAS-V OBE SW Release Details

Release Type	Release Date	Release Revision		
		OBE Package	WSU SW Services	CICAS-V Application
Interface	September 20, 2007	1.2	1.2	N/A
Alpha	November 19, 2007	1.4	1.4	1.0
Baseline	December 21, 2007	1.4.1	1.4	1.1
Pilot FOT	May 15, 2008	1.11	1.11	1.5
Final	July 9, 2008	1.15	1.15	2.0

2.1.4 Acceptance Test Plan and Results

The Acceptance Test Plan defined the tests to execute to verify compliance of the SW implementation with the requirements detailed in the CICAS-V Software Specification. For each test or series of tests the following were identified:

- The requirement(s) under test
- The procedural steps for executing the test
- The required WSU configuration
- The expected and actual test results (see Appendix C for the results)

Note that the tests only validated the CICAS-V Application and did not include validating the operation of WSU Software Services which were not developed as a part of this program.

2.1.4.1 Test Environment & Setup

The ATP was run in the laboratory under controlled conditions with a simulated test environment:

- Vehicle motion was simulated by creating GPS Sequence data using a GPS Generator tool
- A GPS Simulator running on a Linux Personal Computer (PC) transferred GPS National Marine Electronics Association (NMEA) data to the WSU via a serial interface
- SPaT information to be transmitted by a RSE was simulated by creating SPaT Message Sequence data using a SPaT Generator tool
- The SPaT Message Sequence data, as well as GID and GPSC data, was supplied to a RSE Simulator
- The RSE Simulator ran on a Linux PC and transmitted WSMs containing the GPSC, GID, and SPaT messages

The laboratory test setup used for the ATP is illustrated in Figure 5. An alternate setup used for some tests consisted of playing back a recording using a Scenario Replicator tool which was developed as a part of another program.

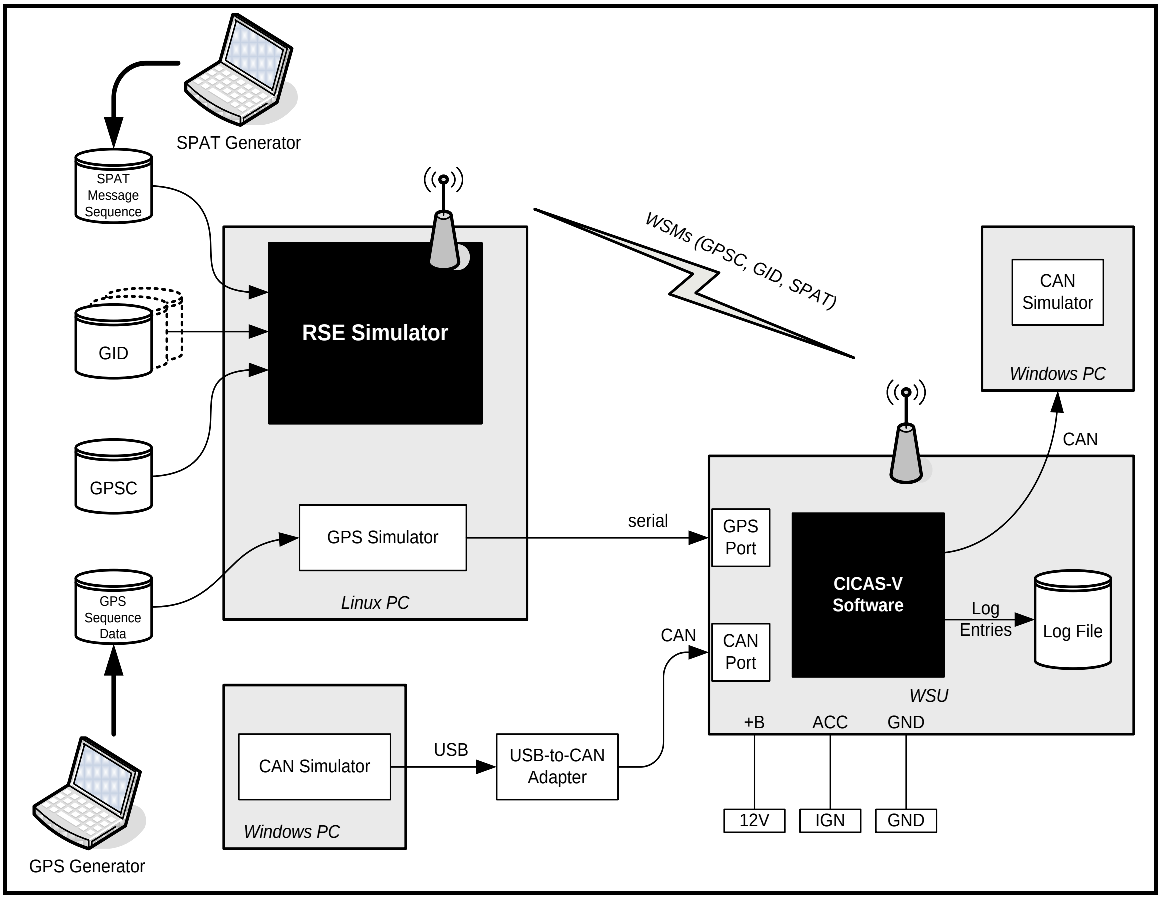

Figure 5: CICAS-V OBE ATP Test Setup

2.1.4.2 Approach

A step-wise approach was used to verify the CICAS-V implementation running on the WSU. The basic building blocks were verified first followed by the modules that required the services of these blocks with expectation that the previously verified blocks continued to operate correctly. The step-wise approach taken is as follows:

1. Verified interfaces operated properly and that interface errors were correctly detected and reported.
2. Verified GPSC data was received and output to the GPS receiver for processing.
3. Verified GID data was received and processed when appropriate, and that the most up-to-date and relevant GID data was stored in the WSU.
4. Verified the WSU correctly used the GPS information and the stored GID information to accurately identify which, if any, CICAS-V equipped intersection the vehicle would pass through next.
5. Verified the WSU correctly used the GPS information and the GID information to accurately identify which, if any, lane of travel was currently being used by the vehicle.
6. Verified the WSU correctly used the GPS information and the GID information to accurately identify which, if any, Intersection Approach applied to the vehicle’s lane of travel.

7. Verified SPaT data was received and processed when appropriate, and that the SPaT data was applied to the appropriate intersection.
8. Verified the WSU correctly used the GPS information, the GID information, and the SPaT information to accurately determine if a warning should be sent to the vehicle driver.
9. Verified that all detected errors were correctly reported to the Error Handler and that the WSU provided the correct warning and error notifications to the DVI.
10. Verified the hardware and software watchdog timers monitored the OBE health and correctly reported and recovered from detected errors.

2.1.4.3 Tests

The ATP grouped the tests according to the application modules listed in Table 1 and thus contains the following test groupings:

- Interface Tests
- Vehicle Message Handler Tests
- Radio Handler / Data Demux Tests
- GPS Handler Tests
- GID Database Handler Tests
- SPaT Handler Tests
- DAS Handler / Logger Tests
- Violation Detection Module Tests: Intersection Identification, Map Matching / Lane Identification, Warning Algorithm / State Machine Tests
- DVI Notifier Tests
- Error Handler Tests

The requirements under test, execution steps, configuration, expected results and actual results for tests corresponding to each of the test groupings listed above can be found in 0.

2.1.4.4 Test Summary

The following table provides a summary of the ATP testing results followed by a discussion on how to interpret the table. In summary, ninety-nine (99) requirements test cases were run with:

- 84 requirements test cases having no issues (approximately 85%)
- 8 requirements test cases having issues (approximately 8%)
- 3 requirements test cases needing to be re-run (approximately 3%)
- 4 requirements test cases not run (approximately 4%)

Table 4: Task 10 ATP Test Result Summary

Requirements Test Procedure Catagories	Requirements Test Status Catagories							
	Identified	Obsolete	Duplicate	To be Tested	No Issues	Issues	Re-run	Not Tested
Vehicle Message Handler	13	0	0	13	11	1	0	1
Radio Handler / Data Demux	8	1	2	5	4	0	0	1
GPS Handler	10	0	2	8	7	0	1	0
GID Database Handler	13	0	2	11	10	1	0	0
SPaT Handler	6	0	2	4	4	0	0	0
DAS Handler / Logger	12	1	1	10	9	1	0	0
Intersection Identification	8	0	2	6	4	2	0	0
Map Matching / Lane Identification	11	0	1	10	8	2	0	0
Warning Algorithm	18	1	1	16	13	1	2	0
DVI Notifier	12	0	3	9	7	0	0	2
Error Handler	9	2	0	7	7	0	0	0
	120	5	16	99	84	8	3	4

Where:

Colour	Meaning
Grey	Test cases obsoleted or duplicated
Red	Test cases with issues not fixed
Yellow	Test cases to be re-run or not run

The following material describes how the 'Requirements Test Status Categories' columns in Table 4 should be read:

- Identified – This column represents the number of requirements identified for test.
- Obsolete – As part of the SW release, test, and feedback loop a number of requirements were no longer valid and thus obsolete. These requirements were marked as 'Obsolete' rather than re-numbering / using the requirement number.
- Duplicate – A number of high level general requirements were duplicates with lower level detailed requirements. For these instances, the detailed requirements test cases were primarily run.
- To be Tested – This was the final count of requirements tests cases to be run after subtracting the 'Obsolete' and 'Duplicate' requirements from the 'Identified' ones.
- No Issues – Of the 'To be Tested' test cases, this is the number of test cases that met the expected results the first time the test case was run.
- Issues – Of the 'To be Tested' test cases, this is the number of test cases that did not meet the expected results and were not fixed. If a Phase II FOT is planned, these issues will need to be evaluated to determine if they should be fixed. Those issues were minor and were found not to impact the functionality or FOT readiness of the system, but may be beneficial to fix in an FOT release.

NOTE: The vast majority of the test cases had multiple actions and items that needed to be confirmed as part of the test case. If any of these actions or items did not meet the expected results, the requirement test case as a whole was marked as having an issue. The majority of these issues had to do with the improper logging of data and did not affect the primary operation of the OBE. Thus, these were given a lower priority for implementing a fix.

- Re-run – Of the 'To be Tested' test cases, this is the number of test cases that, after post analysis of the test results, need be re-run to confirm the actual results with the expected results or re-evaluated for the proper behavior if a Phase II FOT is planned.
- Not Tested – Of the 'To be Tested' test cases, this is the number of test cases that were not run. These were primarily lower priority error handling (deferred, pending a Phase II FOT) and timing conditional tests.

A summary of the ATP test results can be found at the end of 0.

2.2 In-Vehicle Hardware Integration

The in-vehicle HW system integration activities involved identifying, acquiring, and integrating the necessary HW components into an integrated system used in the Task 10 testing activities in preparation for Task 11 objective testing. This HW, as installed in the vehicles purchased by each of the VSC2 Consortium member OEMs and Virginia Tech Transportation Institute (VTTI), comprised the complete in-vehicle prototype FOT platform for the CICAS-V system. It was composed of both a set of CICAS-V hardware system components and a set of DAS hardware system components as detailed in Figure 6 below. Only the CICAS-V hardware components will be discussed further. The DAS hardware components are discussed in detail in the Task 12 Final Report.

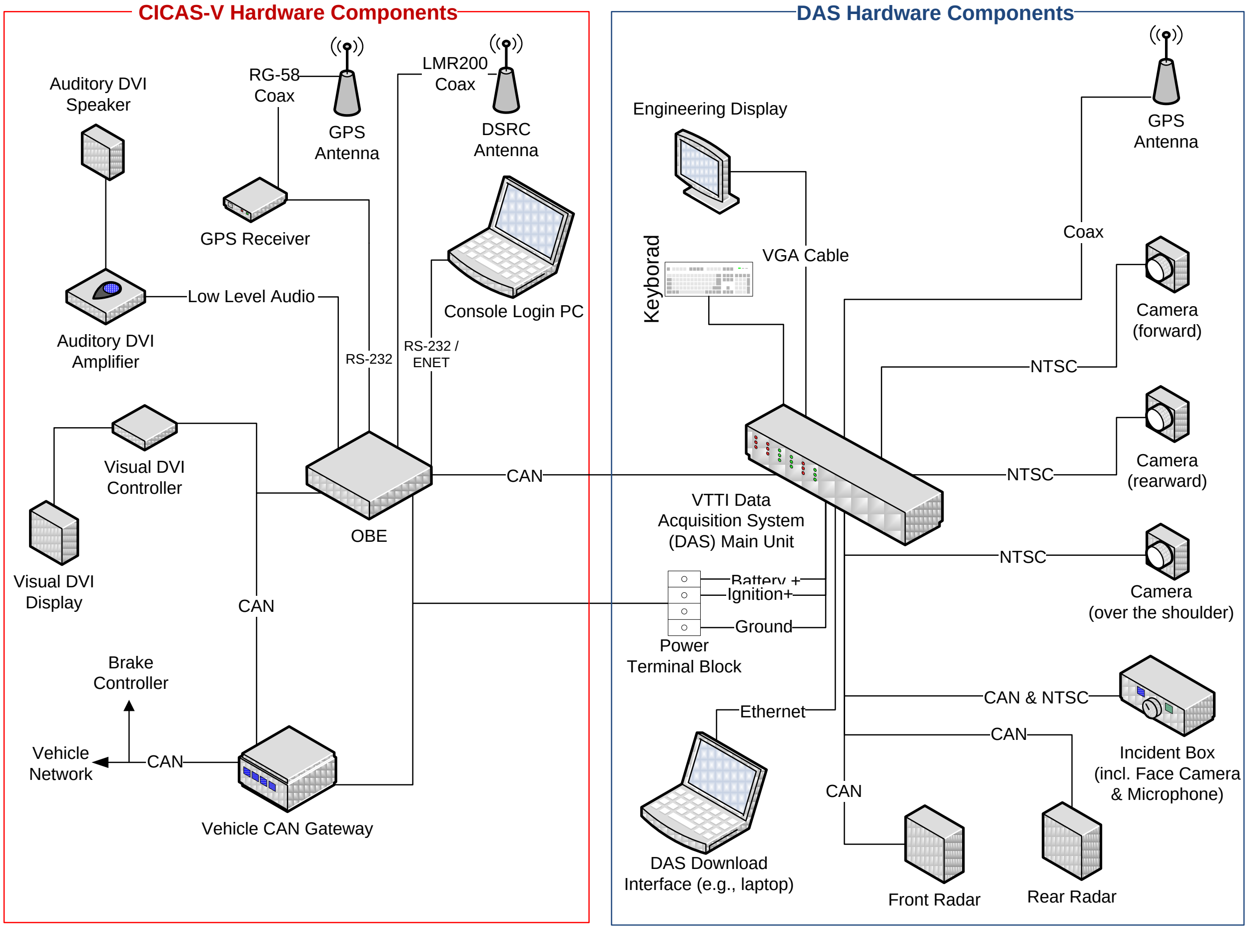

Figure 6: In-Vehicle Prototype FOT HW Platform

The central HW component of the vehicle system was the DENSO WSU on which the CICAS-V SW implementation executed. It included a single board embedded computing device utilizing the Freescale MPC5200B PowerPC and extended automotive and IT peripherals along with DSRC radio support. Other than for a few exceptions which are not noted in this report, when operating in WAVE mode the WSU provides the functionality specified by IEEE P1609.3 [10] [11], IEEE P1609.4 [12], and IEEE 802.11p [13]. Figure 7 details the physical HW interfaces supported by the WSU followed by a WSU HW and SW feature summary listed in Table 5.

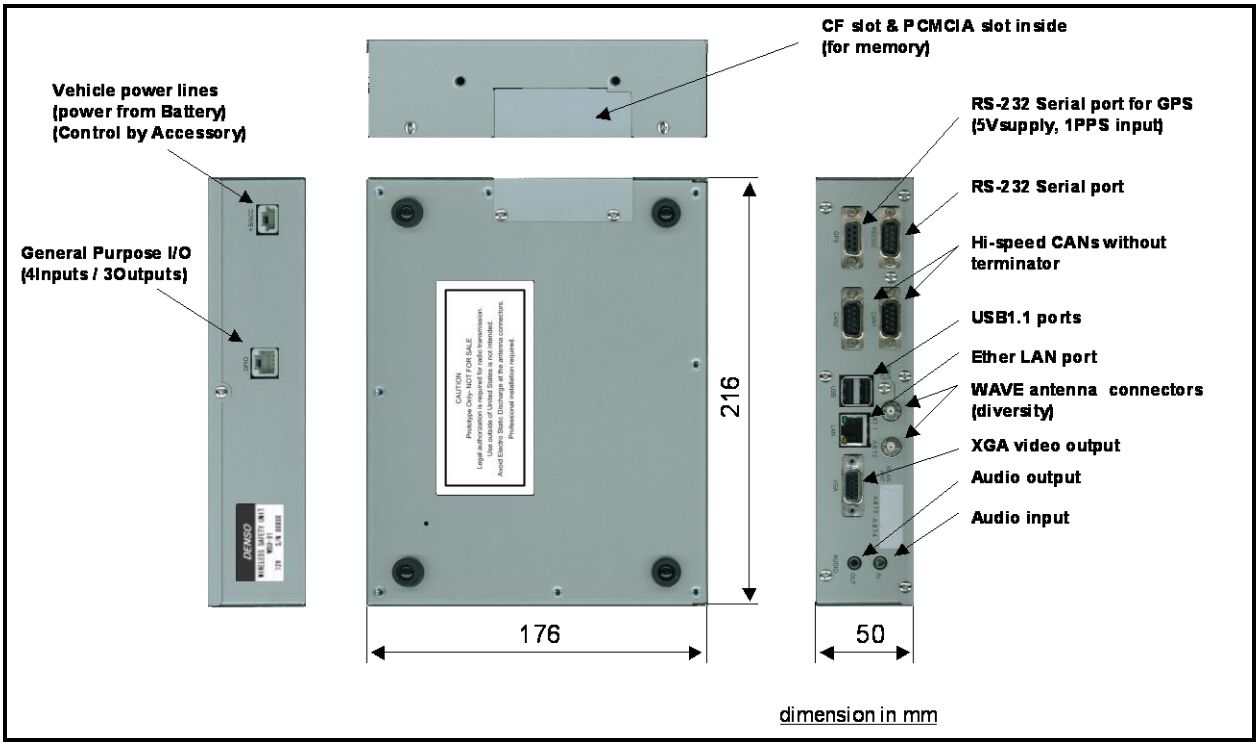

Figure 7: WSU External Physical Interfaces

Table 5: WSU HW and SW Feature Summary List

Linux OS	ANSI C/C++	Application APIs
128MB DDR SDRAM	64MB Onboard Flash ROM	1024 x 768 x 16 Video Res
Antenna Diversity	400MHz MPC5200B PowerPC	Automotive Type Temp.
Automotive Shock & Vibration		

Figure 8 below shows which of the WSU HW interfaces, detailed in Figure 7 above, were used by the CICAS-V application along with the external HW components that were connected to these interfaces. The tables that follow go into some of the specifics of these external HW components.

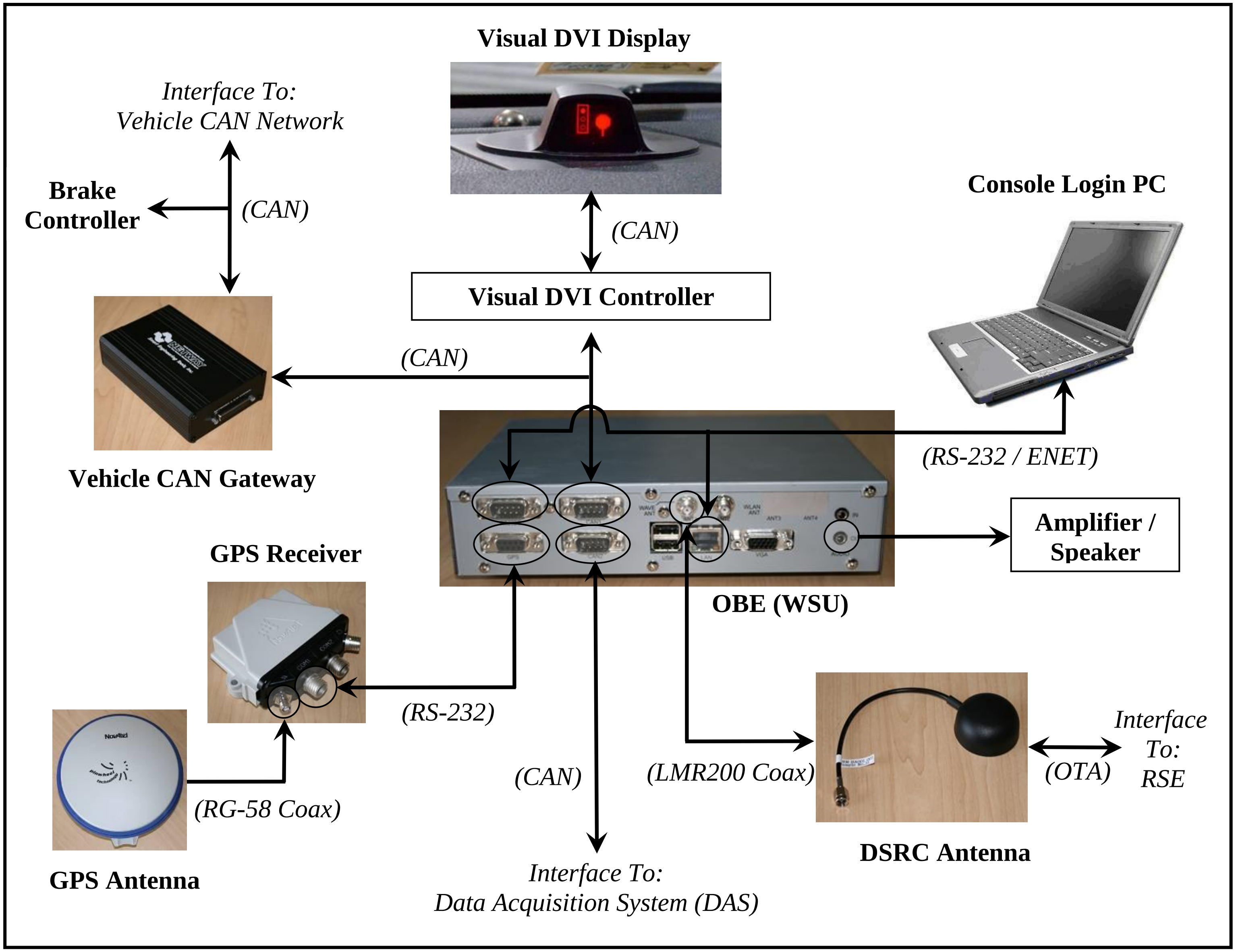

Figure 8: CICAS-V OBE HW Interfaces and Components

Table 6: CICAS-V OBE HW Interface Component Details

HW Component	Manufacturer	Model
OBE	DENSO	WSU-01(A)
DSRC Antenna	Nippon Antenna	DEN-HA001-001
GPS Receiver	NovAtel®	OEMV® Flexpak V1-RT20
GPS Antenna	NovAtel®	GPS-701-GG
Vehicle CAN Gateway	Smart Engineering Tools	Netway 6
Visual DVI Controller	VTTI	Custom for Program
Visual DVI Display	VTTI	Custom for Program
Console Login PC	Various	Various
Amplifier	Cana Kit	UK153
Speaker	Jetstream	JTSP6

Table 7: CICAS-V OBE HW Interface Cable Details

HW Component	Manufacturer	Model
LMR200 Coax Cable	Talley	CXTK20K-xx (SMA Male to SMA Male connectors where xx = length in feet)
RS-232 Cable	Deutsch	17211-1204-078 (Deutsch Circular to DB9 Male connectors)
RG-58 Coax Cable	NavTech GPS	RG58-xx-TM-TM (TNC Male to TNC Male connectors where xx = length in feet)

Table 8: CICAS-V Test Vehicles

Automotive OEM / VTTI	Make	Model	Year
Ford	Volvo	S80	2007
General Motors	Cadillac	STS Sedan	2005
Honda	Acura	RL	2006
Mercedes-Benz	Mercedes	ML-350	2006
Toyota	Toyota	Prius	2006
VTTI	Cadillac	STS Sedan	2006

2.3 Hardware and Over the Air Interfaces

Seven interfaces were defined in this task to allow the CICAS-V application SW on the WSU to work with the in-vehicle HW components identified in Table 6. The interfaces are listed below:

- OBE – Vehicle CAN Gateway CAN Interface
- OBE – GPS Receiver RS232 Interface
- OBE – DAS CAN Interface
- OBE – Visual DVI Controller CAN Interface
- OBE – Audio DVI AC97 Standard Interface
- OBE – Brake Controller DVI CAN Interface
- OBE – RSE DSRC OTA Interface

A brief description of these interfaces and their content is discussed in the following sections and, where appropriate, referenced appendices. Some of the interface content will not be provided due to it being proprietary to the individual OEMs or already readily available in the public domain.

2.3.1 OBE – Vehicle CAN Gateway CAN Interface

This interface existed between the WSU running the CICAS-V SW application and the Netway 6. The Netway 6 provided a consistent interface to the vehicle for the CICAS-V SW application for each of the vehicles listed in Table 8. It monitored vehicle data messages originating from the vehicle systems (e.g., power train control module, braking modules, etc.) on one or more multiplexed data busses. When the Netway 6 detected specific data items of interest to the CICAS-V application, it translated them into a generic representation for use by the application. The use of the Netway 6 simplified the software development by allowing a single CICAS-V application to be developed that worked on all of the program vehicles while allowing the OEM vehicle bus content to remain proprietary.

Four types of general interface messages were defined for exchange between the WSU and the Netway 6 (Note: Here and in the rest of the document the '$' symbol does not represent a monetary value):

1. Vehicle information messages sent from the Netway 6 to the WSU (Messages $600 to $605). The majority of the data items defined in these messages were used for logging purposes by the CICAS-V application as well as for re-transmission to the DAS to aid in testing and data analysis. The core CICAS-V application only required three of the data items to aid in determining if a violation was potentially going to occur. These items were:
    1. Brakes Active (Message $600)
    2. Driver Intended Braking Level (Message $600) – the level at which the driver was applying the brakes
    3. Vehicle Speed (Message $601)

2. A vehicle information availability message sent from the Netway 6 to the WSU which indicated the availability of each of the data items contained in CAN messages $600 to $605 (Message $650)
3. A raw warning information message along with configurable warning flags sent from the WSU to the Netway 6 to be used for the brake pulse DVI indication (Message $703)
4. Heartbeat information messages sent between the WSU and Netway 6 to detect link failures between the two devices (Messages $606 and $704)

The data content definition for this interface is defined in 0.

2.3.2 OBE – GPS Receiver RS232 Interface

This interface existed between the WSU running the CICAS-V SW application and OEMV GPS receiver on the vehicle. Using this interface, sub-meter “Which-Lane” position accuracy, which was required by the CICAS-V SW application, was achievable when locally-produced GPS corrections from an RSE were utilized. This made it possible for the vehicle to use Real-Time Kinematics (RTK) to establish its position relative to the intersection with an accuracy of better than 0.5 m. When local corrections were not available, the GPS receiver was configured to fall back to Wide Area Augmentation System (WAAS) differential correction mode, if possible, which would allow the CICAS-V application to continue to function at a sub-set of intersections requiring only “Which-Road” positioning accuracy. Where, from the Concept of Operations, “Which-Lane” and “Which-Road” refer to:

> Which-Lane: Accuracy level for the vehicle location that enables the vehicle to determine on which lane on which road it is approaching an intersection.
>
> Which-Road: Accuracy level for the vehicle location that enables the vehicle to determine on which road it is approaching an intersection.

Two sets of interface messages were defined for exchange between the WSU and the OEMV:

1. Radio Technical Commission for Maritime Applications (RTCM) v3.0 messages sent from the OBE to the GPS Receiver:
    - RTCM1005 message which contained the coordinates of the antenna reference point for the RSE in addition to other additional minor attributes
    - RTCM1001 message which contained the RSE satellite observations, in particular the single-frequency (L1) corrected pseudo-range and phase-range measurements for each satellite (i.e., based on both basic time of arrival and carrier phase analysis), so vehicle receivers could correct their own local position estimates
2. NMEA messages sent from the GPS Receiver to the OBE:
    - $GPGGA was used to provide the basic position (latitude, longitude, elevation) estimation, GPS fix quality, and the number of satellites used to make the position estimate in addition to providing the GPS-based Coordinated Universal Time (UTC) time
    - $GPRMC was used to provide the "speed over ground" (GPS speed) and "track made good" (GPS heading angle)

- $GPGST was used to provide a position uncertainty / error ellipse at the ground plane in terms of the standard deviation of latitude error and longitude error both in meters
- $GPGSA was used to provide the Dilution of Precision (DOP) expected accuracy factors

Both the RTCM and NMEA messages are publicly available standard messages and, thus, the content definition will not be included as a part of this document.

2.3.3 OBE – DAS CAN Interface

This interface existed between the WSU running the CICAS-V SW application and the DAS. The vehicle DAS recorded data supplied by the WSU and other HW devices (e.g., radar, cameras). For detailed information on the DAS please refer to the Task 12 Final Report. The data sent to the DAS from the WSU consisted of the following general types of interface messages:

1. Vehicle information messages received by the WSU from the Netway 6 and re-transmitted from the WSU to the DAS (Messages $600 to $605)
2. A vehicle information availability message received by the WSU from the Netway 6 and re-transmitted from the WSU to the DAS (Message $650)
3. CICAS-V SW application information messages (Messages $610 to $619) which included data related to the:
    - CICAS-V Warning Application (Messages $610 to $612 and $618)
    - GPS position and status (Messages $614 to $617 and $618)
    - Status of the required conditions for intersection identification processing (Message $619)
    - DVI visual icon state (Message $619)
    - Reception of WSA, SPaT, GID, and GPSC OTA messages (Message $618)
    - OTA message transmission / reception latency between the RSE and OBE (Message $751)
4. Heartbeat and other health status information messages sent between the WSU and DAS to detect link failures between the two devices as well as notify each device of errors internal to the other device (Message $606 and $701)

The data content definition for this interface is defined in 0.

2.3.4 OBE – DVI Interfaces

One of the goals of Task 3 was to issue a recommendation for the DVI to be used for the Pilot FOT phase of the CICAS-V system. The recommendation was that a visual indication, speech announcement, and brake pulse activation be included as part of the DVI warning approach for CICAS-V. Each of these DVI indications was controlled via an interface between the WSU and other HW components. Following is a brief discussion of each DVI indication along with details on the interface content where possible. For additional information on the following DVI indications, please refer to the Task 3 Final Report.

- Visual DVI Controller CAN Interface

This interface existed between the WSU running the CICAS-V SW application and the Visual DVI CAN Interface device as depicted in Figure 8. The device the WSU interfaced with was developed internally by VTTI and resided on the same CAN bus as the Netway 6 device. The visual DVI indication had three states:

1. An inactive state when the vehicle was not approaching a CICAS-V intersection
2. A blue visual indication when the vehicle was approaching a CICAS-V intersection
3. A red visual indication when a warning was provided to the driver at a CICAS-V intersection

Two types of interface messages were defined for exchange between the WSU and the Visual DVI CAN Interface device:

1. A message to control the visual aspects of the indication such as color, brightness, and flash frequency (Message $700)
2. Heartbeat message information sent between the WSU and Visual DVI CAN Interface to detect link failures between the two devices (Message $700 and $702)

The data content definition for this interface is defined in 0.

- Audio DVI AC97 Standard Interface

This interface existed between the WSU running the CICAS-V SW application and the Amplifier / Speaker combination. The speech DVI indication consisted of a female voice stating either “Stop Light” or “Stop Sign” when a warning was provided to the driver and was played out of the 3.5mm AUDIO OUT jack of the WSU.

- Brake Controller DVI CAN Interface

This interface existed between the WSU running the CICAS-V SW application and vehicle brake controller with the Netway 6 acting as a gateway. The brake controller side of the interface is proprietary and, thus, will not be documented in this report. When the Netway 6 received the $703 message from the WSU containing the raw CICAS-V warning information message, along with the configurable warning flags, it translated this information into the proprietary brake controller message(s) and forwarded this information on to the brake controller.

The haptic brake pulse DVI activation consisted of a single brake pulse presented in conjunction with the visual DVI indication and speech DVI announcement when a warning was provided to the driver. This interface was implemented on two Cadillac STS supplied by VTTI and one Cadillac supplied by GM.

2.3.5 OBE – RSE DSRC OTA Interface

The cooperative nature of the CICAS-V system required the definition of the messages that were sent OTA between the intersection RSE and the vehicle OBE. The OTA messages sent from the RSE are discussed in Section 3.3.4.Two messages were defined to be sent from the OBE to the RSE. While these messages were implemented they were not used by the CICAS-V system. The messages came from the stakeholders via the ConOps and were implemented to make sure the messages were included in the design, so that the design would not preclude future implementation. Thus they are presented below and defined in detail in 0 for completeness.

- Traffic Signal Violation Warning Given (TSVWG) – The TSVWG was defined to alert the infrastructure that a CICAS-V warning was provided to a vehicle's driver
- Remote Command (RCMD) – The RCMD Message was defined to provide a command from the OBE to the RSE force a signal change. This was intended to aid in the Task 11 Objective Testing, however, a different approach was ultimately used to achieve the same results

3 Intersection FOT Ready Prototype Development

The majority of the groundwork for the Task 10 intersection development work took place as part of the Task 8 activities which included the intersection selection process, initial intersection RSE infrastructure builds, initial RSE application module development, preliminary OTA message set definition, infrastructure functional testing, and communications range testing. For the details of these activities, please refer to the Task 8 Final Report. With this groundwork in place, the Task 10 intersection activities primarily focused on updating the RSE SW applications, making the final HW selection along with installing the equipment in the remainder of the intersections, and refining the OTA message formats to meet the final FOT CICAS-V system design.

3.1 Intersection Software Development

The CICAS-V wireless communication system relied on the Institute of Electrical and Electronic Engineers (IEEE) 1609 WAVE and IEEE 802.11p physical layer standards. When taken together, these are commonly referred to as DSRC. CICAS-V used DSRC for broadcasting messages from the intersection RSE to the vehicle OBE and vice versa. The following RSE SW Service Provider Applications (SPAs) were updated for compatibility with the DSRC standards as part of the Task 10 activities:

- GPS Corrections Service Provider Application
- Geometric Intersection Description Service Provider Application
- Signal Phase and Timing Service Provider Application

In addition to making changes to each of the SPAs to support DSRC, changes were also required to support a change in the RSE HW platform used in Task 8 to the DENSO WSU used in Task 10 and Task 11. The SW architecture that was developed to support both of these changes, like the CICAS-V OBE, consisted of two sets of modules:

1. CICAS-V Application Modules – Modules specific to the CICAS-V RSE SW Application
2. WSU Software Services Modules – Generic modules supplied with the DENSO WSU HW that provided services and an API to enable applications to interface to the CAN bus, GPS receiver, and the WAVE Radio

The RSE SW architecture is shown in the following figure followed by a table providing a brief description of each SPA module.

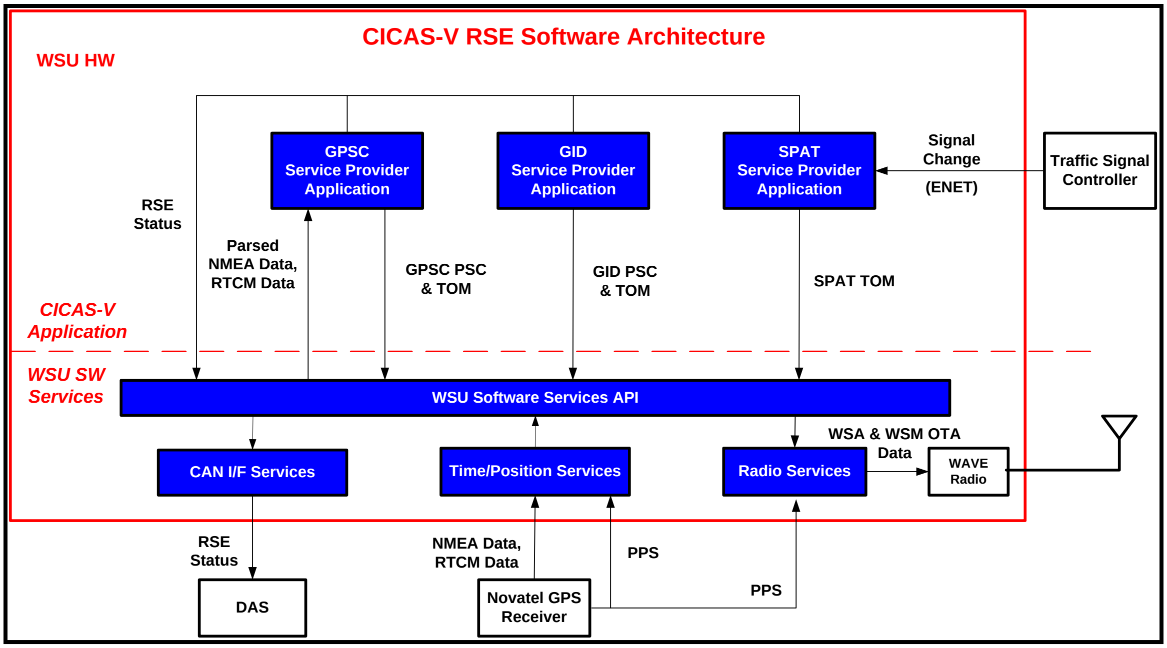

Figure 9: CICAS-V RSE Software Architecture

Table 9: CICAS-V RSE Application Module Summary

Module	Description
GPSC Service Provider Application	• Received corrections data periodically from the base station GPS receiver • Confirmed the validity of this data • Advertised the availability of this service via a WSA broadcast on the control channel • Broadcast the correction data via a WSM on the service channel
GID Service Provider Application	• Retained storage of a map (GID) of the intersection which defined the geometry of the intersection • Advertised the availability of this service via a WSA broadcast on the control channel • Broadcast the GID data via a WSM on the service channel
SPaT Service Provider Application	• Received traffic signal change notifications from the traffic signal controller • Composed the signal, phase, and timing information for each of the intersections approaches into the SPaT message • Broadcast this message via a WSM on the control channel.

Unlike the OBE application modules, the RSE application modules were highly decoupled while sharing just the interfaces to the WSU Software Services. Thus, they were developed separately as stand-alone application executables and were each started up automatically from a script at startup.

0 contains the details for each RSE SPA in the order in which they are listed in Table 9. This information includes the following for each application:

- System Context
- Required and Optional Features
- Functional Requirements
- Constraints and Performance Requirements
- Reference Implementation Overview
- Observed Performance
- Enhancements

3.2 Intersection Hardware Integration

The intersection HW system integration activities involved upgrading the prototype Task 8 intersections with the new RSE HW selected as part of the Task 10 activities, as well as installing all of the intersection HW into the remaining intersections identified in Task 8. Due to variations in the intersection configurations and signal controllers, primarily between CA and MI / VA, the intersection installations had significant differences. The intersection installations that took place in MI and VA were considered to be the most ideal of the two intersection hardware configurations and were considered as characteristic of a future CICAS-V FOT installation. Thus, the configuration of these two intersections will be the one that is discussed in the remainder of this document. The Task 8 Final Report contains a detailed discussion on the CA and MI intersection installations that took place as part of that task.

The complete intersection prototype FOT platform for the CICAS-V system consisted of both a set of CICAS-V hardware system components and a set of DAS hardware system components as detailed in Figure 10 below. Only the CICAS-V hardware components will be discussed further. The DAS hardware components, which were only installed in the VA intersections in conjunction with the Pilot FOT testing, are discussed in detail in the Task 12 Final Report.

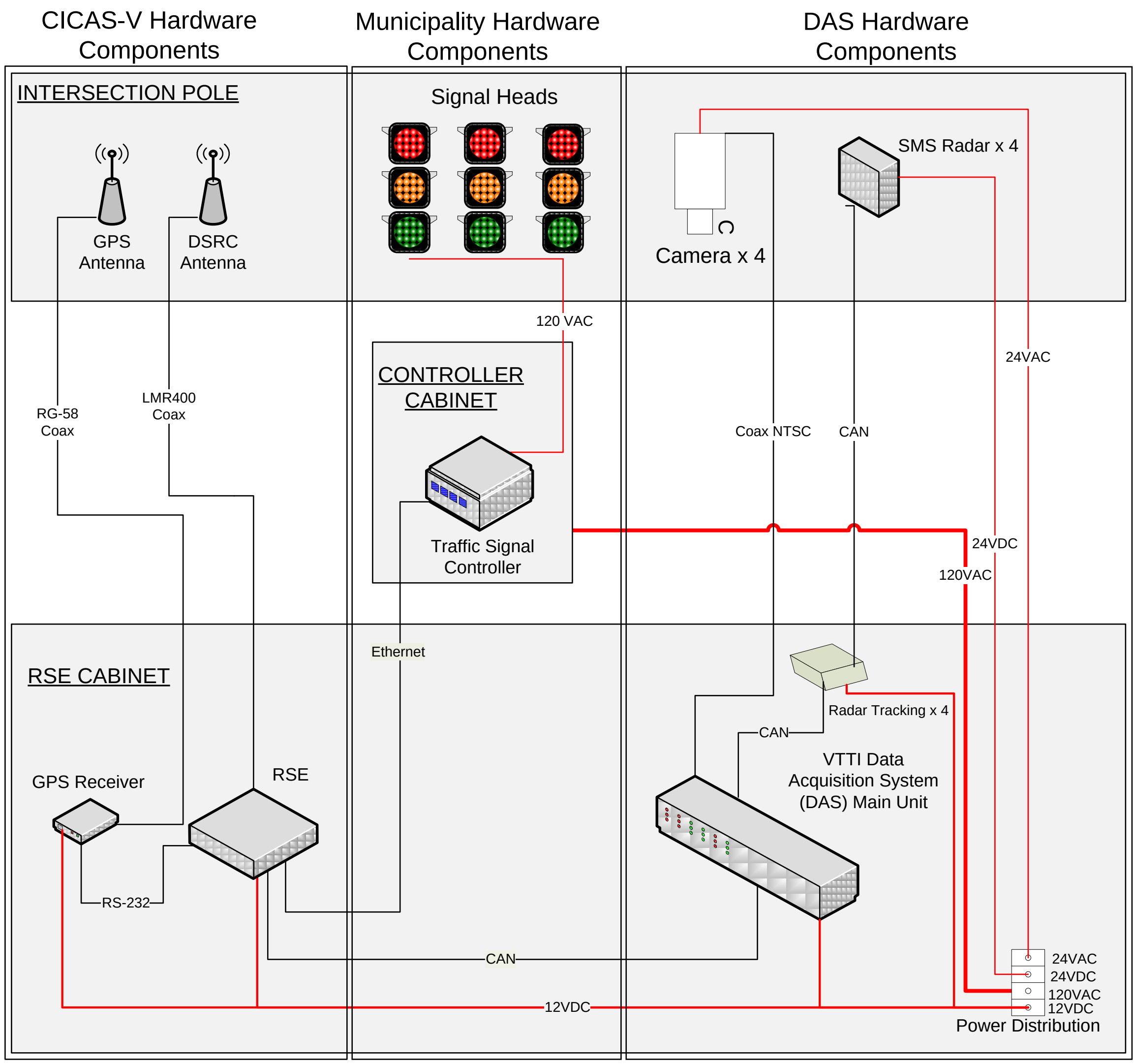

Figure 10: Intersection Prototype FOT HW Platform

Like the vehicle system, the central HW component of the intersection system was the DENSO WSU on which the CICAS-V RSE SW implementation executed. It included a single board embedded computing device utilizing the Freescale MPC5200B PowerPC and extended automotive and IT peripherals along with DSRC radio support. Section 2.2 and Figure 7 of this report detail the physical HW interfaces supported by the WSU followed by a WSU HW and SW feature summary listed in Table 5.

Figure 11 below shows the WSU HW interfaces that were used by the CICAS-V RSE application along with the external HW components that were connected to these interfaces. The tables that follow go into some of the specifics of these external HW components.

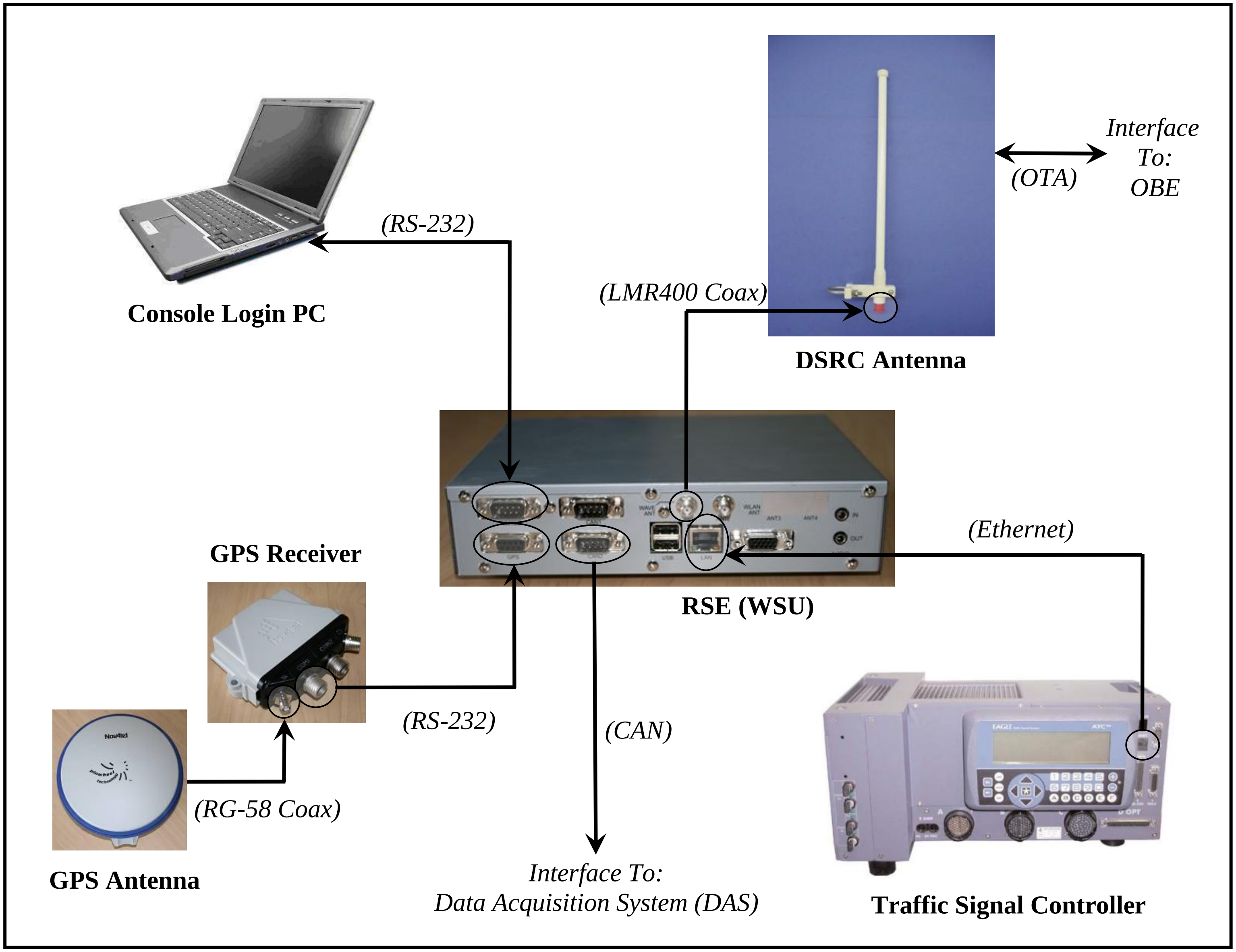

Figure 11: CICAS-V RSE HW Interfaces and Components

Table 10: CICAS-V RSE HW Interface Component Details

HW Component	Manufacturer	Model
RSE	DENSO	WSU-01(A)
Traffic Signal Controller	Siemens	EAGLE EPAC3108 M52
DSRC Antenna	Mobile Mark	ECO12-5800
GPS Receiver	NovAtel®	OEMV® Flexpak V1-L1
GPS Antenna	NovAtel®	GPS-702-GG
Console Login PC	Various	Various

Table 11: CICAS-V RSE HW Interface Cable Details

HW Component	Manufacturer	Model
LMR400 Coax Cable	Various	Various (N Male to N Male with N Female / SMA Male adaptor)
RS-232 Cable	Deutsch	17211-1204-078 (Deutsch Circular to DB9 Male connectors)
RG-58 Coax Cable	NavTech GPS	RG58-xx-TM-TM (TNC Male to TNC Male connectors where xx = length in feet)

3.3 Hardware and OTA Interfaces

For the CICAS-V application SW running on the WSU to work with RSE intersection HW components identified in Table 10, the content for the following interfaces was defined as a part of this Task:

- RSE – GPS Receiver RS232 Interface
- RSE – Traffic Signal Controller Interface
- RSE – DAS CAN Interface
- RSE – OBE DSRC OTA Interface

A brief description of these interfaces and their content is discussed in the following sections and, where appropriate, referenced appendices. Some of the interface content will not be provided due to it being readily available in the public domain.

3.3.1 RSE – GPS Receiver RS232 Interface

This interface existed between the WSU running the CICAS-V RSE SW application and NovAtel OEMV GPS receiver at the intersection. Using the information obtained from this interface the RSE was able to provide locally-produced GPS corrections information to equipped vehicles within transmission range of the RSE. CICAS-V equipped vehicles were then able to use this information to improve their position estimate to better than one meter accuracy allowing them to achieve the “which-lane” positioning determination required by the system.

Two sets of interface messages were defined for receipt by the RSE from the GPS Receiver:

1. RTCM v3.0 messages:
    - RTCM1005 message which contained the coordinates of the antenna reference point for the RSE in addition to other additional minor attributes
    - RTCM1001 message which contained the RSE satellite observations, in particular the single-frequency (L1) corrected pseudo-range and phase-range measurements for each satellite (i.e., based on both basic time of arrival and carrier phase analysis), so vehicle receivers could correct their own local position estimates upon receiving this information from the RSE
2. NMEA messages sent from the GPS Receiver to the OBE:
    - $GPZDA was used to provide the UTC time and date and was used by the GPSC SPA
    - $GPGSA was used to provide the DOP expected accuracy factors and active satellites and was used by the GPSC SPA
    - $GPGGA was used by the WSU SW Time / Position Services
    - $GPRMC was used by the WSU SW Time / Position Services
    - $GPGST was used by the WSU SW Time / Position Services

Both the RTCM and NMEA messages are publicly available standard messages and thus the content definition will not be included as a part of this document.

3.3.2 RSE – Traffic Signal Controller Interface

In order to provide DSRC equipped vehicles within transmission range of a CICAS-V intersection the required SPaT information, a communication interface was designed and implemented to allow status information to be freely shared between a Siemens brand traffic control device and a road-side unit. The status information is transmitted by a Siemens EPAC3108 Model52 using a User Datagram Protocol (UDP) based Ethernet interface and a custom protocol. The protocol allows the traffic control device to transmit current phase and phase timing measurements for all traffic lights at an appropriately equipped intersection. Status updates are provided by the traffic control device whenever there is a change in traffic signal phase (e.g., green-yellow, yellow-red, red-green). This status update also includes the time to next phase with a precision of 1/10 of one second. For adaptive intersections that use infrastructure sensors to modify traffic flow, green phase times typically cannot be reported accurately by the traffic control device. In this case, the CICAS-V system can still function as designed due to the timing of the warnings typically taking place in the yellow phase which remains fixed regardless of traffic flow.

3.3.3 RSE – DAS CAN Interface

This interface existed between the WSU running the CICAS-V RSE SW application and the DAS. The intersection DAS recorded data supplied by the WSU and other HW devices (e.g., radar, cameras). For detailed information on the DAS, please refer to Task 12 Final Report. The RSE updated the infrastructure DAS after transmitting the GPSC, GID, or SPaT DSRC messages with the following data:

- Transportation Object Message (TOM) Layer Type which indicated which message was being transmitted
- Message transmission date and time
- Message transmission counter maintained by each SPA

The OBE logged a similar message upon receipt of these messages and, when taken in combination with the message logged on the RSE, various intersection to vehicle message transmission and reception characteristics could be analyzed.

The data content definition for this interface is defined in 0.

3.3.4 RSE – OBE DSRC OTA Interface

The cooperative nature of the CICAS-V system required the definition of the messages that were being sent OTA between the intersection RSE and the vehicle OBE.

3.3.4.1 WAVE Short Message OTA Message Details

In order to provide a common framework for all the WSM messages, the CICAS-V project created the TOM framework that was based on Extensible Markup Language (XML) but streamlined the message for byte efficiency. A TOM message frame began each message with a Message Header and ended it with a Message Footer. Everything between the header and footer was considered message content which was composed of a hierarchical set of TOM objects. There could only be one frame per message and the frame size had to stay within the limits set by the IEEE 1609 standards including room for potential security and other overhead. Note that security was not implemented as part of this program.

The primary message content object was the Layer Object and was the only object besides a Close Object that was explicitly common to all TOM messages. Even though the object was a Layer Object, each was given a modified name based on its Layer Type. The Layer Objects and thus Layer Types defined for CICAS-V were:

1. GID Layer Object / Type
2. SPaT Layer Object / Type
3. GPSC Layer Object / Type
4. TSVWG Layer Object / Type
5. RCMD Layer Object / Type

Following is a brief description of the first three of the five layer objects listed above. They were defined and implemented for DSRC transmission from the RSE to the OBE as necessary for the system to function. The two remaining layer objects were defined and implemented for DSRC transmission from the OBE to the RSE, however were not used, and are briefly discussed in Section 2.3.5. The detailed definition for all of these objects and their encapsulated objects can be found in 0.

1. Geometric Intersection Description – The GID was defined to provide a digital map of an intersection down to the lane level if necessary. It was designed to provide vehicles with:
    - A local, geo-referenced coordinate system
    - The location of drivable lanes and their corresponding stop bar location
    - A means of mapping lanes to signal phase and timing approach information received separately in the SPaT
    - The movements that were allowable for a given lane

   Note: The GID Layer Object had an Area Object which when incorporated into a TOM message created an Area GID (AGID). The AGID was defined to uniquely identify a collection of intersections with a single wrapper identifier (ID) and encapsulated one or more GIDs into a single message. This was used for transmitting a group of stop sign GIDs in the vicinity of the RSE as a single message which assisted in adhering to the limited payload size of the WSA Provider Service Content (PSC) (explained in the following section).
2. Signal Phase and Timing – The SPaT was designed to provide intersection traffic signal phase and timing information for each approach organized in such a way that a vehicle could reliably determine whether it needed to stop or not before proceeding, depending on which approach it was in.
3. GPS Correction – The GPSC was defined to provide locally-produced GPS corrections information. It was designed to provide vehicles local position correction information allowing them to achieve the "which-lane" positioning determination required by the system.

3.3.4.2 WAVE Service Advertisement OTA Message Details

For the CICAS-V communication, both the GID and GPSC messages were developed as services offered by the RSE and as such required WSAs be transmitted from the RSE on the Control Channel (CCH) per the IEEE 1609.3 standards. For each service being offered, the WSA message content that was defined consisted of:

1. The Provider Service ID (PSID) to uniquely identify the available service to the vehicle.
2. The PSC to provide additional information about the service. This additional information was used by the vehicle to determine the need for parsing the corresponding WSMs being broadcast on the Service Channel (SCH). The size of each PSC was limited to 32 bytes so the additional information that could be provided was limited.
3. The channel number for the SCH on which the service was offered. The SCH was a configurable as part of the RSE SPA start-up script and the same one was used for both the GID and GPSC services.

The detailed definition for the GID and GPSC PSC can be found in 0.

3.3.4.3 Provider Service Identifier Details

PSIDs were used to identify services advertised in WSAs broadcast on the CCH and the corresponding WSM service content broadcast on the SCH. WSMs that were broadcast on the CCH and thus did not require any WSA content to be defined still required a PSID. As was mentioned above, both the GID and GPSC were developed as services to be broadcast on the SCH requiring WSA content to be defined. The SPaT was developed as a service to be broadcast on the CCH. The PSIDs for these services were defined to work within the framework of the Vehicle-Infrastructure Integration Proof-of-Concept (VII POC) program and can be found in 0.

4 System Testing

The purpose of the Task 10 system testing effort was to verify that all of the system components were operational, integrated, and working properly through a combination of component level and intersection level system testing efforts. The execution of these tests was seen as a precursor to the execution of the Task 11 Objective Testing. However there was some overlap between this testing effort and the Task 11 testing effort.

Test procedures were created to test not only the OBE as a whole but also the individual SW components that comprised the OBE. The component test case procedures were based on the SW specification, while the whole OBE test case procedures consisted of a set of intersection driving scenarios intended to verify the OBE was operating as intended.

4.1 Component Test Case Procedures

The component test procedures focused on testing the expected as well as error handling operation of each OBE application module and not the CICAS-V OBE application as a whole. The procedures were written to test the CICAS-V OBE SW specifications developed for each of the CICAS-V application modules shown in Figure 2 and described in Table 1, with the addition of a Power Moding and SW Watchdog set of procedures. In summary these modules are:

- Vehicle Message Handler
- Radio Handler / Data Demux
- GPS Handler
- GID Database Handler
- SPaT Handler
- DAS Handler / Logger
- Violation Detection Modules
- Intersection Identification
- Map Matching / Lane Identification
- Warning Algorithm
- DVI Notifier
- Power Moding
- SW Watchdog Monitor

Where possible, the physical HW components of the entire CICAS-V system were to be used to ensure the whole system was working as intended. Depending on what was required to execute an individual component test case, the system testing may have taken place in either a lab or intersection setting. Some test cases required special SW to be developed for some of the external HW components in order to force some of the required conditions including the error conditions.

Any component level test case procedure that did not meet the expected results, after confirming there was not a system issue, was considered an implementation issue requiring an OBE SW fix. This assumed the severity of the issue warranted a fix. Refer to 0 for the requirements, processing, inputs, and outputs for each of these components and 0 for the individual test cases.

4.2 Intersection Test Case Procedures

4.2.1 Test Procedures Types

The intersection test procedures focused on testing the expected as well as error handling operation of the CICAS-V application as a whole. Three sub-sets of intersection procedures were developed to be tested:

- System Integration Procedures – These tests were run at multiple intersections to verify that some combination of the RSE, OBE, Netway, DVI Icon / Audio, and CICAS-V Warning Algorithm were are all working properly in conjunction with one-another.
- Performance Procedures – These procedures tested various aspects of system performance.
- Operation Procedures – These tested various basic approach maneuvers at each of the intersection types to verify that a warning was or was not provided to the driver as expected.

A scenario test case procedure that did not pass, after confirming there was not a system issue, either indicated an issue with the CICAS-V OBE SW implementation or potentially an issue with the initial prototype algorithm(s) developed under Task 8. Thus, a failed scenario test case procedure resulted in either requiring a corresponding SW fix or a change in implementation, assuming the severity of the issue warranted a fix. Refer to 0 for the individual test cases.

4.2.2 Scenarios Addressed

The procedures tested aspects of the majority of the normal operation scenarios of the CICAS-V system as well as some of the system failure scenarios as listed in the CICAS-V Task 4 – 'Concept of Operations' [4] (ConOps) to include the following:

- Normal Operation Scenarios
    - Simple Traffic Signal Approach Layout
    - Simple Stop Sign Approach Layout
    - Intersections with Dedicated Left or Right Turn Lanes
    - Approaching an Intersection with Limited Positioning Services
    - Flashing Traffic Signal
- System Failure Scenarios
    - Communication Failure
    - Geospatial Information
    - Traffic Signal Phase and Timing
    - Positioning System Correction

The intersection test case procedures were primarily performed at the two signalized intersections and one stop-sign intersection located in MI. These intersections, taken in combination, had the characteristics necessary to test the operation and system failure scenarios listed above. Procedures that required traffic violations to take place in order to confirm the expected results were performed at the VTTI Smart Road located in VA. The geometry and lane information for the MI intersections can be found in 0.

4.3 Test Results

The following table provides a summary of the Task 10 system testing results followed by a discussion on how to interpret the table. In summary, 149 test cases were identified to be run with:

- 131 test cases having no issues (approximately 88%)
- 14 test cases having issues (approximately 9%)
- 13 of the 14 test cases that had issues confirmed fixed
- 1 of the 14 test cases that had issues not confirmed fixed
- 4 test cases not run (approximately 3%)

To see the status for each of the individual test cases please refer to 0.

Table 12: Task 10 System Test Result Summary

Test Procedure Grouping		Grouping Tag	Totals - Grouped							
			Written	Removed	Final Count	No Issues	Issues Identified	Issues Fixed	Issues Not Fixed	Not Tested
Component	Vehicle Message Handler	CMP-VEH	11	0	11	9	2	1	1	0
	Radio Handler / Data Demux	CMP-RAD	27	4	23	22	0	0	0	1
	GPS Handler	CMP-GPSH	9	2	7	4	3	3	0	0
	GID Database Handler	CMP-GID	16	2	14	8	3	3	0	3
	SPAT Handler	CMP-SPAT	4	0	4	4	0	0	0	0
	DAS Handler / Logger	CMP-DASH	8	1	7	6	1	1	0	0
	Intersection Identification	CMP-IID	5	0	5	5	0	0	0	0
	Map Matching / Lane Identification	CMP-LID	6	0	6	6	0	0	0	0
	Warning Algorithm	CMP-WARN	16	6	10	9	1	1	0	0
	DVI Notifier	CMP-DVI	6	0	6	6	0	0	0	0
	Power Moding	CMP-PWR	2	0	2	2	0	0	0	0
	SW Watchdog Monitor	CMP-WDG	2	0	2	2	0	0	0	0
Intersection	System Integration	INT-SI	14	0	14	13	1	1	0	0
	Performance	INT-PRF	13	1	12	10	2	2	0	0
	Operating Scenarios - Simple Traffic Signal Approach	INT-OPS-SIM	16	3	13	12	1	1	0	0
	Operating Scenarios - Simple Traffic Signal Approach	INT-OPS-STP	7	0	7	7	0	0	0	0
	Operating Scenarios - Dedicated Turn Lanes	INT-OPS-DTL	6	0	6	6	0	0	0	0
			168	19	149	131	14	13	1	4

Where:

(red)	Test cases that had issues not confirmed fixed
(yellow)	Test cases not run
(grey)	Test cases removed

Following is how the 'Totals – Grouped' columns should be read:

- Written – Each test procedural grouping was assigned to an OEM to write the test cases. This column represents the number of test cases written for each grouping.
- Removed – When it came time to execute the test cases a number of test cases were removed from consideration and were not run. Some of these reasons included: the test case was invalid; the test case required modifying the CICAS-V application SW, which was out of scope for these types of tests; the test case was covered by other test cases.
- Final Count – This is the number of test cases remaining for each grouping after the 'Removed' test cases had been subtracted from the 'Written' test cases.

- No Issues – Of the ‘Final Count’ test cases this is the number of test cases that met the expected results the first time the test case was run.
- Issues – Of the ‘Final Count’ test cases this is the number of test cases that did not meet the expected results.
- Fixed – Of the ‘Issues’ test cases this is the number that were confirmed fixed in subsequent SW releases. Note: One of the test cases that had an issue was not fixed. It had to do with error reporting from the Netway 6 and was deemed low priority.
- Not Tested – Of the ‘Final Count’ test cases this is the number of test cases that were not run. These were re-prioritized in order to prepare for the Pilot-FOT and Objective testing efforts.

5 References

[1] National Highway Traffic Safety Administration. (2005). *Traffic Safety Facts 2005. (Report No. DOT HS 810 631).* Washington, DC: National Highway Traffic Safety Administration.

[2] Campbell, B., Smith, J., & Najm, W. (2004). *Analysis of Fatal Crashes Due to Signal and Stop Sign Violations. (Report No. DOT HS 809 779).* Washington, DC: National Highway Traffic Safety Administration.

[3] Neale, V., Doerzaph, Z., Perez, M., Sudweeks, J., Kiefer, R., and Maile, M. (In Print). *Cooperative Intersection Collision Avoidance System Limited to Stop Sign and Traffic Signal Violations: Task 3 Final Report.* Washington, DC: National Highway Traffic Safety Administration.

[4] Maile, M., Ahmed-Zaid, F., Caminiti, L., Lundberg, J., Mudalige, P., Pall, C., Garrett, J. K., Kaiser, J. L., Mixon, L. T., and Smith, G. D. (In Print). *Cooperative Intersection Collision Avoidance System Limited to Stop Sign and Traffic Signal Violations (CICAS-V) Concept of Operations*. Washington, DC: National Highway Traffic Safety Administration.

[5] Maile, M., Ahmed-Zaid, F., Basnyake, C., Caminiti, L., Kass, S., Losh, M., Lundberg, J., Masselink, D., McGlohon, E., Mudalige, P., Pall C., Peredo, M., Stinnett, J., and VanSickle, S. (In Print). *Cooperative Intersection Collision Avoidance System Limited to Stop Sign and Traffic Signal Violations (CICAS V) Task 8 Final Report: Prototype Build and Testing.* Washington, DC: National Highway Traffic Safety Administration.

[6] Maile, M., Ahmed-Zaid, F., Basnyake, C., Caminiti, L., Kass, S., Losh, M., Lundberg, J., Masselink, D., McGlohon, E., Mudalige, P., Pall C., Peredo, M., Popovic, Z, Stinnett, J., and VanSickle, S. (In Print). *Cooperative Intersection Collision Avoidance System Limited to Stop Sign and Traffic Signal Violations (CICAS-V) Task 11 Final Report: Objective Tests.* Washington, DC: National Highway Traffic Safety Administration.

[7] Stone, S., Neale, V. L., Wiegand, K., Doerzaph, Z. R. and Maile, M. (In Print). *Cooperative Intersection Collision Avoidance Systems Limited to Stop Sign and Traffic Signal Violations (CICAS-V) Task 12 Final Report: Infrastructure and Vehicle DAS Functional Designs.* Washington, DC: National Highway Traffic Safety Administration.

[8] Vincenty, T., *Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations*, Survey Review XXII, 176, 1975

[9] *NTCIP 9001 – The NTCIP Guide*, v03.02b, published October 2002 by AASHTO, ITE, NEMA

[10] *IEEE P1609.3™ Standard for Wireless Access in Vehicular Environments (WAVE) – Networking Services*. Revision D18. December 2005.

[11] *IEEE P1609.3™/D21 Trial-use Standard for Wireless Access in Vehicular Environments (WAVE) – Networking Services*. September 2006.

[12] *IEEE P1609.4^{TM} Standard for Wireless Access in Vehicular Environments (WAVE) – Multi-channel Operation*. Revision D07. December 2005.

[13] *IEEE 802.11p Draft Amendment to Standard for Information Technology – Telecommunications and information exchange between systems – Local and metropolitan networks – specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical (PHY) specifications: Amendment 3: Wireless Access in Vehicular Environments (WAVE)*. Revision D1.0. February 2006.

OBE SW Specification Details

Figure 12 illustrates the CICAS-V OBE Software Architecture. The CICAS-V Application modules (above the dotted line) are specific to the CICAS-V project. The WSU Software Services modules (below the dotted line) are generic modules supplied with the WSU that provide services and an API to enable applications to interface to the CAN buses, GPS receiver, and the Wireless Access in Vehicular Environments (WAVE) Radio. The scope of the details to follow includes the CICAS-V Application modules only.

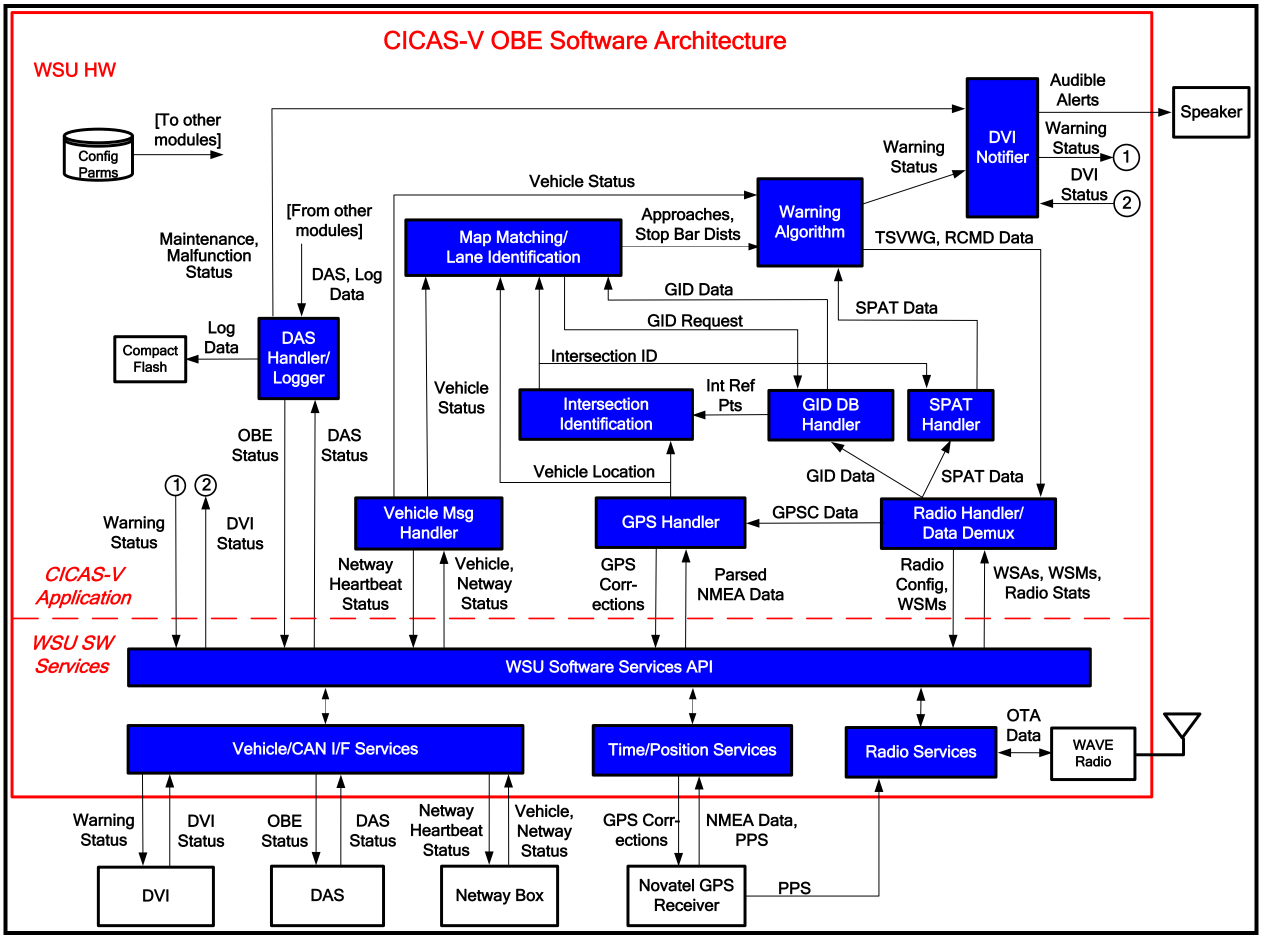

Figure 12: CICAS-V OBE Software Architecture

Figure 13 illustrates the grouping of the software modules into two categories:

- Interface/Message Handling Modules – Interface to external devices and/or perform message handling and parsing functions.
- Violation Detection Modules – Process the latest vehicle, GPS, Geometric Intersection Description (GID), and Signal Phase and Timing (SPaT) data to determine whether an intersection violation is likely to occur.

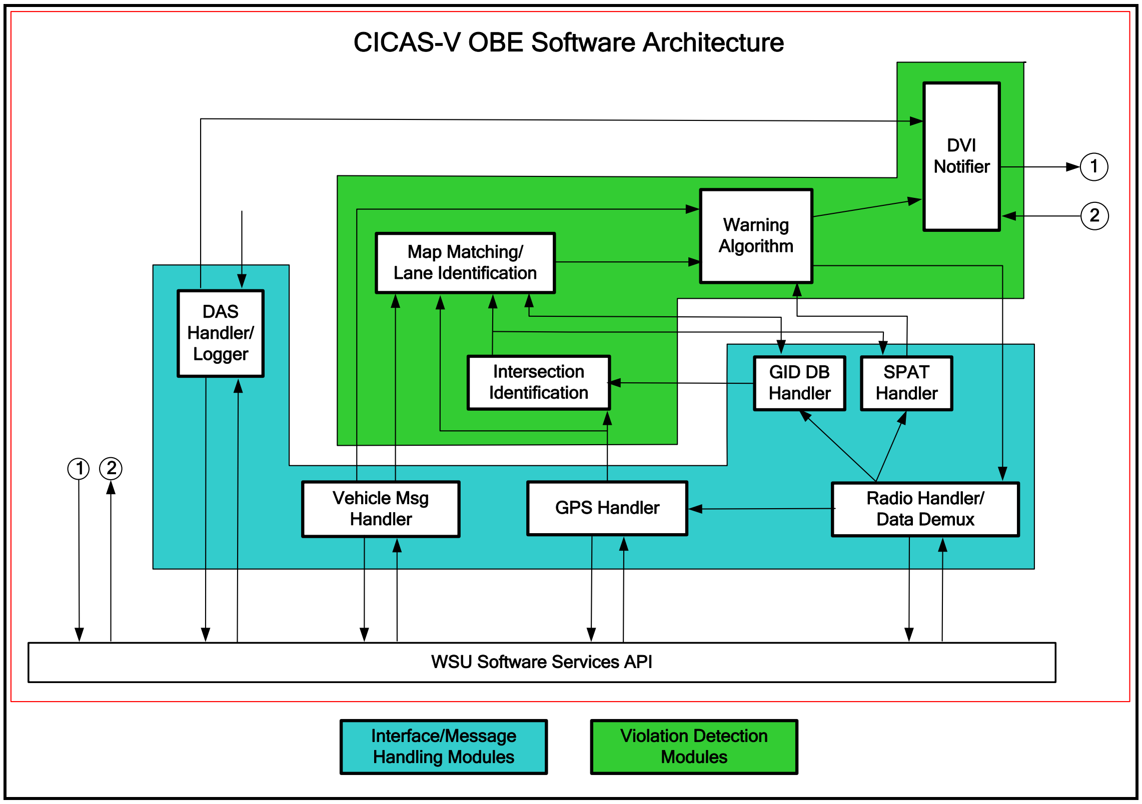

Figure 13: CICAS-V OBE Software Module Categories

The sections that follow will go into the specification details of the CICAS-V Application modules grouped per the two categories discussed above preceded with a discussion of the HW interfaces and some of their high level requirements.

A.1 HW Interfaces

The CICAS-V software executes on the DENSO Wireless Safety Unit (WSU) hardware. Table 13 lists the available WSU hardware interfaces and the intended usage for CICAS-V.

Table 13: WSU Hardware Interfaces

WSU Interface	CICAS-V Usage
Power (+B/ACC)	+ B connected to battery power Accessory (ACC) connected to ignition (IGN) line
CAN 1	Rx – Netway Box, Driver Vehicle Interface (DVI) Tx – Netway Box, DVI
CAN 2	Tx – Data Acquisition System (DAS) Rx – Data Acquisition System
GPS (RS-232C and Pulse Per Second (PPS))	NovAtel OEMV GPS receiver
Audio Out	Speaker (used for audible alerts)
Compact Flash	Application logging
Antenna 1	Radio antenna
Antenna 2	Unused

WSU Interface	CICAS-V Usage
Ethernet	Optional Linux Ethernet console port
RS-232C	Optional Linux serial console port
Audio In	Unused
Cardbu	Unused
General Purpose I/O (GPIO)	Unused
USB 1 and USB 2	Unused
VGA	Unused

Figure 14 illustrates the CICAS-V usage of the hardware interfaces. The following sections provide the requirements for each of these interfaces.

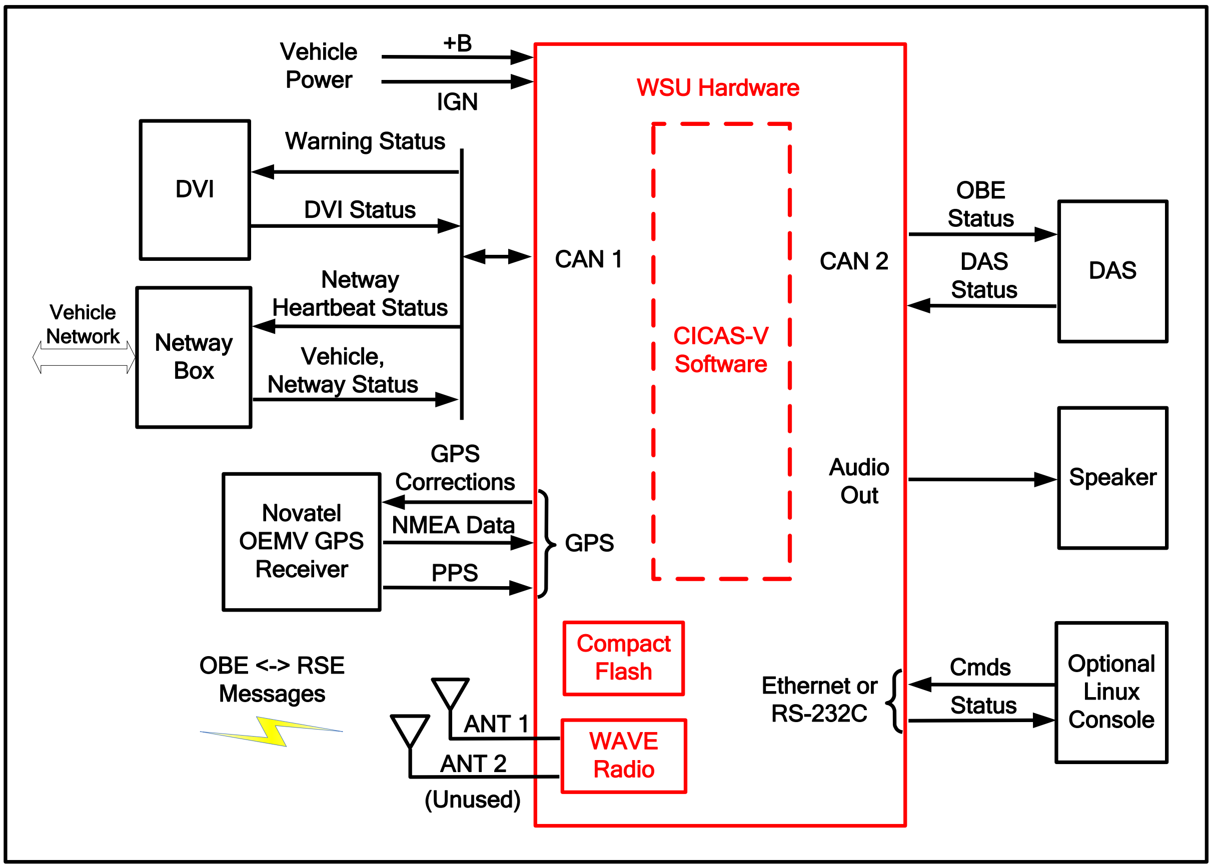

Figure 14: CICAS-V Hardware Interfaces

Vehicle Power

The WSU hardware operates from vehicle Battery (+B) power and provides the software with the capability of monitoring the IGN line. The CICAS-V software shall manage the WSU power up and control the power down based on the IGN status.

Driver Vehicle Interface (DVI)

The CICAS-V software shall interface to the DVI to control the DVI icon and input DVI status. The hardware interface shall be a two wire high speed CAN bus operating at 500 kbps. The DVI will transmit and receive the CAN messages in standard (i.e., 11 bit message identifier) format. Appendix A.22 defines the format and contents of the CAN messages.

Netway Box

The CICAS-V software shall interface to a Netway Box to input vehicle and Netway box status, and output Netway heartbeat information. The CICAS-V software shall support the interface to the Netway6 box. However, other Netway models may be used if the interface is compatible with the Netway6 box. The hardware interface shall be a two wire high speed CAN bus operating at 500 kbps. The Netway Box will transmit and receive the CAN messages in standard (i.e., 11 bit message identifier) format. Appendix A.22 defines the format and contents of the CAN messages.

NovAtel OEMV GPS Receiver

The CICAS-V software shall interface to a NovAtel OEMV GPS Receiver. The hardware interface for the serial input (National Marine Electronics Association (NMEA) data) and serial output (GPS correction data) shall be RS-232C operating at 57,600 bps. The WSU shall interface to the COM2 port of the OEMV receiver. The OEMV PPS signal shall be connected to Pin 1 of the WSU DB9 connector used for the GPS receiver. The software shall output GPS correction data received from the Roadside Equipment (RSE). The software shall use the NMEA and PPS inputs to obtain time and location.

The user must configure the OEMV. The CICAS-V software will not perform any configuration of the device. The CICAS-V software requires the $GPGGA and $GPRMC NMEA messages to be sent at 10 Hz. The software also requires the $GPGSA and $GPGST NMEA messages, which were sent at a lower rate of 2 Hz

Roadside Equipment (RSE) (Over-The-Air Messages)

The CICAS-V software shall interface to the WSU WAVE radio to transmit and receive messages over-the-air to/from the RSE. The CICAS-V WAVE Service Advertisement (WSA) (Appendix A.30), the CICAS-V Provider Service Identifier (PSID) Assignments (Appendix A.31), and the DSRC Message Descriptions and Examples (Appendix A.29) define the format and content of the messages.

Data Acquisition System (DAS)

The CICAS-V software shall interface to the DAS to output OBE status and input DAS status. The hardware interface shall be a two wire high speed CAN bus operating at 500 kbps. The Appendix A.24 defines the CAN message contents. The CICAS-V software shall transmit and receive the messages in standard (i.e., 11 bit message identifier) format.

Speaker

The CICAS-V software shall interface to a speaker to generate audible alerts using the WSU Audio Out interface. The Audio Out interface is compliant with the AC97 standard.

Optional Linux Console

The WSU software supports optionally connecting a Linux console to the Ethernet port or the serial port.

A.2 Interface/Message Handling Software Modules

Vehicle Message Handler

The Vehicle Message Handler interfaces to the Netway box through the WSU Vehicle/CAN Interface Services (VIS). It processes received vehicle status data and distributes the data to other modules. It also exchanges heartbeat messages with the Netway box, and uses timeouts and data validity checks to detect CAN bus communication errors.

A.2.1.1 Requirements

The Vehicle Message Handler requirements are based on the following assumptions:

1. The Netway box transmits and receives the CAN messages defined in the Vehicle-OBE CAN Interface Specification (Appendix A.22). It transmits CAN messages $600 - $605 as a group in numerical order, and messages $606 and $650 on an asynchronous basis.
2. The Netway box transmits the group of messages at a nominal rate of 10 Hz (i.e., every 100 ms). However, the Vehicle Message Handler implementation will support other transmission rates to the extent possible without degrading the overall application performance.

The Vehicle Message Handler shall perform the following functions:

1. Register the Vehicle Message Handler process with the WSU VIS to receive CAN messages $600-605, $606, and $650.
2. Periodically transmit CAN message $704 to the Netway box at a configurable interval. Upon startup, initialize the OBE to Netway Heartbeat Sequence counter to 1, and increment the counter by 1 for each subsequent transmission of $704 to the Netway box.
3. Upon receipt of the group of CAN messages $600-605, check if any of the messages were missing. If so, discard the data. Upon receipt of the complete group of messages, output the data to other modules.
4. Upon receipt of CAN message $650, output the data to other modules.
5. Upon receipt of CAN message $606:
    - Check for an OBE to Netway heartbeat error indication.
    - Check if the Netway to OBE heartbeat sequence counter matches the last number output in the $704 message.
    - Check for vehicle CAN timeout indications.
    - After transmitting a CAN message $704, if no CAN message $606 is received prior to the next periodic transmission of the $704, this will be considered a message $606 timeout.
6. If no complete set of CAN messages $600-$605 has been received for a configurable period, the previous data shall be considered expired. Output a data invalid indication to other modules.
7. Report an error indication (via data to be output to the DAS and/or log entries) to the DAS Handler/Logger upon any of the following conditions:

- WSU VIS indicates a CAN bus driver error has occurred
- Missing CAN message in group of $600-$605
- OBE to Netway heartbeat error indication received in CAN message $606
- Incorrect Netway to OBE heartbeat sequence counter received in CAN message $606
- Vehicle CAN data timeout indication received in CAN message $606
- CAN message $606 timeout
- CAN data expiration

8. Periodically output a message to the DAS Handler/Logger containing data required for the DAS.

A.2.1.2 Inputs

Table 14 summarizes the Vehicle Message Handler Inputs. The Vehicle-OBE CAN Interface Specification defines the contents of the CAN messages.

Table 14: Vehicle Message Handler Inputs

Source	Category	Data
Configuration Parameter File	CAN Parameters	CAN message $704 transmission interval CAN data expiration period
WSU Vehicle/CAN Interface Services	Vehicle Status	CAN bus messages • Messages $600-$606, $650

A.2.1.3 Processing

Figure 15 illustrates the Vehicle Message Handler logic flow.

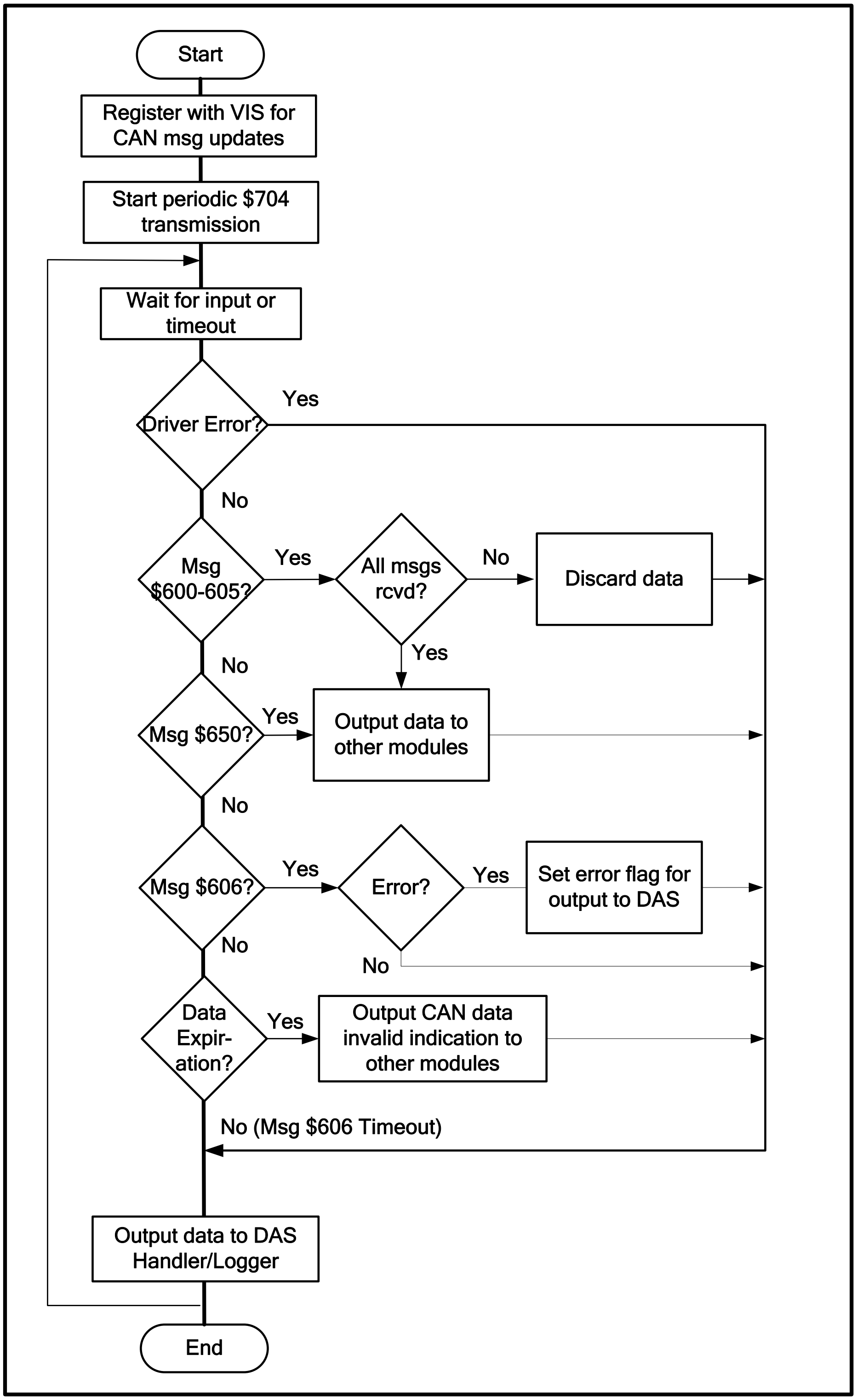

Figure 15: Vehicle Message Handler Logic Flow

A.2.1.4 Outputs

Table 15 summarizes the Vehicle Message Handler Outputs.

Table 15: Vehicle Message Handler Outputs

Destination	Category	Data
WSU VIS	Registration Request	Requested CAN message IDs
	Heartbeat Message	CAN Message $704
Intersection Identification	Vehicle status	Data valid/invalid flag Vehicle speed (Message $601)
Warning Algorithm	Vehicle Status	Data valid/invalid flag Brakes active (Message $600) Driver intended braking level (Message $600) Vehicle speed (Message $601)
DAS Handler/Logger	Vehicle/Netway Status	Vehicle-OBE 1-8 messages $600-$606, $650. Netway to OBE heartbeat error Netway to OBE timeout
	Log Entries	As required by log entries (Appendix 0)

Radio Handler/Data Demux

The Radio Handler/Data Demux module interfaces to the WSU WAVE radio through the WSU Radio Services. The WSU Radio Services includes the WAVE protocol stack.

A.2.1.5 Requirements

The Radio Handler/Data Demux requirements are based on the following assumptions:

1. The RSE transmits the GID, SPaT, and GPSC Transportation Object Modules (TOMs) in separate WSMs.
2. The nominal transmission rates are GID at 2 Hz, SPaT at 10 Hz, and GPSC at 1 Hz. However, the implementation will support other transmission rates to the extent possible without degrading the overall application performance.
3. The GID and GPSC may be broadcast on the Control Channel or Service Channel. The SPaT is always broadcast on the Control Channel. The RSE transmits WSAs for a service only if the data is being broadcast on the Service Channel.

The Radio Handler/Data Demux shall perform the following functions:

1. Configure the WSU WAVE Radio to WAVE mode to support CICAS-V operation.
2. Periodically request radio statistics at a configurable rate (nominally 1 Hz) to support logging of the statistics.
3. Register the CICAS-V application with the WSU Radio Services as a User of the CICAS-V GID, SPaT, GPSC, RCMD, and TSVWG services. The registration requests identify the CICAS-V Provider Service Identifiers (PSIDs). The PSIDs are configurable parameters.

4. Upon notification by the WSU Radio Services that a WAVE Service Advertisement (WSA) has been received, check if the TOM framework and format versions are supported. If so, output the WSA to the GID Handler.
5. Upon receipt of a WAVE Basic Service Set (WBSS) join request from the GID Handler, forward the request to the WSU Radio Services.
6. Upon receipt of a WSM, parse the message to determine if it is a TOM, the TOM framework and layer versions are supported, and it contains GPSC, GID, or SPaT data. If so, provide the TOM to the GPS Handler, GID DB Handler, or SPaT Data handler. If not, report an error indication (via log entry) to the DAS Handler/Logger and discard the message.
7. Upon receipt of a request from the Warning Algorithm to transmit a Traffic Signal Violation Warning Given (TSVWG) or Remote Command (RCMD) message:
    - Compose the TOM.
    - Request the WSU Radio Services to transmit a WSM with the TOM as the payload on the Control Channel. The data rate, transmit power, and priority are specified by configurable parameters.
8. Periodically output a message to the DAS Handler/Logger containing data required for the DAS.

A.2.1.6 Inputs

Table 16 summarizes the Radio Handler/Data Demux inputs.

Table 16: Radio Handler/Data Demux Inputs

Source	Category	Data
Configuration Parameter File	WAVE Parameters	Provider Service Identifiers (PSIDs) for GID, SPaT, GPSC, RCMD and TSVWG services VIIC RSE Flag
	WSM Transmit Parameters	TSVWG transmit parameters: • Data rate • Transmit power • Priority • Repeat count • Repeat Interval RCMD transmit parameters: • Data rate • Transmit power • Priority • Repeat Count • Repeat Interval
	Radio Statistics	Polling interval for radio statistics
WSU Radio Services	WAVE Protocol	WSA indication (plus RSSI) WBSS Join indication WBSS Un-join indication
	Over-the-Air Data	WSMs (plus RSSI) • GPSC TOM • GID TOM • SPaT TOM
	Radio Statistics	RSSI
GID Database Handler	Join Request	WBSS join request
Warning Algorithm	Tx Message	RCMD Message TSVWG Message

A.2.1.7 Processing

Figure 16 and Figure 17 illustrate the Radio Handler/Data Demux logic flow.

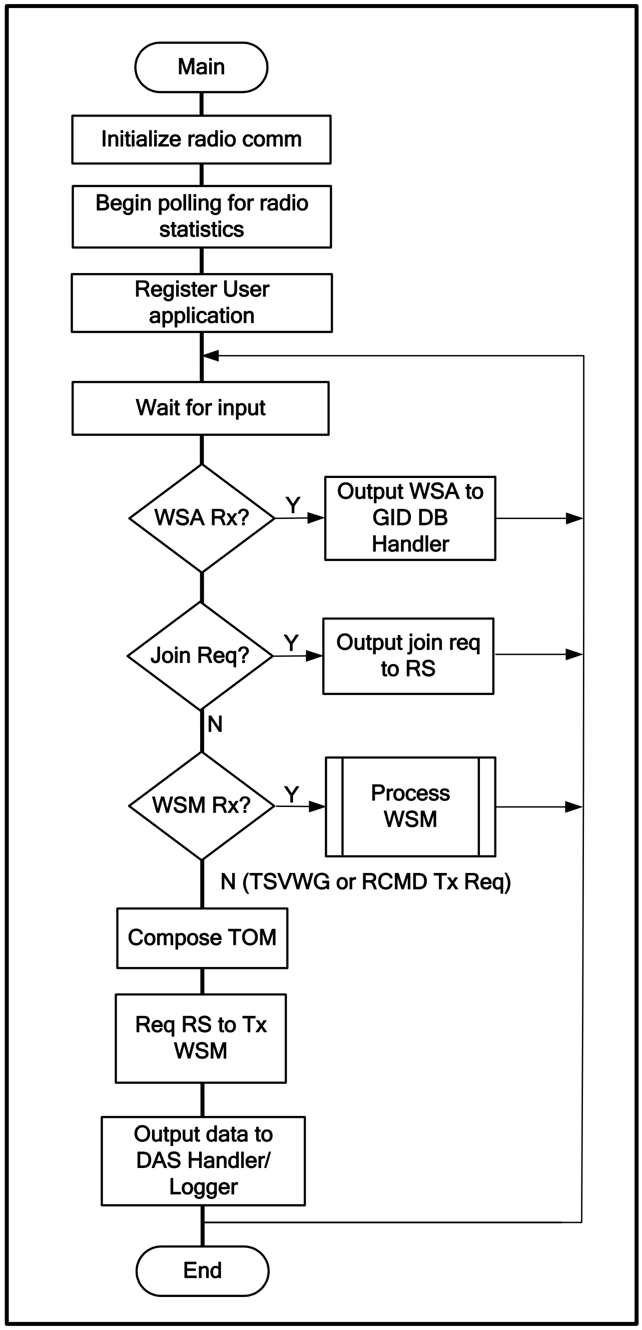

Figure 16: Radio Handler/Data Demux Main Logic Flow

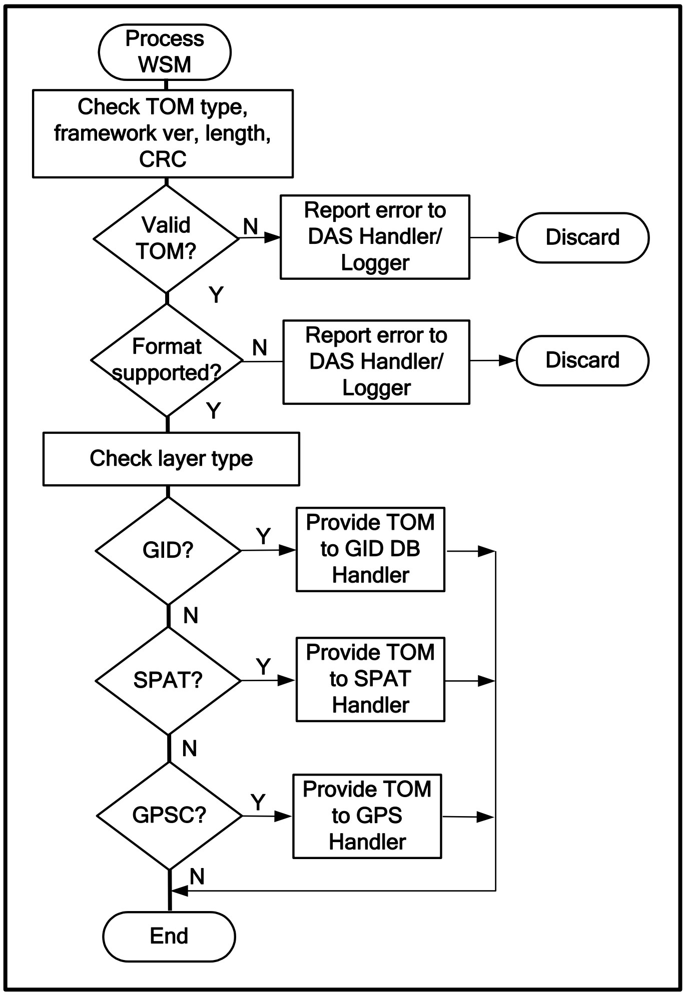

Figure 17: Radio Handler/Data Demux TOM Processing Logic Flow

A.2.1.8 Outputs

Table 17 summarizes the Radio Handler/Data Demux outputs.

Table 17: Radio Handler/Data Demux Outputs

Destination	Category	Data
WSU Radio Services	WSU radio commands	Mode command Request radio driver statistics
	WAVE Commands and Responses	Register User Service command Join WBSS request
GPS Handler	Over-the-Air Data	GPSC TOM
GID Database Handler	Over-the-air Data	GID and GPSC WSAs GID TOM
SPaT Handler	Over-the-air Data	SPaT TOM
DAS Handler/Logger	Over-the-Air Status	WSA received GID TOM Checksum Failure SPaT TOM Checksum Failure GPSC TOM Checksum Failure SPaT received GID received
	Log Entries	As required by log entries (Appendix 0)

GPS Handler

The GPS Handler Module interfaces to the NovAtel OEMV GPS receiver through the WSU Time/Position Services (TPS).

A.2.1.9 Requirements

The GPS Handler requirements are based on the following assumptions:

1. The GPS will be configured to send the $GPGGA and $GPRMC NMEA messages at 10 Hz. It will also be configured to send the $GPGST and $GPGSA messages, however, these may be sent at a lower rate.

The GPS Handler shall perform the following functions:

1. Register the GPS Handler module with the WSU TPS to receive GPS data updates and error status.
2. Upon receipt of NMEA input check for the solution. Check if a checksum error has occurred or no solution is available. If a solution is available, output the parsed NMEA data to other modules.
3. If no GPS input has been received or no GPS solution has been received for a configurable period of time, the previous data shall be considered expired. Output a data invalid indication to other modules.
4. Upon receipt of a TOM with GPS Corrections:

- If the TOM contains a Metric Object, calculate the elapsed time since the time in the Metric object and output the elapsed time data to the DAS Handler/Logger module.
- Check if the GPS Status flag indicates Healthy. If not, discard the data.
- If the TOM contains a zero length for the Radio Technical Commission for Maritime Services (RTCM) 1005 and 1001 messages, perform no additional TOM processing.
- If the TOM contains a non-zero length for the RTCM messages, validate the checksums. If correct, output the corrections to the GPS receiver. If incorrect, discard the data.

5. Report an error indication (via data to be output to the DAS and/or log entries) to the DAS Handler/Logger upon any of the following conditions:
    - WSU TPS indicates a NMEA checksum error has occurred
    - No GPS input
    - No GPS solution (short term)
    - No GPS solution (long term)
    - GPS data expiration
    - GPSC indicates RSE GPS status is not healthy
    - RTCM checksum error
6. Periodically output a message to the DAS Handler/Logger containing data required for the DAS.

A.2.1.10 Inputs

Table 18 summarizes the GPS Handler inputs.

Table 18: GPS Handler Inputs

Source	Category	Data
Configuration Parameter File	GPS Parameters	GPS expiration period GPS solution lost (long term) threshold
WSU TPS	NMEA Data	No solution indication NMEA checksum error Parsed NMEA data
Radio Handler/Data Demux	Over-the-Air Data	GPSC TOM

A.2.1.11 Processing

Figure 18 illustrates the GPS Handler logic flow.

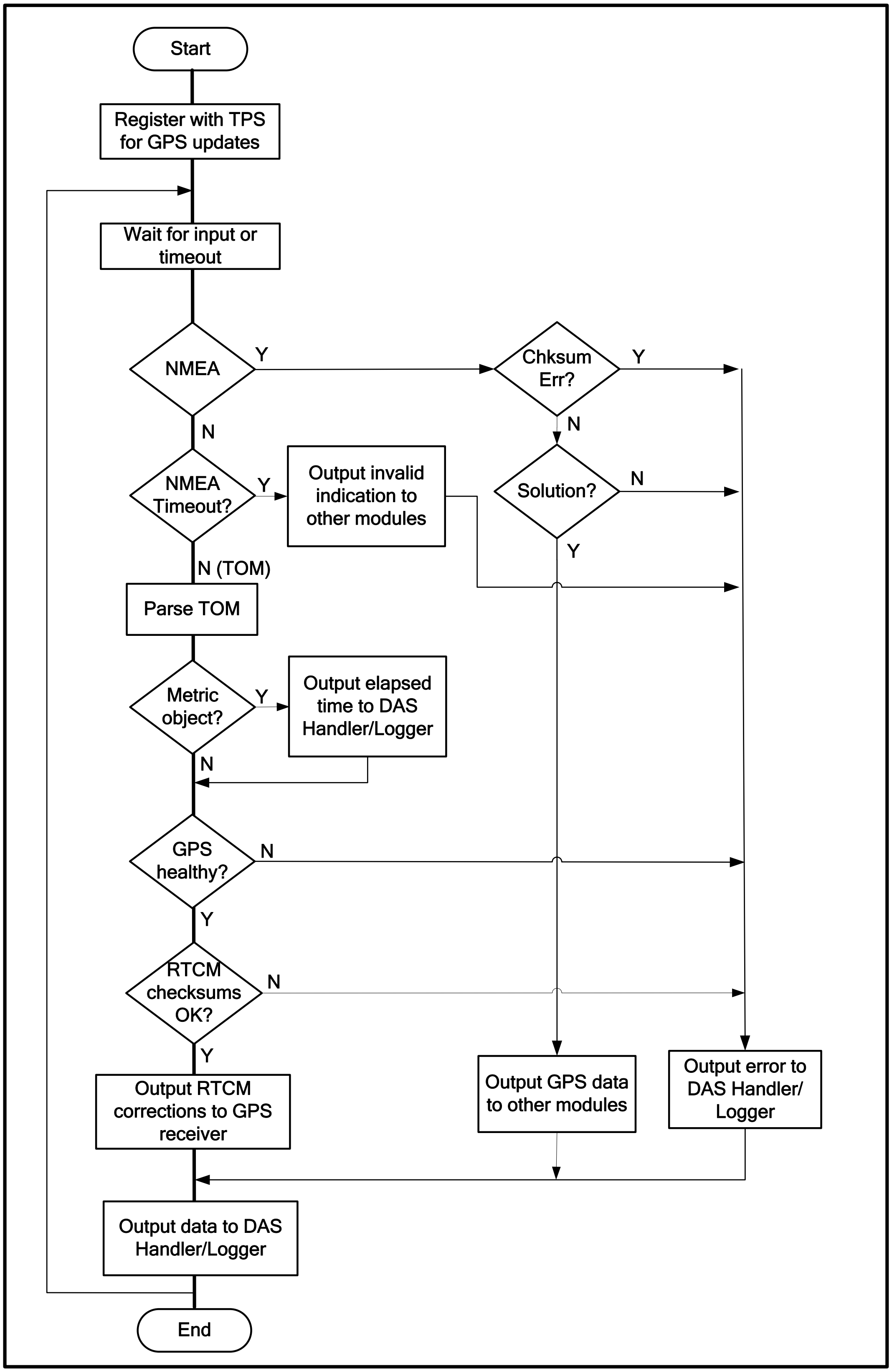

Figure 18: GPS Handler Logic Flow

A.2.1.12 Outputs

Table 19 summarizes the GPS Handler Outputs

Table 19: GPS Handler Outputs

Destination	Category	Data
GPS Receiver (through WSU TPS)	GPS receiver parameters	GPS RTCM corrections
WSU TPS	Registration Request	For NMEA data and error status
Intersection Identification	Vehicle Location and Accuracy	Data valid/invalid flag Latitude Longitude Horizontal Dilution of Precision (HDOP) Heading (Course) Velocity made good Quality indication
Map Matching/Lane Identification	Vehicle Location and Accuracy	Data valid/invalid flag Latitude Longitude HDOP Predicted latitude error Predicted longitude error
DAS Handler/Logger	GPS Data	Vehicle latitude Vehicle longitude Vehicle heading Vehicle altitude Local GPS milliseconds in week Position dilution of precision (PDOP) HDOP Number of Satellites Used GPGST error ellipse Local GPS differential Age Local GPS speed Local GPS solution age Local GPS week number Rover GPS quality/mode GPS communication timeout Inaccurate or no GPS solution (short term) No GPS solution (long term) GPS NMEA checksum error RTCM checksum error
	RSE GPS Status	Base Status GPS Status GPSC reception

Destination	Category	Data
	Elapsed time	Elapsed time since Metric Object time (years, months, days, hours, minutes, seconds, milliseconds) Layer type Message counter
	Log Entries	As required by log entries (Appendix 0.)

GID Database Handler

The GID Database Handler maintains the intersection GID database containing geo-spatial information for all CICAS-V intersections loaded within the configurable expiration period (nominally 30 days) up to a maximum configurable storage limit (nominally 1 MB). It also performs WBSS selection if the GID or GPSC data is being broadcast on the Service Channel.

A.2.1.13 Requirements

The GID Database Handler requirements are based on the following assumptions:

1. The use of Z offsets is not required. However, if the Z offsets are present (indicated by the Z bit in the Node Config object), the GID parsing algorithm must be able to recognize their presence and skip over them.
2. Support for the compressed format (indicated by the C bit in the Node Config object) is not required.
3. The X and Y offsets may be stored in units of centimeters as signed 16 bit integers. If the Node Config Object contains a non-default value for Node Offset Granularity (NOG), the X and Y offset values in the Node List multiplied by the NOG will not exceed values that can be stored in 16 bits.
4. The GID for any intersection will not exceed the following sizes:
    - 32 approaches
    - 6 lanes per approach
    - 24 nodes per lane
    - Maximum of 250 total nodes per intersection

The GID Database Handler module shall perform the following functions:

1. Upon startup, delete GID records that are older than the GID expiration period. If the current database size exceeds the configurable storage allocation (i.e., the allocation was reduced since the previous execution), delete the records with the oldest load times to meet the storage allocation.
2. Output the list of intersection reference points to the Intersection Identification module.
3. Upon receipt of a GID WSA:
    - If WSA contains area GID information, determine the intersection(s) corresponding to the Area ID for the following processing.

- Determine if the GID information has already been received and stored in the local database based on the Area or Intersection ID and content version in the Provider Service Context (PSC) data. If so, update the load time for the corresponding intersection(s) and discontinue processing the WSA.
- If a GID WSA has been received from only one RSE where the GID information is not in the database, request the Radio Handler/Data Demux module to join the corresponding WBSS.
- If a GID WSA has been received from more than one RSE where the GID information is not in the database, select the closest RSE based on the PSC Reference Point information and request the Radio Handler/Data Demux module to join the corresponding WBSS.

4. Upon receipt of a GPSC WSA:

    - If (a) no WBSS has been joined based on GID WSAs, and (b) the GPSC is for the approaching intersection or no intersection has been selected, and (c) the status is healthy and monitored, request the Radio Handler/Data Demux module to join the WBSS.

5. Upon receipt of a GID request from the Map Matching/Lane Identification module, send the parsed GID data for the requested intersection.

6. Upon receipt of a GID TOM:

    - Parse the TOM and convert the data to a format usable by other modules. If the GID contains an Area object, process individual intersection objects within the Area object.
    - If the TOM contains a Metric Object, calculate the elapsed time since the time in the Metric object and output the elapsed time data to the DAS Handler/Logger module.
    - If the database does not contain GID data for the intersection:
        - If the database will exceed the configurable storage allocation upon adding a record, delete the record with the oldest load time.
        - Add a record in the database with the load time (used in determining the expiration time and the oldest records to delete) and the parsed GID data.
        - Send the intersection reference point to the Intersection Identification module.
    - If the database already contains GID data for the intersection:
        - Update the load time.
        - If the received data has a different content version than the stored data, update the parsed GID data and send the intersection reference point to the Intersection Identification module.
    - If a WBSS was joined to obtain the GID, set the join request to inactive.

7. Periodically output a message to the DAS Handler/Logger containing data required for the DAS.

A.2.1.14 Inputs

Table 20 summarizes the GID Database Handler inputs.

Table 20: GID Database Handler Inputs

Source	**Category**	**Data**
Configuration Parameter File	GID Parameters	GID expiration period GID storage allocation
Radio Handler/Data Demux	Over-the-air Data	GID and GPSC WSAs GID TOM
Map Matching/Lane Identification	GID Data Request	Intersection ID

A.2.1.15 Processing

Figure 19 and Figure 20 illustrate the GID Database Handler logic flow.

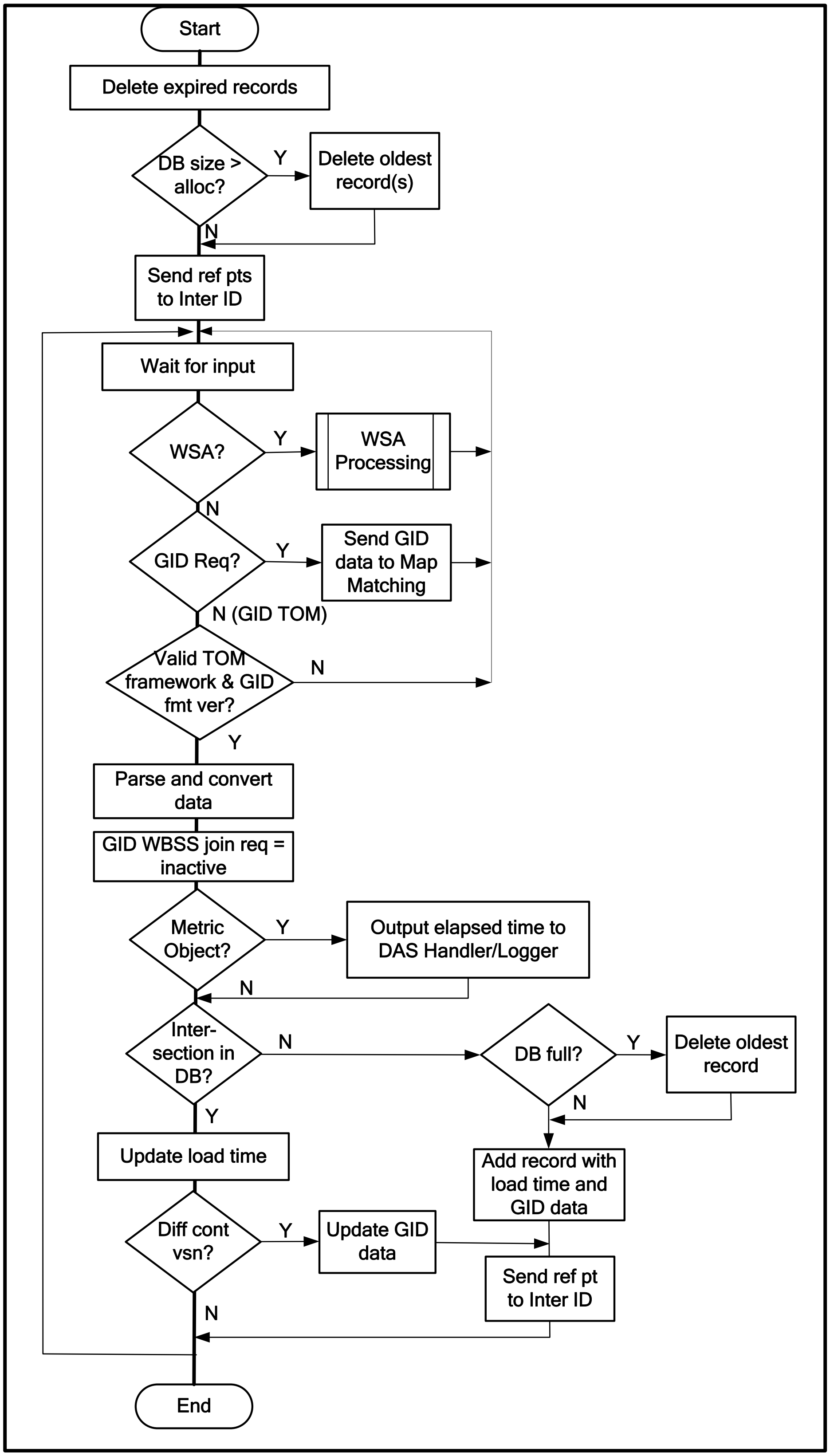

Figure 19: GID Database Handler Logic Flow

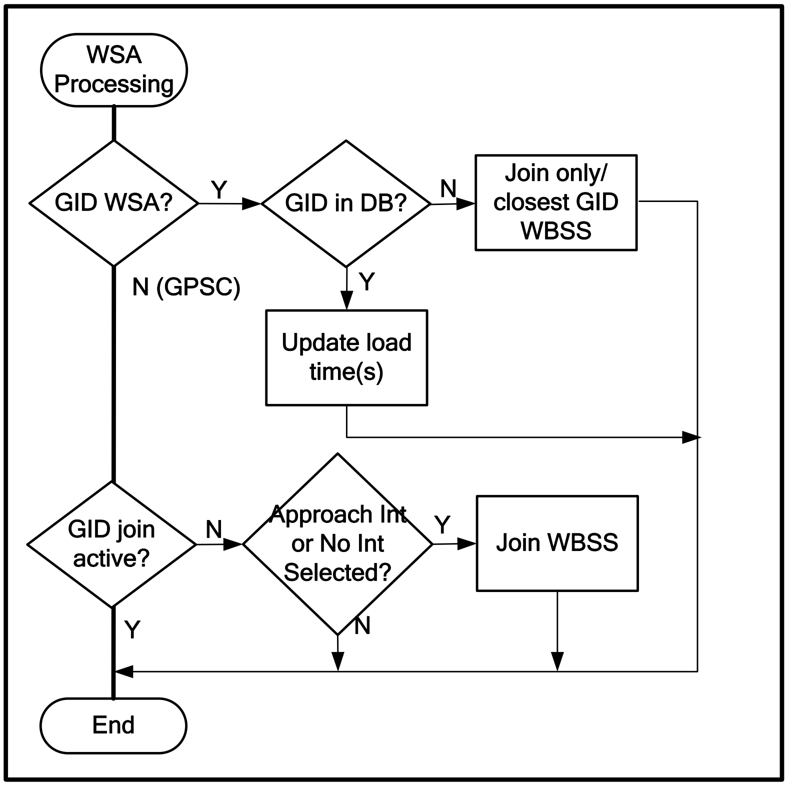

Figure 20: GID Database Handler WSA Processing

A.2.1.16 Outputs

Table 21 summarizes the GID Database Handler module outputs.

Table 21: GID Database Handler Module Outputs

Destination	Category	Data
Radio Handler/Data Demux	Join Request	WBSS join request
Intersection Identification	GID Reference Points	Intersection ID Intersection reference point latitude & longitude Intersection attributes • Lane level accuracy required
Map Matching/Lane Identification	GID Data	Intersection ID Intersection attributes • Signalized or stop sign • Lane level accuracy required Approach information Lane information Stop and start nodes for all lane segments Lane widths for all lane segments
DAS Handler/Logger	Elapsed time	Elapsed time since Metric Object time (years, months, days, hours, minutes, seconds, milliseconds) Layer type Message counter
	Log Entries	As required by log entries (Appendix 0)

SPaT Handler

The SPaT Handler processes the SPaT TOMs.

A.2.1.17 Requirements

The SPaT Handler module shall perform the following functions:

1. Upon receipt of a SPaT TOM, parse the TOM and convert the data to a format usable by other modules.
2. Check if the SPaT data corresponds to the approaching intersection ID (provided by the Intersection Identification Module). If so, output the data to the Warning Algorithm module and check to see if the age exceeds a threshold. If not, discard the data. The OBE may receive SPaT data for other intersections if it is in radio range of more than one intersection.
3. If the TOM corresponds to the approaching intersection ID and contains a Metric Object, calculate the elapsed time since the time in the Metric object and output the elapsed time data to the DAS Handler/Logger module.
4. If no valid SPaT data has been received for a configurable period of time, the previous data shall be considered expired. Output a data invalid indication to the Warning Algorithm module.

5. Report an error indication (via data to be output to the DAS and/or log entries) to the DAS Handler/Logger upon any of the following conditions:
   - SPaT age greater than configurable threshold
   - SPaT expiration
6. Periodically output a message to the DAS Handler/Logger containing data required for the DAS.

A.2.1.18 Inputs

Table 22 summarizes the SPaT Handler inputs.

Table 22: SPaT Handler Inputs

Source	Category	Data
Configuration Parameter File	SPaT Expiration	SPaT expiration period SPaT age threshold
Radio Handler/Data Demux	Over-the-air Data	SPaT TOM
Intersection Identification	Intersection Data	Approaching Intersection ID (or none available)

A.2.1.19 Processing

Figure 21 summarizes the SPaT Handler logic flow.

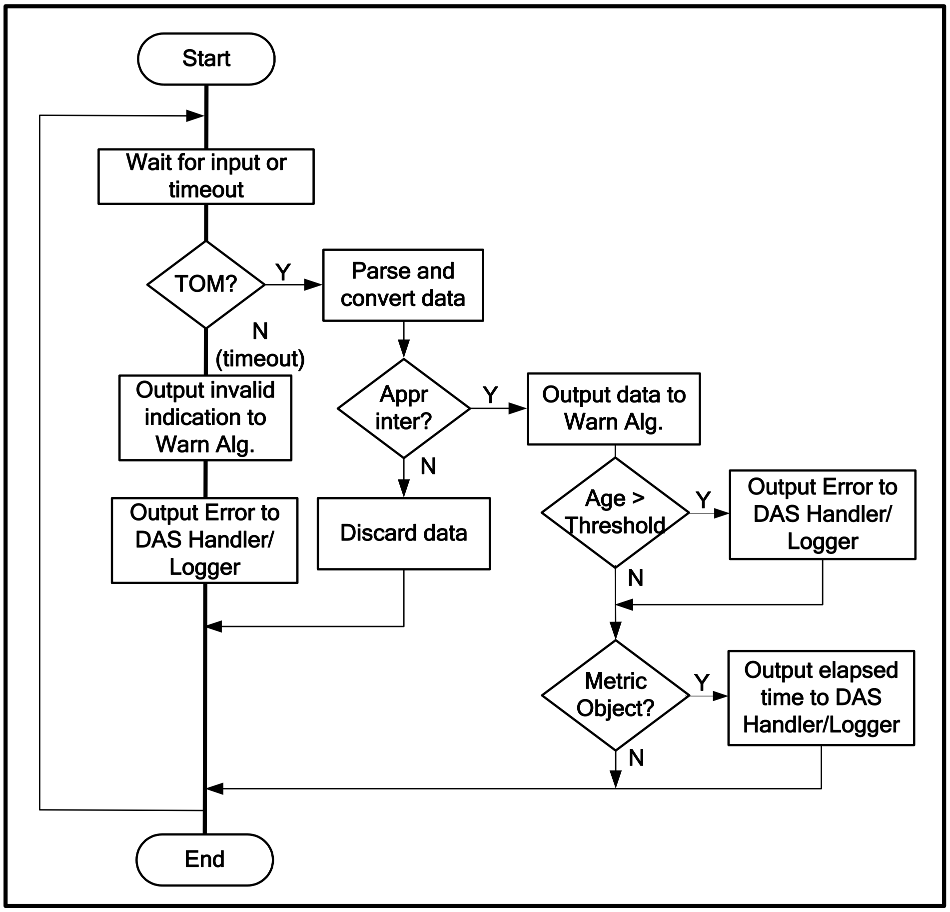

Figure 21: SPaT Handler Logic Flow

A.2.1.20 Outputs

Table 23 summarizes the SPaT Handler module outputs.

Table 23: SPaT Handler Module Outputs

Destination	Category	Data
Warning Algorithm	SPaT Data	Data valid/invalid flag Intersection ID Signal phase information (color) Time until next signal phase Yellow duration SPaT current time
DAS Handler/Logger	SPaT Data	SPaT timeout SPaT age greater than configurable threshold
	Elapsed time	Elapsed time since Metric Object time (years, months, days, hours, minutes, seconds, milliseconds) Layer type Message counter
	Log Entries	As required by log entries (Appendix 0)

DAS Handler/Logger

The DAS Handler/Logger interfaces to the DAS through the WSU Vehicle/CAN Interface Services to receive DAS status and output OBE status. The DAS Handler/Logger outputs over-the-air Metric Object data to the DAS on an event-driven basis and other OBE status to the DAS at a 10 Hz rate. It also generates an application log file based on a configurable log mask. The OBE-DAS CAN Interface Definition for CICAS-V (Appendix A.24) defines the DAS message format.

The DAS Handler/Logger also performs OBE error handling functions. It processes error indications from other modules, maintains the OBE health status as normal, maintenance or malfunction, and manages the hardware and software watchdog timers.

A.2.1.21 Requirements

The DAS Handler/Logger shall perform the following functions:

1. Register with the WSU VIS to receive CAN message $701.
2. Upon startup, initialize the OBE to DAS Heartbeat Sequence counter to 1, and increment the counter by 1 for each subsequent transmission of CAN message $606 to the DAS.
3. Upon startup, activate the WSU hardware watchdog timer to a configurable interval and set a software watchdog timer to a configurable interval.
4. Upon receipt of CAN message $701:
    - Check for an OBE to DAS heartbeat error indication. Also check if the DAS heartbeat sequence counter matches the last number output in the $606 message.
    - Check the other DAS error indications.
5. Receive input data from other modules.

6. If the data must be output to the DAS:
   - Check if the data is for DAS message $751. If so, output the message immediately.
   - If the data is not for message $751, buffer the data. Output DAS messages $600-$606, $610-$619, and $650 at the next 10 Hz interval.
   - Check if a DAS data message has been received from all of the CICAS-V modules since the last software watchdog timer reset. If so, reset the software and hardware watchdog timers.
7. If the data must be logged (enabled by a user configurable log mask), write a record to the log file.
8. Maintain the status of OBE error conditions and set the OBE health state (Normal, Maintenance, or Malfunction) based on the current error status.
   - If the CAN message $701 or input data from other modules contains an error indication (set or clear), update the corresponding error status if the indication (set or clear) persists for more than a configurable duration.
   - Set the OBE health state based on the most severe currently active error (error levels for each error type are user configurable).
   - Notify the DVI Notifier when the OBE health state changes.
9. Upon expiration of the software watchdog timer, log the error. If the configurable amount of time elapses without the hardware watchdog timer being reset, the hardware watchdog will reset the WSU.

A.2.1.22 Inputs

Table 24 summarizes the DAS Handler/Logger Inputs for the DAS messages. The CICAS-V Log File Definition (Appendix 0) defines the parameters for the log file entries.

Table 24: DAS Handler/Logger Inputs

Source	**Category**	**Data**
Configuration Parameter File	Log Mask	Mask indicating which log entries should be written to the log file DAS message logging interval Hardware watchdog enable Hardware watchdog timeout Software watchdog timeout Malfunction persistence threshold Maintenance persistence threshold DAS drive space threshold DAS low battery threshold Error level type (maintenance or malfunction) for each error affecting the OBE health status
WSU Vehicle/CAN Interface Services	DAS Status	OBE-DAS message 11 ($701)
Vehicle Message Handler	Vehicle Status	Vehicle-OBE 1-8 messages ($600-$606, $650) Netway to OBE heartbeat error Netway to OBE timeout
Radio Handler/Data Demux	Over-the-Air Status	WSA received GID TOM Checksum Failure SPaT TOM Checksum Failure GPSC TOM Checksum Failure SPaT received GID received

Source	**Category**	**Data**
GPS Handler	GPS data	Vehicle latitude Vehicle longitude Vehicle heading Vehicle altitude Local GPS milliseconds in week Position dilution of precision (PDOP) HDOP Number of Satellites Used GPGST error ellipse Local GPS differential Age Local GPS speed Local GPS solution age Local GPS week number Rover GPS quality/mode GPS communication timeout Inaccurate or no GPS solution (short term) No GPS solution (long term) GPS NMEA checksum error RTCM checksum error
	RSE GPS Status	Base Status GPS Status GPSC reception
SPaT Handler	SPaT Data	SPaT timeout SPaT age greater than configurable threshold
GPS Handler, GID Handler, or SPaT Handler	Metric Object Data	Elapsed time (years, months, days, hours, minutes, seconds, milliseconds) Layer type Message counter
Intersection Identification	Intersection Data	Approaching intersection ID (or none available) GID map version
	IID Status	GPS data validity GPS position validity GPS solution (HDOP) status Intersection in range status Intersection closing status Low vehicle speed threshold status GPS high accuracy mode status
Map Matching / Lane Identification	Map Matching Status	Vehicle is off GID Vehicle is off lane on GID Approach validated

Source	Category	Data
	Lane Matching Data	Number of approaches Present approach Approach likelihood Number of lanes Present lane Lane likelihood Intersection type Road level/lane level Distance to lane / approach center line Distance to stop bar
Warning Algorithm	Warning Data	Warning Algorithm status Time to intersection
DVI Notifier	DVI Data	DVI to OBE heartbeat error OBE to DVI heartbeat error
All	Log Entries	As required by log entries (Appendix 0)

A.2.1.23 Processing

Figure 22 and Figure 23 summarize the DAS Handler/Logger processing.

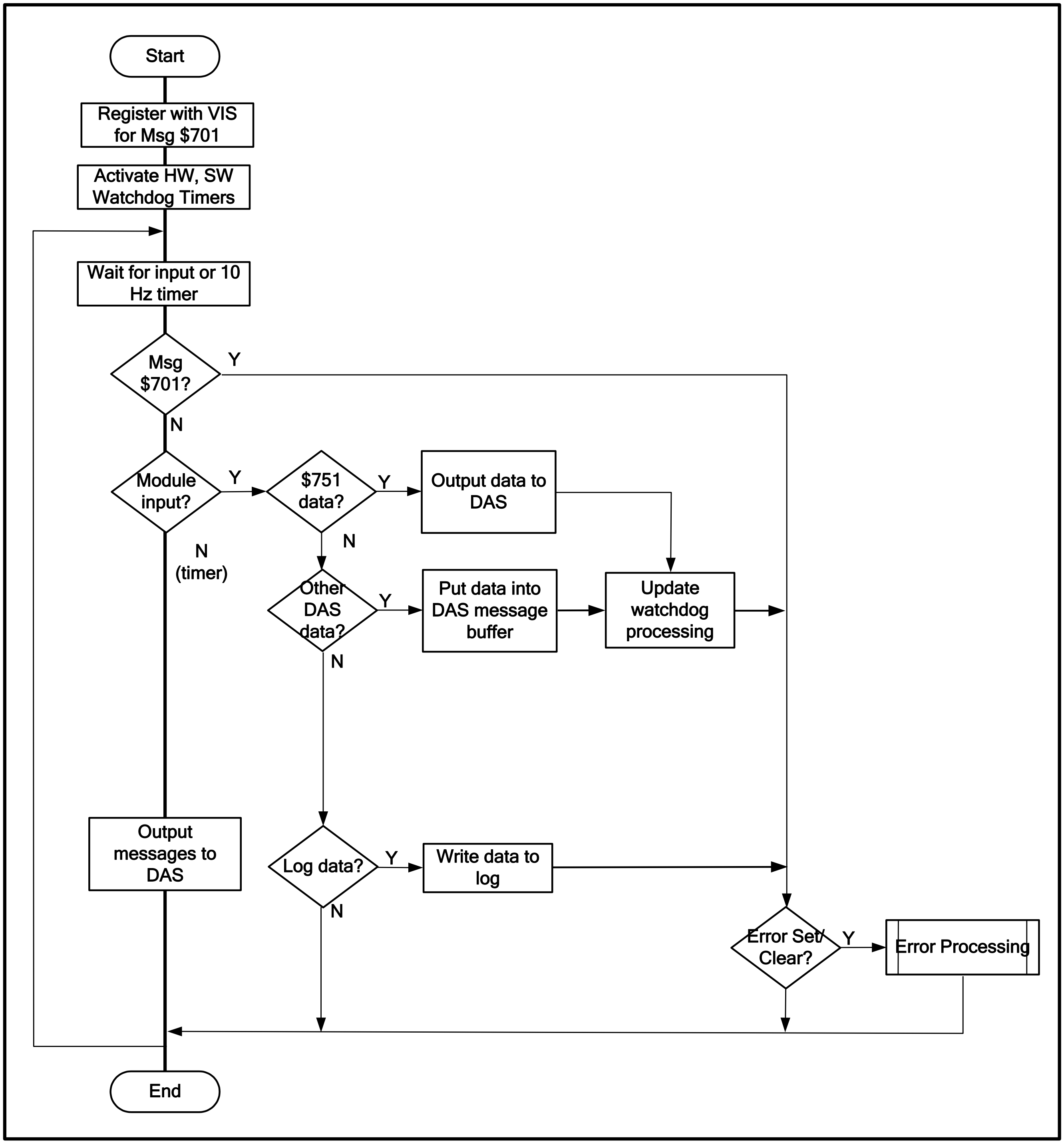

Figure 22: DAS Handler/Logger Logic Flow

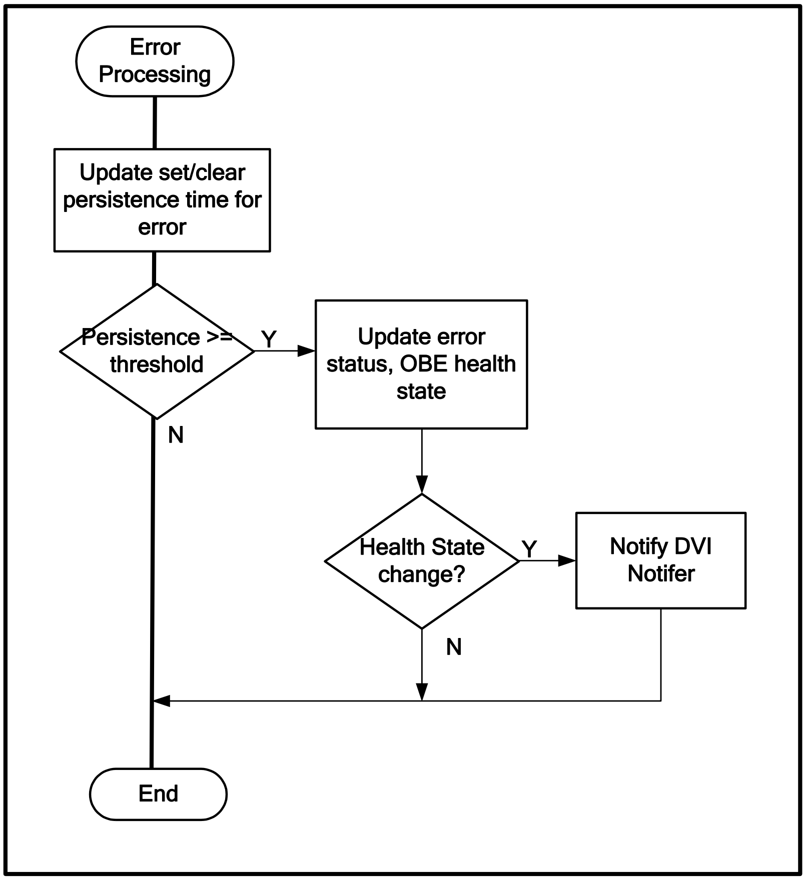

Figure 23: DAS Handler/Logger Error Processing

A.2.1.24 Outputs

Table 25 summarizes the DAS Handler/Logger Outputs.

Table 25: DAS Handler/Logger Outputs

Destination	Category	Data
DAS (through WSU VIS)	All	CAN Bus messages Vehicle-OBE 1-8 messages ($600-$606, $650) OBE-DAS 1-10, 12 messages ($610-619, $751)
Application Log file	Log Parameters	Determined by configurable log mask.
DVI Notifier	OBE health status	Maintenance status Malfunction status

A.3 Violation Detection Software Modules

Intersection Identification

The Intersection Identification module determines the nearby intersections and identifies the most likely approaching intersection among them based on the vehicle location and the information in the GID database.

A.3.1.1 Requirements

The Intersection Identification module shall perform the following functions:

1. Report an error (GPS Insufficient Solution) to the Error Handling module whenever HDOP exceeds a configurable threshold for a configurable timeout period. Report an error (Invalid Position) if the Latitude or Longitude is invalid. No intersection is identified.
2. Using the Haversine formula, calculate the great circle distance from all intersection locations found in the GID database to the current vehicle position.
3. Shortlist intersections that are within a certain configurable distance threshold (nominally 300 m) as nearby intersections.
4. Calculate the rate of change in distance to each short listed intersection location using three methods. Use the result from the method selected by a configuration parameter:
    a. CAN vehicle speed and GPS heading.
    b. GPS location data.
    c. GPS location data with filter.
5. If the low speed filtering is disabled (by a configurable parameter) or the vehicle speed exceeds a configurable threshold (nominally 3 miles per hour), identify the most likely approaching intersection as the one:
    - The rate of change in distance indicating the vehicle's direction of travel is toward the intersection.
    - Meets the configurable intersection selection criterion which is either approaching most quickly or closest distance.
6. If low speed filtering is enabled and the vehicle speed is equal to or lower than the configurable threshold, identify the most likely intersection as the previously identified intersection (or no intersection if none was previously identified).
7. If the identified intersection indicates lane level accuracy is required and the vehicle GPS quality indicates it is not in high accuracy mode, then no intersection is identified.
8. Log the intersection rate of change from all three methods along with CAN vehicle speed, GPS reported heading and velocity made good.
9. Periodically output a message to the DAS Handler/Logger containing data required for the DAS.

A.3.1.2 Inputs

Table 26 summarizes the Intersection Identification module inputs.

Table 26: Intersection Identification Module Inputs

Source	Category	Data
Configuration Parameter File	Algorithm Parameters	HDOP threshold for usable GPS data HDOP threshold timeout Great circle distance threshold for candidate intersections Moving average duration (used in GPS location method) Filter sampling time (used in GPS location data with filter method) Filter cut off frequency (used in GPS location data with filter method) Low vehicle speed filtering enable Low vehicle speed threshold Intersection rate of change method result to be used Intersection selection criterion (Approaching most quickly or Closest distance).
Vehicle Message Handler	Vehicle Status	Data valid/invalid flag Vehicle speed
GPS Handler	Vehicle Location and Accuracy	Data valid/invalid flag Latitude Longitude HDOP Heading (Course) Velocity made good Quality indication
GID Database Handler	GID Data	Intersection ID Intersection reference point latitude & longitude Intersection attributes • Lane level accuracy required
Intersection Identification	Computed data	Previous great circle distance

A.3.1.3 Processing

Figure 24 illustrates the Intersection Identification module logic flow.

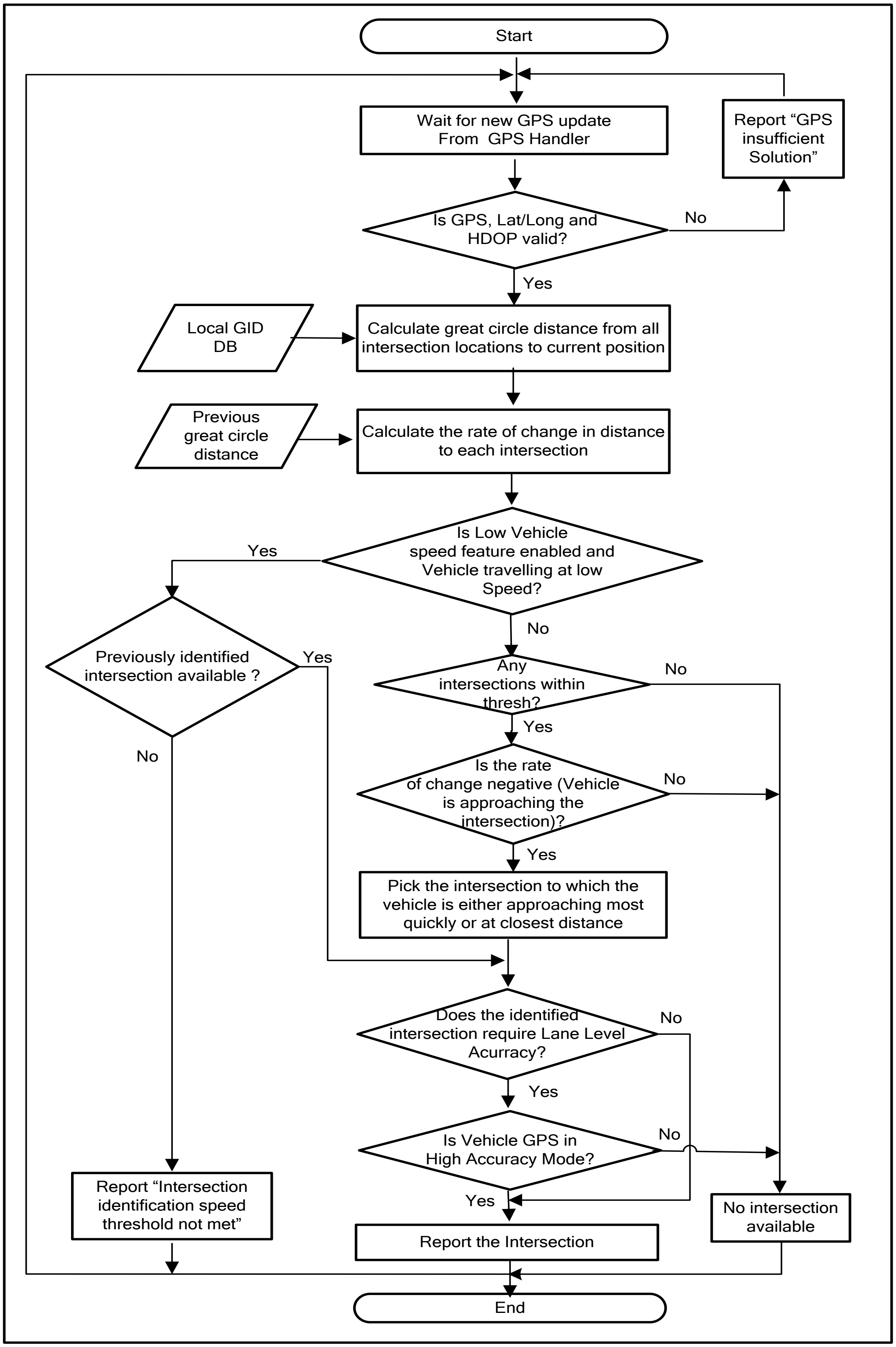

Figure 24: Intersection Identification Module Logic Flow

A.3.1.4 Outputs

Table 27 summarizes the Intersection Identification module outputs.

Table 27: Intersection Identification Module Outputs

Destination	Category	Data
SPaT Handler	Intersection Data	Approaching Intersection ID (or none available)
Map Matching/Lane Identification	Intersection Data	Approaching intersection ID (or none available)
DAS Handler/Logger	Intersection Data	Approaching intersection ID (or none available) GID map version
	IID Status	GPS data validity GPS position validity GPS solution (HDOP) status Intersection in range status Intersection closing status Low vehicle speed threshold status GPS high accuracy mode status
	Log Entries	As required by log entries (Appendix 0)

Map Matching/Lane Identification

The Map Matching/Lane Identification module determines the approach ID of the current lane of travel, the distance to the stop bar and likelihood of approach based on the vehicle location and the information in the GID database. Depending upon number of approaches matched, this module may report information for more than one approach.

A.3.1.5 Requirements

The Map Matching/Lane Identification module shall perform the following functions:

1. Map the location of the vehicle to the intersection GID.
2. Calculate the distance between each lane segment and the current vehicle position to identify the nearest segment for each lane on the GID.
3. If the vehicle is inside the lane or the configurable always return match flag is true:
    a. Identify the lane as the probable lane of travel.
    b. Use the GPS reported 1-sigma error to determine the confidence level of the probable lane of travel.
    c. If the configurable baseline behavior flag is true, report the closest lane approach as the match. Otherwise, calculate the likelihood of all lanes in the intersection excluding the probable lane of travel.
4. If a GID is constructed in such way that 2 lanes have 1 or more node points that are identical (e.g., a right or left turn lane branches off an existing lane), then it is possible for the vehicle to exist in two lanes simultaneously. If this occurs, calculate the likelihood of all lanes in the intersection and select the most likely lane.
5. If the vehicle is not inside a lane (no probable match exists), check if the vehicle is off GID using one of two methods selected by a configuration parameter:

a. Distance from centerline method – the mathematical minimum of all lane segments' distances to the vehicle is greater than a configurable off GID threshold.
b. Distance from lane edge method – the minimum distance to nearest lane edge is greater than a configurable off GID threshold.

If so, report Off GID. Otherwise, calculate the likelihood of all lanes in the intersection.

6. If the likelihood of one lane exceeds a configurable threshold, then it is the only lane to be reported. Otherwise, consider the configurable maximum number of lanes to report, selecting the lanes with the highest likelihood. Identify the corresponding approach number for the lane(s) from the information in GID database.
7. If the number of approaches identified is more than one, then correlate the Lane matches to the Approach matches and aggregate the approach likelihood.
8. Select the approaches to be reported to the Warning Algorithm module:
   a. If the highest likelihood approach is not above a configurable minimum likelihood approach threshold, do not report any approaches.
   b. Report additional approaches if the difference between the highest likelihood approach and their likelihood is less than or equal to a configurable likelihood difference threshold.
   c. The minimum and maximum likelihood threshold values used above differ based on the availability of local GPS corrections.
9. Calculate the distance along the most likely path from the vehicle location to the stop bar for each approach that is reported to Warning Algorithm. Report this information along with Approach ID and approach likelihood.
10. Periodically output a message to the DAS Handler/Logger containing data required for the DAS.

A.3.1.6 Inputs

Table 28 summarizes the Map Matching/Lane Identification module inputs.

Table 28: Map Matching/Lane Identification Module Inputs

Source	Category	Data
Configuration Parameter File	Config data	Always return match flag Off-GID method Off-GID threshold Baseline behavior flag Map match single match likelihood threshold Maximum number of returned lanes Minimum approach likelihood for Warning Algorithm Maximum approach likelihood difference for Warning Algorithm Minimum approach likelihood with local corrections for Warning Algorithm Maximum approach likelihood difference with local corrections for Warning Algorithm GPS antenna offset, vehicle travel direction Note: Offset is from the GPS antenna to the front bumper.
GPS Handler	Vehicle Location and Accuracy	Data valid/invalid flag Latitude Longitude HDOP Predicted latitude error Predicted longitude error
Intersection Identification	GID Data	Approaching intersection ID (or none available)
GID Database Handler	GID Data	Intersection ID Intersection attributes • Signalized or stop sign • Lane level accuracy required Approach information Lane information Stop and start nodes for all lane segments Lane widths for all lane segments

A.3.1.7 Processing

Figure 25 and Figure 26 illustrate the Map Matching/Lane Identification module logic flow.

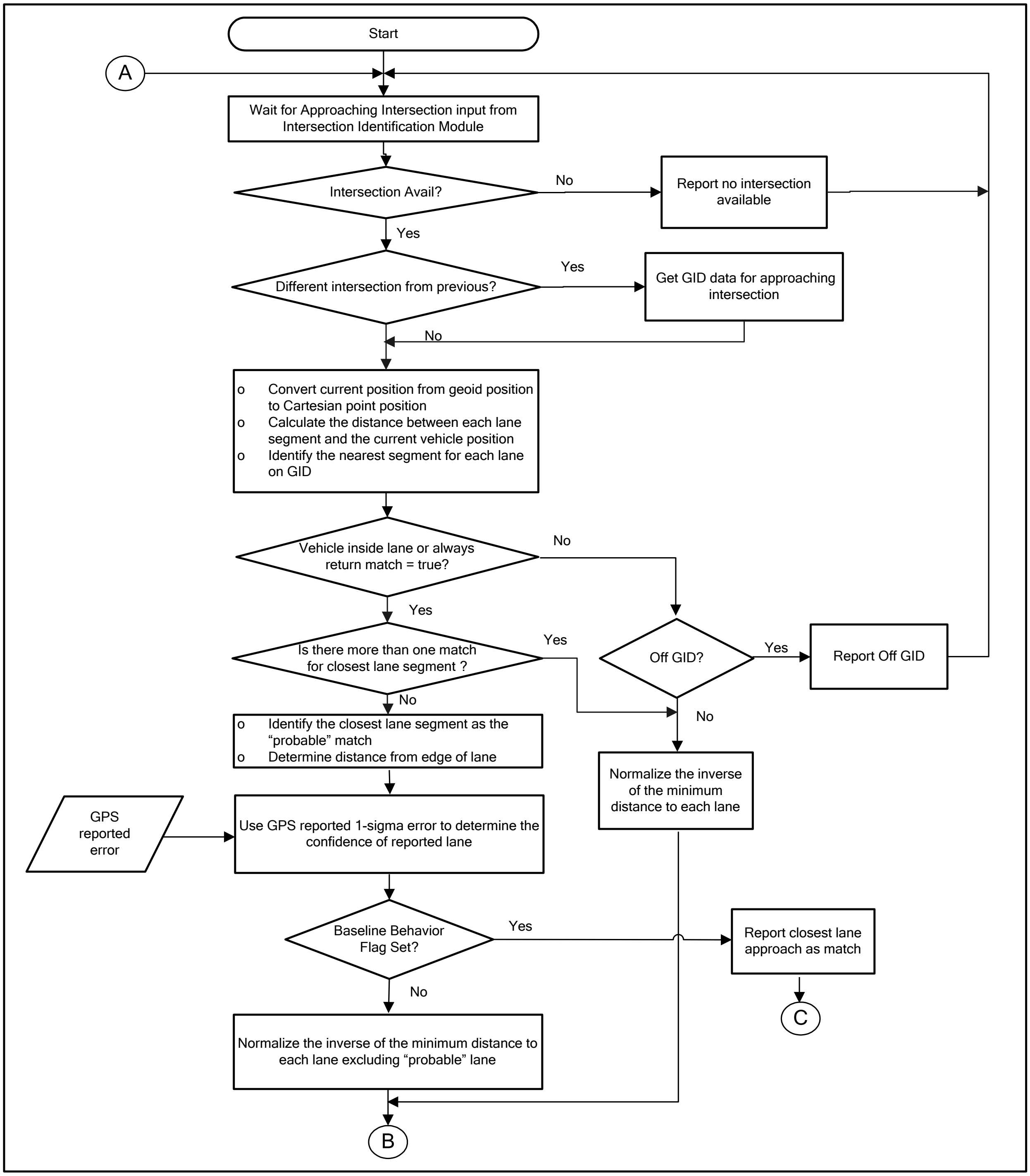

Figure 25: Map Matching/Lane Identification Module Logic Flow

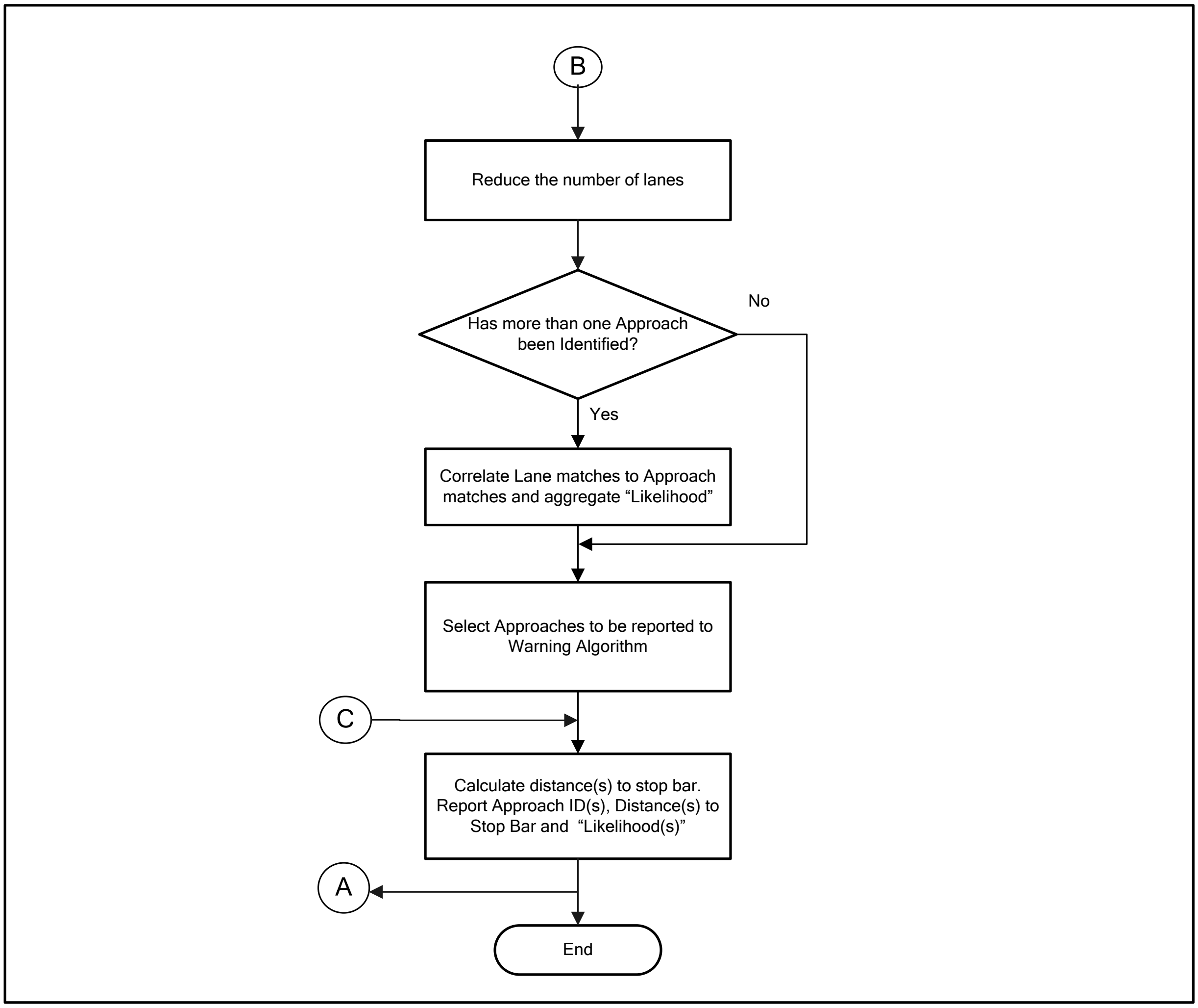

Figure 26: Map Matching/Lane Identification Module Logic Flow (continued)

A.3.1.8 Outputs

Table 29 summarizes the Map Matching and Lane Identification module outputs.

Table 29: Map Matching/Lane Identification Module Outputs

Destination	Category	Data
Warning Algorithm	Vehicle Lane Position	Approaching intersection ID (or none available) Intersection type (signalized or stop sign) Number of approaches reported For each approach: • Approach ID • Distance to the stop bar • Likelihood
GID Database Handler	GID Data request	Intersection ID
DAS Handler / Logger	Map Matching Status	Vehicle is off GID Vehicle is off lane on GID Approach validated
	Lane Matching Data	Number of approaches Present approach Approach likelihood Number of lanes Present lane Lane likelihood Intersection type Road level/lane level Distance to lane / approach center line Distance to stop bar
	Log Entries	As required by log entries (Appendix 0)

Warning Algorithm

The Warning Algorithm module detects when a red light violation or stop sign violation is likely to occur. It executes periodically, and based on information regarding the approaching intersection and the vehicle, determines the current warning status. It outputs status to the DVI Notifier indicating “Insufficient Information Available,” “Intersection Equipped, No Warning,” or “Warning.”

A.3.1.9 Warning Algorithm Requirements

The Warning Algorithm has the following prerequisites:

1. Approaching intersection information.
2. If more than one approach is possible, then information regarding all such approaches must be provided.
3. If the approaching intersection is a signalized intersection, valid (unexpired) SPaT data must also be available.

Note: The Map Matching/Lane Identification module will not identify an approaching intersection unless the intersection is within a configurable distance, and the GPS and GID data are available.

The following limitations apply to this algorithm:

1. The algorithm handles only the following signal phases defined in the SPaT. No arrow signal phases are supported. Stop sign intersections are supported:
   a. Green
   b. Yellow
   c. Red
   d. Flashing Red
2. The signal phase Countdown Timer will be considered the exact time (i.e., confidence information will be ignored).

The Warning Algorithm shall perform the following functions:

1. Execute periodically based on a configurable parameter (nominally every 100 ms).
2. Initialize the warning status to the “Insufficient Information Available.”
3. Get the current Map Matching, and CAN data/status. If the CAN data is invalid, go to step 13 below.
4. For a stop sign intersection, set the warning status to “Intersection Equipped, No Warning,” upon receiving an approaching intersection ID from the Map Matching/Lane Identification module.
5. For a signalized intersection, set the warning status to “Intersection Equipped, No Warning ,” upon receiving:
   a. An approaching intersection ID from the Map Matching/Lane Identification module.
   b. SPaT data for the approaching intersection within the configurable SPaT timeout applicable to the intersection in range indication. Note this timeout is different from and may be longer than the timeout used by the SPaT Handler to determine SPaT validity.
6. If Remote Command Message (RCMD) is enabled and the time to stop bar for the most likely approach is less than the configurable time at which to preempt, send an RCMD message to the Radio Handler/Data Demux module.
7. Check if the SPaT info is valid. If not, go to step 13 below.

8. For a signalized intersection, if no valid (unexpired) SPaT data is available, do not set the "Warning" status, but go to the last step.
9. Calculate whether the vehicle is slowing down or not, based on the following:
    a. When vehicle brake intent >= the configured minimum driver intended brake threshold OR
    b. When the vehicle speed is below the configured minimum threshold. The minimum speed threshold for traffic light and stop sign intersection are separately configurable.
10. If the vehicle is slowing down based on the above criteria, do not set the "Warning" status, but go to the last step.
11. Loop through and do the following steps from ‘a’ through ‘i,’ for each possible approach:
    a. Calculate the timeToStopBar (time taken by the vehicle to approach the intersection or stop bar assuming no change in speed or approach). Note the algorithm does not consider pedestrians, or other vehicles stopped at the intersection.
    b. Calculate timeToRed (time taken by signal to switch to red), based on the information received from the latest (not yet expired) SPaT message. If the SPaT data is not current (i.e., some SPaT updates were missed), estimate the time of the next phase based on the information from the latest SPaT. Specifically, decrement the time of the next phase in the latest SPaT message by the elapsed time since the SPaT current time. Use the signal phase, time of the next phase, and yellow duration from the SPaT message corresponding to the vehicle’s approach ID (as determined by the Map Matching/Lane Identification module).
    c. If the timeToRed is greater than or equal to the timeToStopBar, or timeToStopBar is the default maximum value (vehicle is stationary or moving backwards), then there is no warning for this approach, so exit the loop and do not process any more approaches.
    d. Obtain distToStopBar from the Map Matching module data.
    e. Check if the distToStopBar lies between the configured values for the Minimum and Maximum Warning Distances. If not, there is no warning for this approach, so exit the loop and do not process any more approaches.
    f. Obtain the distForWarn (distance it would take for vehicle to come to a complete stop). This “distance for warning” is a summation of driver brake reaction time, vehicle stopping distance, and other system delays (e.g., interface delay), and is provided in the form of a configurable array of minimum distances required to stop, indexed by speed. A separate array is used for stop sign and signalized intersections. An additional distance calculated as vehicle speed times the sum of a configurable reaction time, pre-charge delay time, and brake ramp-up time is added to this distance.
    g. If the distToStopBar is greater than or equal to distForWarn, then there is no warning for this approach, so exit the loop and do not process any more approaches.

h. Check if there is a braking exception for this case (i.e., the signal is flashing red and the lowest speed over the last configurable hysteresis time is less than a configurable hysteresis speed threshold). If so, clear the warning status and exit loop.

i. Set the status for this approach to warning should be generated.

12. If all of the approaches resulted in a warning should be generated, set the status to "Warning."
13. Output the appropriate warning status to the DVI Notifier.
14. If status is "Warning," and the Traffic Signal Violation Warning Given (TSVWG) message is enabled, format and send a TSVWG message to the Radio Handler/Data Demux module.
15. Output a message to the DAS Handler/Logger containing data required for the DAS.

A.3.1.10 Inputs

Table 30 summarizes the Warning Algorithm inputs.

Table 30: Warning Algorithm Inputs

Source	**Category**	**Data**
Configuration Parameter File	Configurable Data	Warning algorithm execution interval SPaT expiration time RCMD parameters: • RCMD message enable • Time to preempt • Preempt type Vehicle slowing parameters: • Minimum stop sign driver brake intent • Minimum signal driver brake intent • Minimum threshold speed for stop sign intersections • Minimum threshold speed for signal intersections Warning distance parameters: • Distance for Warning (array of speed versus distance) for stop sign intersections • Distance for Warning for signalized intersections • Additional reaction time • Pre-charge delay time • Brake pulse ramp-up time • Minimum warning distance • Maximum warning distance Braking exception parameters • Hysteresis speed • Hysteresis time TSVWG message enable
Vehicle Message Handler	Vehicle Status	Data valid/invalid flag Brakes active Driver intended braking level Vehicle speed
SPaT Handler	SPaT Data	Data valid/invalid flag Intersection ID Signal phase information (color) Time until next signal phase Yellow duration SPaT current time

Source	Category	Data
Map Matching	Vehicle Position	Approaching intersection ID (or none available) Intersection type (signalized or stop sign) Number of approaches reported For each approach: • Approach ID • Distance to the stop bar • Likelihood

A.3.1.11 Processing

Figure 27 illustrates the overall structure of the warning algorithm.

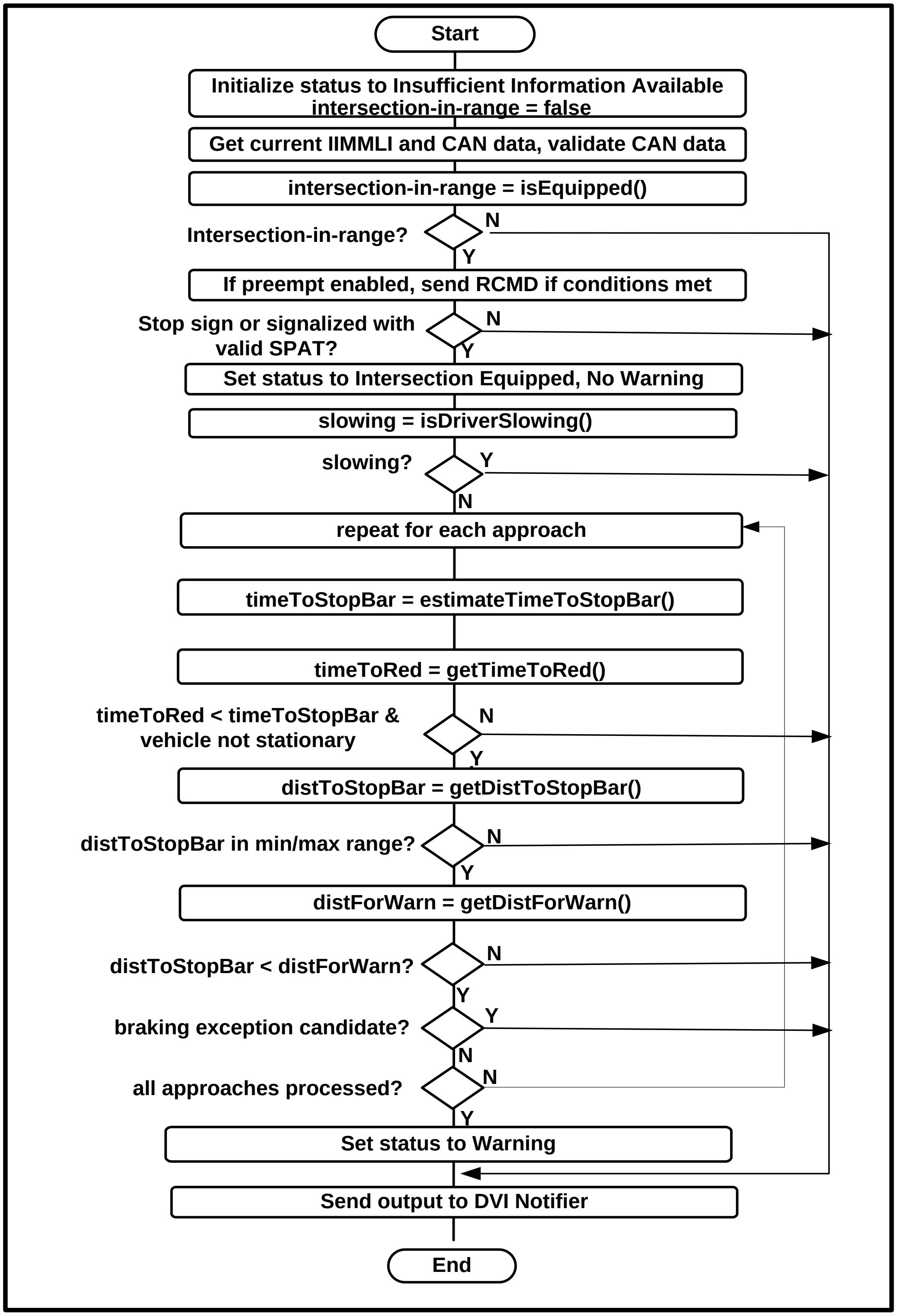

Figure 27: Warning Algorithm Logic Flow

A.3.1.12 Outputs

Table 31 summarizes the Warning Algorithm outputs.

Table 31: Warning Algorithm Outputs

Destination	Category	Data
DVI Notifier	Warning status	Insufficient Information Available, Intersection Equipped, No Warning, or Warning status Intersection type (signalized or stop sign) Brakes active status
Radio Handler/Demux	Tx Message	RCMD Message TSVWG Message
DAS Handler/Logger	DAS data	Time to StopBar Warning Algorithm Status SPaT data validity Current signal phase Time to next signal phase CAN data validity
	Log Entries	As required by log entries (Appendix 0.)

DVI Notifier

The DVI Notifier interfaces to the DVI through the WSU Vehicle/CAN Interface Services (VIS). It sets the DVI icon based on inputs from the Warning Algorithm and Error Handler to provide a visual indication to the driver of the OBE status. The DVI icon may be set to the following states:

1. Standby - No Icon
2. Maintenance - Solid Red Icon
3. Equipped - Solid Blue Icon
4. Pre-Warning - Solid Blue Icon
5. Warning - Flashing Red Icon
6. Malfunction - Solid Red Icon

The DVI Notifier also controls the flexible warning triggers (flags in a CAN message) and provides audible alerts.

A.3.1.13 Requirements

The DVI Notifier shall perform the following functions:

1. Register the DVI Notifier module with the WSU VIS to receive CAN message $702.
2. Periodically transmit CAN message $700 to the DVI at a configurable interval. Upon startup, initialize the OBE to DVI Heartbeat Sequence counter to 1, and increment the counter by 1 for each subsequent transmission of $700 to the DVI.
3. Upon receipt of the raw warning status from the Warning Algorithm, perform the following processing:

a. Upon a raw warning status change from no warning to warning, set the flexible warning trigger flags in next series of CAN $703 messages according to the configurable enable, delay, and duration for each flag. Complete the series of messages even if the raw warning status goes low prior to the sequence completion.

b. Update the DVI state machine based on the current status from the Warning Algorithm, and the Maintenance and Malfunction flags which are set or cleared by the Error Handler. Perform the state machine update in accordance with Table 32. For the warning and equipped states, use the configurable keep high and keep low durations to maintain the state for the desired duration and prevent an immediate return to the state.

c. Upon a change of state, set the DVI icon parameters for the state in the CAN $700 message after the configurable delay. If the configurable audible alert flag is enabled for the state, play the configurable audio file after the configurable delay.

d. Update the CAN $700 and $703 message parameters and output the messages via the WSU VIS.

4. Upon receipt of a CAN message $702:

   a. Check for an OBE to DVI heartbeat error indication.

   b. Check if the DVI to OBE heartbeat sequence counter matches the last number output in the $700 message. If not, set the DVI to OBE heartbeat error indication in the next $700 message.

   c. Check for a DVI system error.

   d. After transmitting a CAN message $700, if no CAN message $702 is received prior to the next periodic transmission of the $700, this will be considered a message $702 timeout.

5. Upon receipt of an OBE health status message from the DAS Handler/Logger, update the Malfunction and Maintenance flags based on the message. Use these flags in the next update of the state machine.

6. Output a message to the DAS Handler/Logger containing data required for the DAS.

Table 32: DVI Notifier State Machine

	to Standby	**to Maintenance**	**to Equipped**	**to Pre-Warning**	**to Warning**	**to Malfunctioning**
From Standby	NA	• maintenance flag goes high • equipped flag stays low • filtered warning flag stays low • malfunction flag stays low	• equipped flag goes high • filtered warning flag stays low • malfunction flag stays low	• filtered warning flag goes high • malfunction flag stays low	NA	• malfunction flag goes high
From	• maintenance flag goes low	NA	• equipped flag goes	• filtered warning	NA	• malfunction flag goes high

Maintenance	• equipped flag stays low • filtered warning flag stays low • malfunction flag stays low		high • filtered warning flag stays low • malfunction flag stays low	flag goes high • malfunction flag stays low		
From Equipped	• maintenance flag low • equipped flag goes low • filtered warning flag stays low • malfunction flag stays low	• maintenance flag high • equipped flag goes low • filtered warning flag stays low • malfunction flag stays low	NA	• filtered warning flag goes high • malfunction flag stays low	NA	• malfunction flag goes high
from Pre-Warning	• Brake active goes high • maintenance flag low • equipped flag low • filtered warning flag goes low • malfunction flag stays low	• Brake active goes high • maintenance flag high • equipped flag low • filtered warning flag goes low • malfunction flag stays low	• Brake active goes high • equipped flag high • filtered warning flag goes low • malfunction flag stays low	NA	• Brake active stays low • PRE_WARN_DURATION_SEC timer expires	• malfunction flag goes high
From Warning	• maintenance flag low • equipped flag low • filtered warning flag goes low • malfunction flag stays low	• maintenance flag high • equipped flag low • filtered warning flag goes low • malfunction flag stays low	• equipped flag high • filtered warning flag goes low • malfunction flag stays low	NA	NA	• malfunction flag goes high
From Malfunctioning	• maintenance flag low • equipped flag low • filtered warning flag low • malfunction flag goes low	NA	NA	NA	NA	NA

A.3.1.14 Inputs

Table 33 summarizes the DVI Notifier inputs.

Table 33: DVI Notifier Module Inputs

Source	Category	Data
Configuration Parameter File	Configurable data	Pre-Warning duration For the Maintenance, Equipped, Pre-Warning, Warning and Malfunction states: Icon enable Icon brightness Icon flash frequency Icon delay

Source	Category	Data
		Audio enable Audio file(s) (separate files for stop sign and signalized warning) Audio delay Warning icon keep high duration Warning icon keep low duration Equipped icon keep high duration Equipped icon keep low duration For flexible triggers 1-7: Trigger enable Trigger delay Trigger duration
Warning Algorithm	Warning status	Insufficient Information Available Intersection Equipped, No Warning, or Warning status Intersection type (signalized or stop sign) Brakes active status
DAS Handler/Logger	OBE health status	Maintenance status Malfunction status
WSU Vehicle/CAN Interface Services	CAN Message	CAN Message $702

A.3.1.15 Processing

Figure 28 illustrates the DVI Notifier logic flow.

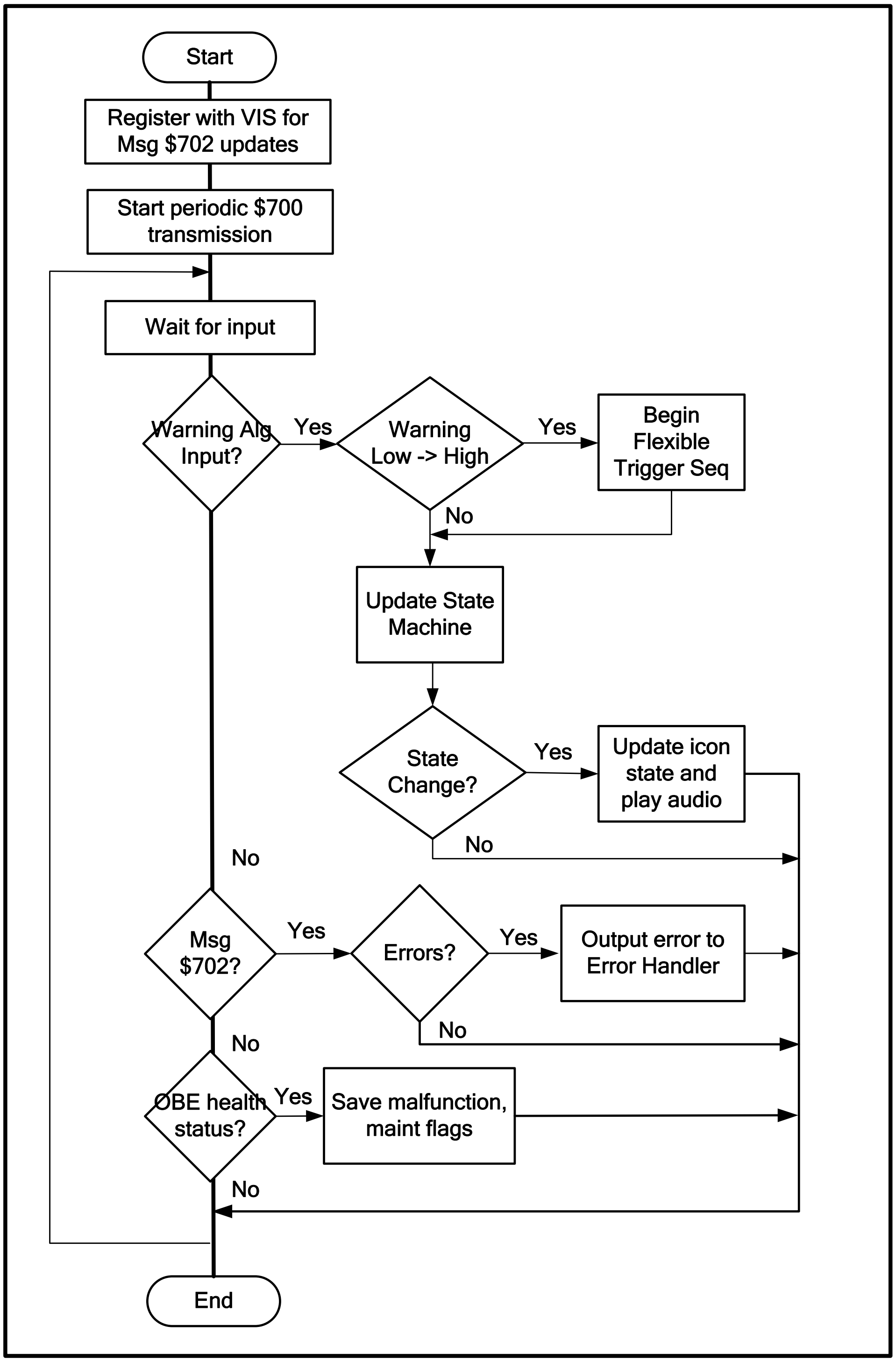

Figure 28: DVI Notifier Logic Flow

A.3.1.16 Outputs

Table 34 summarizes the DVI Notifier Outputs.

Table 34: DVI Notifier Outputs

Destination	Category	Data
WSU VIS	CAN Message $700	DVI Configuration
	CAN Message $703	Time elapsed since last warning Raw status of the Warning algorithm Flexible warning trigger flags
DAS Handler/Logger	DAS data	DVIN icon state DVI warning keep low active DVI To OBE heartbeat error DVI system error OBE to DVI heartbeat error
	Log Entries	As required by log entries (Appendix 0)

A.4 Configuration Parameters and Log File Definitions

The Software Configuration Parameter and Log File Definition tables define the user-configurable parameters and the log file entries. It supplements the CICAS-V Software Specification and CICAS-V Software Design. The tables were maintained separate from the specification and design documents because the information was expected to evolve during the software implementation and testing phases.

Configuration Parameters

The following table defines the user-configurable parameters and provides the following information:

- The process and module that uses each parameter
- The parameter name used in the cicas-v.dflt and cicas-v.conf files that were provided with the SW implementation releases
- The parameter units if applicable
- The default configured value along with the minimum and maximum values that the parameter could be configured within if the default value was to be over-ridden

Table 35: Configuration Parameters Table

#	Process	Module	Config File Name	Parameter Description	Default Value	Min Value	Max Value
1.	CWA	All	• CWADebugFlag	• Enables CWA debug output to console	0	0	1
			• CWAInfoFlag	• Enables CWA information output to the console	0	0	1
			• CWALogFlag	• Enables CWA writing of debug logs	0	0	1

#	Process	Module	Config File Name	Parameter Description	Default Value	Min Value	Max Value
2.		Intersection Identification	• IntersectionIdentificationMethod	• Intersection selection method (1=CAN vehicle speed and GPS heading, 2=GPS location calculation, 3=GPS location calculation with filter)	2	1	3
			• HDOPThreshold	• HDOP threshold for usable GPS data	3.0	1.0	10.0
			• HDOPThresholdTimeout	• Time the HDOP must be above HDOPThreshold before the GPS data is considered unusable	3.0 sec	0.0	20.0
			• GCDThreshold	• Distance threshold for candidate intersections	300 m	50	1000
			• MovingAverageTime	• Moving average duration	1.0 sec	0.0	10.0
			• IntersectionIDCalculationFilterCutOffFrequency	• GPS with filter cutoff frequency	2.68	0.01	100
			• IntersectionIDCalculationFilterSamplingTime	• GPS filter sampling time	0.10 sec	0.05	1.00
			• LowVehicleSpeedThreshold	• Low vehicle speed detection threshold	4.8 kmph	0.0	16.0
			• EnableLowVehicleSpeedFiltering	• Low vehicle speed filtering enable/disable flag	1	0	1
3.		Map Matching	• OffGIDThreshold	• Off-lane but on GID threshold	4.0 m	0	1000
			• OffGIDMethod	• Off GID method 0 = Distance from Centerline method, 1 = Distance from Lane Edge method	1	0	1
			• MapMatchSingleMatchThreshold	• Threshold for excluding all other lane matches	95 (%)	0	100
			• MapMatchMaxNumberOfLanes	• Max no. of lanes map matching will return for approach correlation	3	1	4
			• MapMatchAlwaysReturnMatch	• Map matching will always return a lane match	0	0	1
			• BaselineBehaviorFlag	• MMLI baseline behavior, reports closest APPROACH-LANE-SEGMENT match	0	0	1
			• MinApproachLikelihoodLocalCorr	• Minimum approach likelihood required when local GPS corrections are available	1	0	100

#	Process	Module	Config File Name	Parameter Description	Default Value	Min Value	Max Value
			• MinApproachLikelihood	• Minimum approach likelihood required when no local GPS corrections are available	1	0	100
			• ApprLikelihoodDiffLocalCorr	• Maximum difference between highest likelihood approach and likelihood of other approaches selected when local GPS corrections are available	15	0	100
			• ApprLikelihoodDiff	• Maximum difference between highest likelihood approach and likelihood of other approaches when no local GPS corrections are available	15	0	100
4.		Lane Identification	• MaxLaneConfidenceThreshold	• Maximum lane certainty threshold	1.6 sigma	0.0	10.0
			• AntennaOffsetNS	• Antenna offset, vehicle travel direction (same parameter as for Intersection Identification)	-2.5 m	-9.9	9.9
5.		Warning Algorithm	• WAExecInterval	• Warning algorithm execution interval	100 ms	100	100
			• SPATTimeout	• SPaT timeout applicable to the intersection-in-range-indication	400 ms	200	100000
			• MinStopSignBrakeIntent [Parameter moved to WA table files]	• Minimum driver intended braking value for the vehicle to be considered slowing at a stop sign intersection	1	0	10
			• MinSignalBrakeIntent [Parameter moved to WA table files]	• Minimum driver intended braking value for the vehicle to be considered slowing at a signalized intersection	10	0	10
			• MinStopSignSpeedThreshold [Parameter moved to WA table files]	• Minimum threshold speed for stop sign intersections	24.14 kmph	0	200
			• MinSignalSpeedThreshold [Parameter moved to WA table files]	• Minimum threshold speed for signal intersections	16.09 kmph	0	200
			• DistanceToWarn [Array moved to WA table files]	• Distance threshold for warning driver (array indexed by speed in km/h)	See .dflt file	N/A	N/A
			• MinWarnDistMeters	• Minimum warning distance meters	0 m	0	1000
			• MaxWarnDistMeters	• Maximum warning distance meters	500 m	0	1000
			• AddReactTimeinSec	• Reaction time factor seconds	0.0 s	0.0	100.00
			• PreChargeDelay	• Brake precharge delay seconds	0.0 s	0.0	10.0
			• BrakePulseRampup	• Brake Pulse ramp up time seconds	0.0 s	0.0	10.0
			• HysteresisTime	• Hysteresis time seconds	5.0 s	0.0	100.00

#	Process	Module	Config File Name	Parameter Description	Default Value	Min Value	Max Value
			• HysteresisSpeed	• Hysteresis speed kmph	5.4 kmph	0.0	180.00
			• TimeToPreempt	• Time to intersection for preempt seconds	5.0 s	0.0	100.0
			• PreemptType	• default = turn-red, 0 = clear, 1 = green	2	0	2
			• TSVWGEnable	• Enable TSVWG message	0	0	1
			• RCMDEnable	• Enable RCMD message	0	0	1
6.	DVIN	DVINotifier	• DVINDebugFlag	• Debug Flag	0	0	1
			• DVINInfoFlag	• Info Flag	0	0	1
			• DVINLogFlag	• Log Flag	0	0	1
			• HeartbeatTimeout	• Timeout for software heartbeat messages	500 ms	200	1000
			• MaintenanceAudioEnable	• Maintenance audio alerting enable	0	0	1
			• MaintenanceAudioFile	• Maintenance audio alert file	/rwflash/configs/cicas_maint.wav	N/A	N/A
			• MaintenanceAudioDelay	• Delay between entering maintenance state and playing audible alert	0.0	0.0	100.0
			• MaintenanceIconEnable	• Maintenance icon enable	1	0	1
			• MaintenanceIconBrightness	• Maintenance icon brightness	10	0	10
			• MaintenanceIconFlashFrequency	• Maintenance icon freq from 0 to 50, 255 = no flash	255	0	255
			• MaintenanceIconDelay	• Delay between entering maintenance state and setting icon	0.0 (sec)	0.0	100.0
			• MalfunctionAudioEnable	• Malfunction audio alerting enable	0	0	1
			• MalfunctionAudioFile	• Malfunction audio alert file	/rwflash/configs/cicas_malfunc.wav	N/A	N/A
			• MalfunctionAudioDelay	• Delay between entering malfunction state and playing audible alert	0.0 (sec)	0.0	100.0
			• MalfunctionIconEnable	• Malfunction icon enable	1	0	1
			• MalfunctionIconBrightness	• Malfunction icon brightness	10	0	10
			• MalfunctionIconFlashFrequency	• Maintenance icon freq from 0 to 50, 255 = no flash	255	0	255
			• MalfunctionIconDelay	• Delay between entering maintenance state and setting icon	0.0 (sec)	0.0	100.0
			• EquippedAudioEnable	• Equipped audio alerting enable	0	0	1

#	Process	Module	Config File Name	Parameter Description	Default Value	Min Value	Max Value
			• EquippedAudioFile	• Equipped audio alert file	/rwflash/configs/cicas_equipped.wav	N/A	N/A
			• EquippedAudioDelay	• Delay between entering equipped state and playing audible alert	0.0 (sec)	0.0	100.0
			• EquippedIconEnable	• Equipped icon enable	1	0	1
			• EquippedIconBrightness	• Equipped icon brightness	10	0	10
			• EquippedIconFlashFrequency	• Equipped icon freq from 0 to 50, 255 = no flash	255	0	255
			• EquippedIconDelay	• Delay between entering equipped state and setting icon	0.0 (sec)	0.0	100.0
			• EquippedIconKeepHighDuration	• Minimum time the equipped icon will stay high	5.0 (sec)	0.0	100.0
			• EquippedIconKeepLowDuration	• Minimum time the equipped icon will stay low after equipped goes false	5.0 (sec)	0.0	100.0
			• PrewarnDuration	• Duration of the prewarning state	0.0 (sec)	0.0	10.0
			• PrewarnAudioEnable	• Prewarn audio alerting enable	0	0	1
			• PrewarnAudioFile	• Prewarn audio alert file	/rwflash/configs/cicas_prewarn.wav	N/A	N/A
			• PrewarnAudioDelay	• Delay between entering prewarn state and playing audible alert	0.0 (sec)	0.0	100.0
			• PrewarnIconEnable	• Prewarn icon enable	1	0	1
			• PrewarnIconBrightness	• Prewarn icon brightness	10	0	10
			• PrewarnIconFlashFrequency	• Prewarn icon freq from 0 to 50, 255 = no flash	255	0	255
			• PrewarnIconDelay	• Delay between entering prewarn state and setting icon	0.0 (seconds)	0.0	100.0
			• WarningAudio_enable	• Warning audio alerting enable	0	0	1
			• WarningSignalizedAudioFile	• Warning audio alert file for signalized intersections	/rwflash/configs/stoplight.wav	N/A	N/A
			• WarningStopsignAudioFile	• Warning audio alert file for stop sign intersections	/rwflash/configs/stopsign.wav	N/A	N/A

#	Process	Module	Config File Name	Parameter Description	Default Value	Min Value	Max Value
			• WarningAudioDelay	• Delay between entering warning state and playing audible alert	0.0 (sec)	0.0	100.0
			• WarningIconEnable	• Warning icon enable	1	0	1
			• WarningIconBrightness	• Warning icon brightness	10	0	10
			• WarningIconFlashFrequency	• Warning icon freq from 0 to 50, 255 = no flash	255	0	255
			• WarningIconDelay	• Delay between entering warning state and setting icon	0.0 (sec)	0.0	100.0
			• WarningIconKeepHighDuration	• Minimum time the warning icon will stay high	5.0 (sec)	0.0	100.0
			• WarningIconKeepLowDuration	• Minimum time the warning icon will stay low after a warning	30.0 (sec	0.0	100.0
			• FlexWarnTrig1Enable	• Enable for flex warning trigger flag 1	0	0	1
			• FlexWarnTrig1Offset	• Offset between raw warning and flag 1 going high	0.0 (sec)	0.0	10.0
			• FlexWarnTrig1Duration	• Duration the flex 1 flag will stay high	0.0 (sec)	0.0	100.0
			• FlexWarnTrig2Enable	• Enable for flex warning trigger flag 2	0	0	1
			• FlexWarnTrig2Offset	• Offset between raw warning and flag 2 going high	0.0 (sec)	0.0	10.0
			• FlexWarnTrig2Duration	• Duration the flex 2 flag will stay high	0.0 (sec)	0.0	100.0
			• FlexWarnTrig3Enable	• Enable for flex warning trigger flag 3	0	0	1
			• FlexWarnTrig3Offset	• Offset between raw warning and flag 3 going high	0.0 (sec)	0.0	10.0
			• FlexWarnTrig3Duration	• Duration the flex 3 flag will stay high	0.0 (sec)	0.0	100.0
			• FlexWarnTrig4Enable	• Enable for flex warning trigger flag 4	0	0	1
			• FlexWarnTrig4Offset	• Offset between raw warning and flag 4 going high	0.0 (sec)	0.0	10.0
			• FlexWarnTrig4Duration	• Duration the flex 4 flag will stay high	0.0 (sec)	0.0	100.0
			• FlexWarnTrig5Enable	• Enable for flex warning trigger flag 5	0	0	1
			• FlexWarnTrig5Offset	• Offset between raw warning and flag 5 going high	0.0 (sec)	0.0	10.0
			• FlexWarnTrig5Duration	• Duration the flex 5 flag will stay high	0.0 (sec)	0.0	100.0
			• FlexWarnTrig6Enable	• Enable for flex warning trigger flag 6	0	0	1
			• FlexWarnTrig6Offset	• Offset between raw warning and flag 6 going high	0.0 (sec)	0.0	10.0

#	Process	Module	Config File Name	Parameter Description	Default Value	Min Value	Max Value
			• FlexWarnTrig6Duration	• Duration the flex 6 flag will stay high	0.0 (sec)	0.0	100.0
			• FlexWarnTrig7Enable	• Enable for flex warning trigger flag 7	0	0	1
			• FlexWarnTrig7Offset	• Offset between raw warning and flag 7 going high	0.0 (sec)	0.0	10.0
			• FlexWarnTrig7Duration	• Duration the flex 7 flag will stay high	0.0 (sec)	0.0	100.0
7.	CDIP	All	• CDIDebugFlag	• Enables CDI debug output to console	0	0	1
			• CDIInfoFlag	• Enables CDI information output to the console	0	0	1
			• CDILogFlag	• Enables CDI writing of debug logs	0	0	1
8.		Radio Handler/Data Demux	• VIICRSEFlag	• Flag to indicate operation with VIIC RSE that adds two extra bytes to WSM header	0	0	1
			• GIDPSID	• PSID for GID WSMs	0x01E00002	0x00000000	0xFFFFFFFF
			• SPATPSID	• PSID for SPaT WSMs	0x01E00001	0x00000000	0xFFFFFFFF
			• GPSCPSID	• PSID for GPSC WSMs	0x01E00003	0x00000000	0xFFFFFFFF
			• TSVWGPSID	• PSID for TSVWG WSMs	0x03E00001	0x00000000	0xFFFFFFFF
			• TSVWGPower	• Power for TSVWG WSMs	20 (dBm)	10	20
			• TSVWGDataRate	• Data rate for TSVWG WSMs	6 (Mbps)	3	27
			• TSVWGPriority	• Priority for TSVWG WSMs	1	0	3
			• TSVWGRepeatCount	• Number of times a TSVWG WSM will be transmitted [not supported – requirement was removed]	1	1	5
			• TSVWGRepeatInterval	• Interval between repeated transmissions of the TSVWG WSM [not supported – requirement was removed]	10 (ms)	5	20
			• RCMDPSID	• PSID for RCMD WSMs	0x07E00001	0x00000000	0xFFFFFFFF
			• RCMDPower	• Power for RCMD WSMs	20 (dBm)	10	20
			• RCMDDataRate	• Data rate for RCMD WSMs	6 (Mbps)	3	27
			• RCMDPriority	• Priority for RCMD WSMs	3	0	3
			• RCMDRepeatCount	• Number of times a RCMD WSM will be transmitted [not supported – requirement was removed]	1	1	5
			• RCMDRepeatInterval	• Interval between repeated transmissions of the RCMD WSM [not supported – requirement was removed]	10 (ms)	5	20

#	Process	Module	Config File Name	Parameter Description	Default Value	Min Value	Max Value
			• RadioStatisticsPollingInterval	• Polling interval for radio statistics	1000 (ms)	1000	1000
9.		GID DB Handler	• GIDExpirationTime • GIDStorageAllocation	• GID expiration period • GID storage allocation	30 (days) 1024 (Kbytes)	1 30	180 1024
10.		SPaT Handler	• SPATExpirationTime • SPATAgeWarnThreshold	• SPaT expiration time • Threshold for SPaT age warning	400 (ms) 400 (ms)	200 200	1000 10000
11.	CVLP	Vehicle Location	• CVLDebugFlag • CVLInfoFlag • CVLLogFlag	• Enables CVL debug output to console • Enables CVL information output to the console • Enables CVL writing of debug logs	0 0 0	0 0 0	1 1 1
12.		GPS Handler	• GPSExpirationTime • GPSSolLostLTTime	• GPS expiration period • Length of time short-term (solution lost) error (invalid TPS frame) needs to be outstanding before it becomes long-term	400 (ms) 10 (minutes)	200 1	1000 60
13.		Log flags to enable sending events to CLOGP	• CVLSupGpscFmtsLogFlag • CVLGpsDataRxLogFlag • CVLGpsDataTmoutLogFlag • CVLGpsDataTmoutNoDataLogFlag	• • • No-fix or short-term solution lost • Timeout – no data at all	0 0 0 0	0 0 0 0	1 1 1 1
			• CVLGpscRxLogFlag • CVLGpscRxDataLogFlag • CVLGpsStatusChLogFlag • CVLGpscRtcmCksmFailLogFlag • CVLGpscRtcmCksmFailDataLogFlag • CVLGpsSolLostLTLogFlag	• • • • • • Short-term solution lost persisting for a while, becomes long-term	0 0 0 0 0 0	0 0 0 0 0 0	1 1 1 1 1 1
14.	CVIP	Vehicle Information	• CVIDebugFlag • CVIInfoFlag • CVILogFlag	• Enables CVI debug output to console • Enables CVI information output to the console • Enables CVI writing of debug logs	0 0 0	0 0 0	1 1 1
15.		Vehicle Message Handler	• CANExpirationTime • CAN704TxInterval	• CAN expiration period • Interval at which 704 message is sent to the Netway box	400 (ms) 100 (ms)	200 100	1000 200
16.		Log-enable flags	• CANDataRxLogFlag	•	0	0	1
			• IncompCANDataRxLogFlag • CANExpdLogFlag • CAN606ExpdLogFlag	• • One or more of $600-605 not received • $606 not received	0 0 0	0 0 0	1 1 1

#	Process	Module	Config File Name	Parameter Description	Default Value	Min Value	Max Value
			• CAN606HBErrLogFlag	• Netway reports OBE-Nw Heartbeat Error	0	0	1
			• CAN606SeqErrLogFlag	• OBE detecting mismatch in sequence no.s	0	0	1
			• CANErrLogFlag	•	0	0	1
			• VehCANDataTmoLogFlag	• Netway reports timeout between it and vehicle	0	0	1
17.	CLOGP	CLOG	• CLOGDebugFlag	• Enables CLOG debug output to console	0	0	1
			• CLOGInfoFlag	• Enables CLOG information output to the console	0	0	1
			• CLOGLogFlag	• Enables CLOG writing of debug logs	0	0	1
18.		DAS Handler/Log ger	• GlobalLogFlag	• Global enable/disable	1	0	1
			• DASLogInterval	• Controls the logging rate for DAS messages (log will be written every nth time data is output to the DAS)	1	1	100
			• CLOGDasMsgsLogFlag	• Log Enable Flag for this field	1	0	1
19.		Error Handler	• EnableHWWatchdog	• Allows turning off HW watchdog while debugging	0	0	1
			• EnableHBLogging	• Enables logging of software heartbeat messages	0	0	1
			• HWWatchdogTimeout	• Timeout after which HW watchdog will reset if SW heartbeats are not received from all processes	10000 (ms)	1000	30000
			• SWWatchdogTimeout	• Timeout for SW watchdog timing	1000 (ms)	500	10000
20.		Events/Error s Affecting Health State	• InitErrTypeNN, where NN=00..75 (described below in two groups)	• How to initiate an event into error (0 = normal, 1 = maintenance, 2 = malfunction)	0	0	2

#	Process	Module	Config File Name	Parameter Description	Default Value	Min Value	Max Value
			• Maintenance errors • InitErrType56 • InitErrType57 • InitErrType58 • InitErrType59 • InitErrType62 • InitErrType63 • InitErrType64 • InitErrType65 • InitErrType66	• Default level is 1 • DAS System Error • DAS Boot Up Error • DAS Shutdown Error • OBE to DAS Heartbeat Error • DAS Video Health Status • DAS Radar Health Status • DAS Hard-drive Space Low • DAS Battery Voltage Low • DAS Heartbeat Sequence Mismatch	1	0	2
			• Malfunction errors • InitErrType27 • InitErrType34 • InitErrType35 • InitErrType36 • InitErrType37 • InitErrType68 • InitErrType69 • InitErrType71 • InitErrType73 • InitErrType74 • InitErrType75	• Default level is 2 • GPS Tmout (No Data) • CAN 600-605 Data Timeout • OBE-Netway HB Error • Netway-OBE HB Error / Seq Mismatch • CAN 606 Data Timeout • DVI to OBE HB Error • OBE to DVI HB Error • DVI Sys Error • DVI OBE Timeout • Vehicle CAN Data Timeout • GPS Solution Lost Long-Term	2	0	2
		Health Maintenance Thresholds	• MalfPersThresh	• Length of time (msec) “malfunction” error persists before affecting health state	1000	100	10000
			• MaintPersThresh	• Length of time (msec) “maintenance” error persists before affecting health state	1000	100	10000
			• LowHDThresh	• Hard-drive space (in GigaBytes) falling below this limit triggers error 64	10	1	1000
			• LowVoltThresh	• Low battery voltage limit that triggers error 65	9.5	6	14
		Error Flags for bits in CAN message	• DASSysErrLogFlag	• 0[7] = byte 0, bit 7	0	0	1
			• DASBootupErrLogFlag	• 0[6]	0	0	1
			• DASShutdownErrLogFlag	• 0[5]	0	0	1
			• DASOBEDASHbErrLogFlag	• 0[4]	0	0	1

#	Process	Module	Config File Name	Parameter Description	Default Value	Min Value	Max Value
		$701 from DAS	• DASRunModeLogFlag • DASShutdownModeLogFlag • DASVideoHSLogFlag • DASRadarHSLogFlag • DASHDSpaceLogFlag • DASBatVoltLogFlag • DASHbMismatchLogFlag	• 1[7] • 1[6] • 2[7..4] • 2[3..0] • 3[7..0] (x400 to get space in MB) • 4[7..0] (divide by 10 to get voltage) • Sequence number in 5[7..0], 6[7..0], 7[7..0]	0 0 0 0 0 0 0	0 0 0 0 0 0 0	1 1 1 1 1 1 1

Log File Definitions

The following table defines the log file entries. For each entry the process and module that generates the entry is listed along with the following information:

- Type – Info, Event, or Error
  - Information (Info) entries are written at startup, and record configuration parameters and software capabilities.
  - Event entries are triggered by conditions that may occur during normal operation, and record the parameters associated with the event.
  - Error entries are triggered by conditions that should not occur in normal operation, and record the parameters associated with the error.
- Configuration File Name – identifies the name used in the configuration file. The name shown will be appended by the string "LogFlag" in the configuration file. For example if "CDIWsaRx" is shown in the table, the configuration file name will be "CDIWsaRxLogFlag."
- Parameters – lists the data recorded in the log entry.
- Log Trigger – describes what causes the log entry to be written.

#	Process	Module	Type	Configuration File Name	Parameters	Log trigger
1.	All	All	Info	Conf Params (Logged if GlobalLogFlag=1)	All parameters	Startup
2.	CWA	All	Event	SPaT Data Expd	N/A	Receipt of SPaT expiration from CDIP
3.				GPS Data Expd	N/A	Receipt of GPS expiration from CVLP
4.				CAN Data Expd	N/A	Receipt of CAN expiration from CVIP

#	Process	Module	Type	Configuration File Name	Parameters	Log trigger
5.		Intersection Identification	Event	HDOP Status Chng	OldVal NewVal	Change of status from good to bad or vice versa
6.			Event	New Apprng Intn	Intn Type DistToIntnRefPt RtofChngMovingAvg GPSCorrectnReqd	Change of approaching intersection ID (or change to none available)
7.			Event	ROC method results	GPS CANSpeedHeading GPSFilter	Calculations when an intersection is within GCD threshold
8.		Map matching/ Lane Identification	Event	Map Match Rslts Succ	Intn LaneMatchState ProbLane ProbLaneConf LikelyAppr LikelyApprConf DistToStopBar DistToNearestLaneEdge	Each execution where a lane was identified
9.			Event	Map Match Unsucc	Intn LaneMatchState DistFromClstLane LaneConf	Each execution where a lane was not identified
10.			Event	Veh Posn	X Y	Each execution
11.			Event	Intn/Appr Chng	IntnID ApproachID	Upon change of intersection or approach
12.			Event	Top Approach Matches	Approach Conf Approach Conf Approach Conf	

#	Process	Module	Type	Configuration File Name	Parameters	Log trigger
13.			Event	Top Lane Matches	Lane Approach LaneConf Lane Approach LaneConf Lane Approach LaneConf Lane Approach LaneConf Lane Approach LaneConf	
14.		Warning algorithm	Event	Warn Alg Rslts (Intn In Rng)	SPATSignPhase SPATCntdnTm VehSlowing TmToStopBar TmToRed DistToStop WarnGenFlag	Upon each execution where the warning algorithm ran to completion
15.			Event	Warn Alg Rslts (No Intn In Rng)	IIMDataAvail MM/LIMDataAvail IntnInRngSPAT TmOut	Upon each execution where the warning algorithm could not run to completion
16.			Event	SPaT Intn-In-Rng TmOut	TmSinceLastSPATUpd	Upon expiration of SPaT intersection-in-range timeout
17.		Warning State Machine	Event	Warn State Chng	WarnState	Initially and upon warning state change
18.	CDIP	Radio Handler/ Data Demux	Info	Supp TOM Formats	Fmt	Startup
19.			Event	Radio Stats	RSSI GoodRxFrames RxErrs	Periodically when statistics are polled

#	Process	Module	Type	Configuration File Name	Parameters	Log trigger
20.			Event	WSA Rx	SrcMAC PSID PSC SCH	WSA received
21.			Event	New/Mod WSA Rx	SrcMAC PSID PSC SCH	First WSA received per RSE and upon WSA content change
22.			Event	WBSS Joined	SrcMAC SCH	Each occurrence
23.			Event	WBSS Left	SrcMAC SCH	Each occurrence
24.			Error	TOM Hdr Chk Failed	Chk	Each occurrence
25.			Error	WSA TOM Hdr Chk Failed	Chk	Each occurrence
26.			Error	WSA GPSC Status Chk Failed	IntnID Status	
27.		GID DB Handler	Info	Supp GID Fmts	Fmt1 Fmt2	Startup
28.			Event	Exp GID Rec Del	Intn Fmt Cont LoadTm Data	Startup
29.			Info	Startup GID DB Conts	Intn Fmt Cont Data	Startup
30.			Event	GID Rx	Intn Fmt Cont LoadTm Data	GIDs received
31.			Event	New GID Rx	Intn Fmt Cont LoadTm Data	GID received for which no record exists in DB

#	Process	Module	Type	Configuration File Name	Parameters	Log trigger
32.			Event	Upd GID Rx	Intn OldFmt NewFmt OldCont NewCont LoadTm Data	GID received for which format or content version is different than DB
33.			Event	Unexp GID Del	Intn Fmt Cont LoadTm Data	Record deleted from DB due to space constraints
34.		SPaT Handler	Info	Supp SPaT Fmts	Fmt1 Fmt2	Startup
35.			Event	SPaT Rx	Intn Dat	SPaT message received
36.			Event	SPaT Rx Apprng	Intn Dat	SPaT message received for approaching intersection
37.			Event	Dup SPaT Rx Apprng	Intn Cont Dat	Duplicate SPaT message received (i.e., content version changed from previous SPaT)
38.			Event	1st SPaT Rx Apprng	Intn Dat	First SPaT received since intersection was selected or since previous SPaT data for the intersection expired.
39.			Event	SPaT Val Exp	Intn	No SPaT has been received for the approaching intersection for the expiration period
40.		GID DB Handler or SPaT Handler	Event	Metric Rx	Year Month Day Hour Minute Millisec MsgCounter	Receipt of metric object
41.	CVLP	GPS Handler	Info	Sup GPSC Fmts	Fmt1 Fmt2 Fmt3	Startup

#	Process	Module	Type	Configuration File Name	Parameters	Log trigger
42.			Event	GPS Data Rx	Valid Time Date Latitude Longitude Altitude Groundspeed Course Hdop Pdop Lat_error Lon_error Gps_msec Diff_age Sol_age Fixquality Numsats Gps_week	Receipt of valid GPS data
43.			Event	GPS TmOut (No Fix)	N/A	No valid data received from the GPS receiver for the expiration period due to lack of receiver fix
44.			Error	GPS Tmout (No Data)	N/A	No data received from the GPS receiver for the expiration period
45.			Event	GPSC Rx Hdr	Status WkNum MsInWk Length	GPSC message received
46.			Event	GPSC GPS Status Chng	Status	First GPSC status and upon status change
47.			Error	GPSC RTCM Chksum Failure	MsgID	Checksum failure
48.			Error	GPS Solution Lost Long-Term	N/A	Prolonged short-term solution lost
49.			Event	GPSC Rx Data	1005Data 1001Data	Hex representation of GPSC data received
50.			Info	GPSC RTCM Chksum Fail Data	Data	Hex of failed data
51.	CVIP	Vehicle Message Handler	Event	CAN Data Rx	N/A	All expected CAN messages have been received

#	Process	Module	Type	Configuration File Name	Parameters	Log trigger
52.			Error	Incomp CAN Data Rx	MsgId1 MsgId2 MsgId3 MsgId4 MsgId5	One or more of the expected CAN messages was not received
53.			Error	CAN Expd	N/A	No CAN data has been received for the expiration period.
54.			Error	CAN 606 Expd	N/A	No CAN message 606 received
55.			Error	CAN 606 HB Error	N/A	OBE->Netway HB error flag set in CAN $606
56.			Error	CAN 606 SEQ Error	RxSeq TxSeq	Incorrect sequence number received in CAN $606
57.			Error	CAN Error	ErrorCode	WSU VIS detected CAN bus error
58.			Error	Vehicle CAN Data Timeout	N/A	Netway reports timeout in vehicle CAN bus
59.	DVI	Notifies of Health to Driver Interface	Error	DVIOBEHbError	N/A	Sequence number in $702 received does not match OBE's
60.			Error	OBEDVIHbError	N/A	Perceived error by DVI indicated in $702
61.			Error	DVISysError	N/A	System error reported by DVI in $702
62.			Error	DVIOBETimeout	N/A	An expected periodic $702 was not received from DVI
63.	CLOGP	DAS Handler/Logger	Event	DAS Msgs	DAS message contents	Upon output of messages to DAS
64.			Error	DAS Error Indication	DASSystemError DASBootupError DASShutdownError OBEDASHbError DASRunMode DASShutdownMode DASHDSpaceLow DASBatVoltLow DASHbSeqMismatch	 DAS reports perceived heartbeat error Hard-drive space running low Battery voltage running low Sequence number does not match that received in $701
65.			Info	DAS Video Health Status	Q-Error	
66.			Info	DAS Radar Health Status	FrontRearError	
67.			Event	Watchdog Heartbeat	WDHeartbeatSource	Keepalives from other modules

OBE SW Design Details

Figure 29 illustrates the CICAS-V OBE Software Design. The modules shown in the architecture diagram have been grouped into five processes for implementation. In addition, a CICAS-V main process parses the configuration file and starts up the other processes. Its functionality is limited and will not be described further in this document. Figure 29 also illustrates the primary data flows.

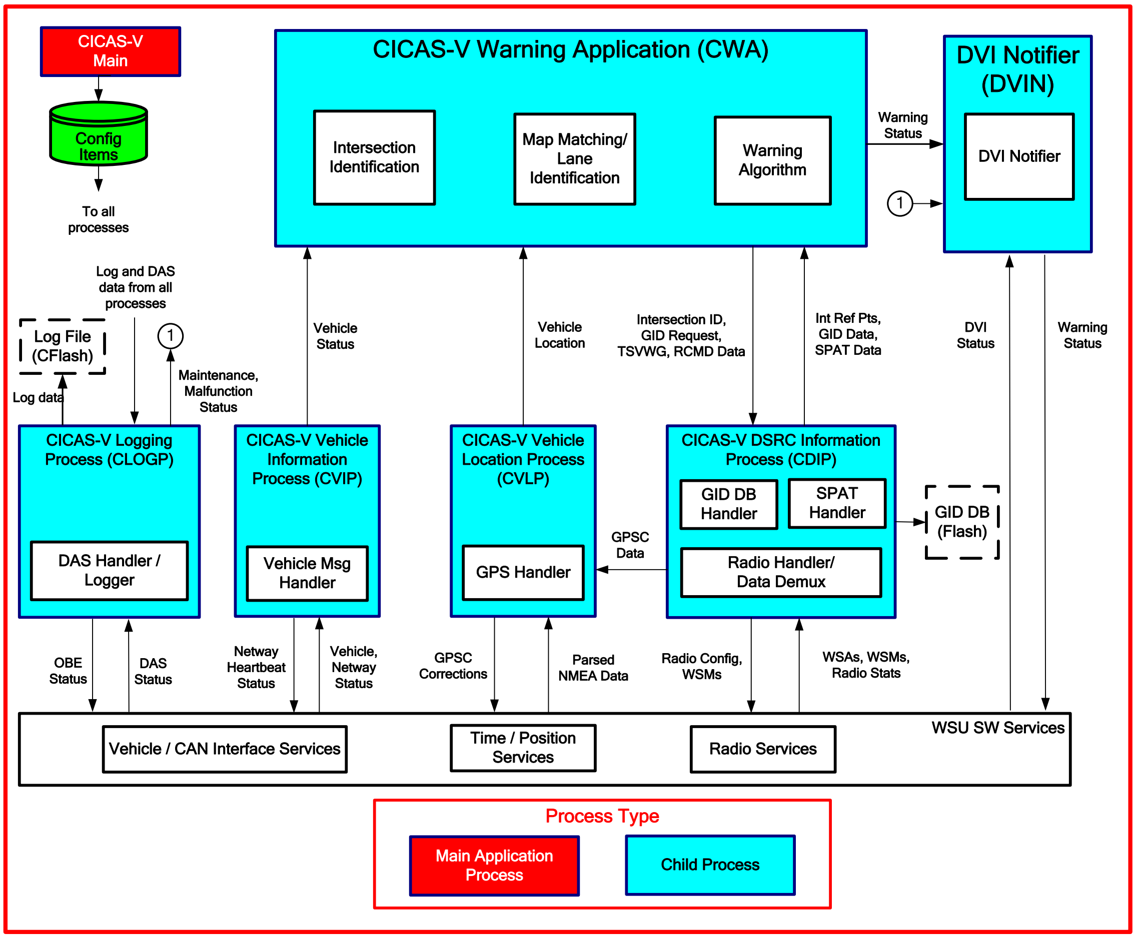

Figure 29: CICAS OBE Software Design

The WSU software may be configured to start the CICAS-V application manually or automatically at power up. When the application is started, this initiates the CICAS-V Main process which parses a configuration file and then starts each of the six OBE child processes. When each child process is started, the creation of a main child process thread is initiated. Each child process then creates additional threads if necessary.

Upon application termination or WSU shutdown, the OS sends a termination signal to each process followed by a kill signal. Each process performs any pre-termination processing that is required (e.g., writing data to flash) upon receiving the termination signal. The kill signal ends the process execution.

The following sections discuss the design details for each of the Child Processes shown in Figure 29.

A.5 CICAS-V DSRC Information Process (CDIP)

Overview

The CICAS-V DSRC Information Process (CDIP) performs the following functions:

1. Configure the WSU WAVE radio through the WSU Radio Services (RS) and register as a User of the CICAS-V services.
2. Interface to the WSU RS to receive radio statistics, WSA indications, and WSMs.
3. Periodically poll for radio statistics at a configurable rate and output selected statistics to the CLOGP for logging.
4. Upon receiving a WSA indication, determine if the CICAS-V WAVE Basic Service Set (WBSS) should be joined.
5. Upon receiving a WSM:
    a. Check if the WSM contains a valid TOM with GPSC, SPaT, or GID. Appendix A.29 – 'WSM OTA Message Definitions' defines the format and content of the TOMs.
    b. Send GPSC TOMs data to the CVLP.
    c. Parse SPaT TOMs. If the SPaT is for the approaching intersection as identified by the CWA, send the required data to CWA. If not, discard the data. Also detect when SPaT messages have been missed and the previous data has expired.
    d. Parse GID TOMs and maintain the GID database in flash memory.
6. Interface to the CWA to receive the approaching intersection ID, and send the GID intersection reference points and requested GID data.
7. Compose and transmit Traffic Signal Violation Warning Given (TSVWG) and Remote Command (RCMD) WSMs upon request from the CWA.
8. Periodically output a message to CLOGP containing data required for the DAS.

Interfaces

Figure 30 illustrates the CDIP interfaces with the WSU RS and other CICAS-V processes.

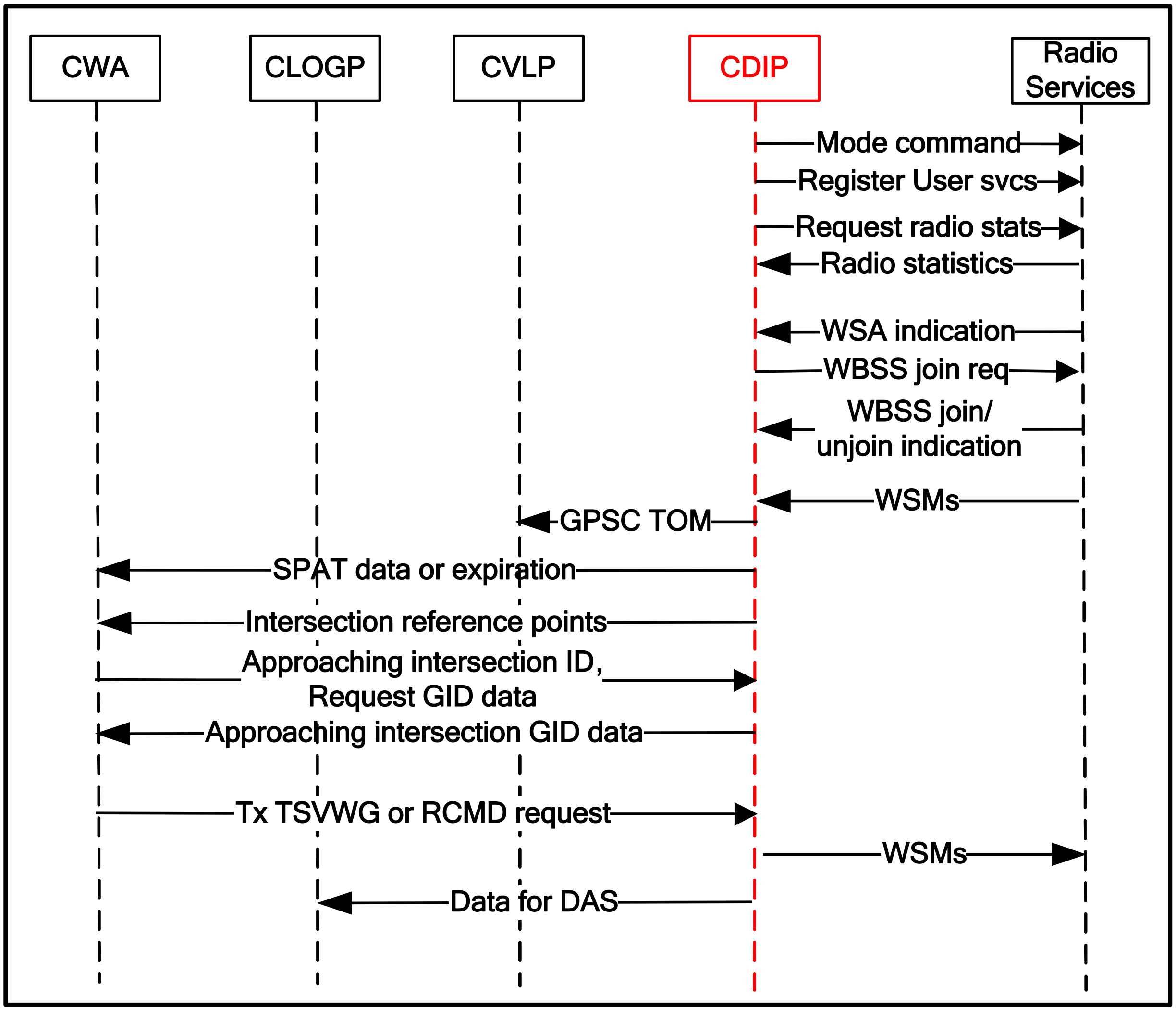

Figure 30: CDIP Interface Diagram

Process Structure

Figure 31 illustrates the CDIP thread decomposition and data flow. The CDIP Main thread configures the radio and registers the application with the WSU RS. The application registration request identifies the CICAS-V Provider Service Identifiers (PSIDs) and two callback functions (WSA and WSM). The main thread also registers for WBSS join/un-join indications and identifies the WBSS callback function.

The CDIP Main thread creates a Tx WSM thread, which processes TSVWG and RCMD Tx requests from CWA. It also sets a periodic timer, which causes a timer thread to execute upon expiration. After this processing is complete, the Main thread waits for inputs from the CWA. It receives the approaching intersection ID and GID requests from the CWA and responds with the requested GID data.

The RS calls the WSA callback function (which executes as part of an RS API thread) when it receives a WSA for a CICAS-V service (identified by the PSID). CDIP determines if the WBSS should be joined, and if so, sends a join request to RS.

The RS calls the WSM callback function (which executes as part of an RS API thread) when it receives a WSM for a CICAS-V service. The callback function validates and processes the TOM. It outputs GPSC data to the CVLP, SPaT data to the CWA, and maintains the GID database. It also outputs the list of GID intersection reference points to the CWA whenever the list changes.

The RS calls the WBSS callback function (which executes as part of an RS API thread) to confirm a WBSS join request and notify CDIP of WBSS un-joins (due to subsequent join request or WBSS timeout due to not receiving any data).

The timer thread handles polling the radio statistics, detecting a SPaT timeout, and periodically sending DAS data to CLOGP. The SPaT timeout occurs when no new SPaT data has been received for the current intersection for a configurable period of time and the previous SPaT data has expired. When this occurs, CDIP sends a SPaT expiration notification to the CWA. The DAS data includes CDIP status and error indications.

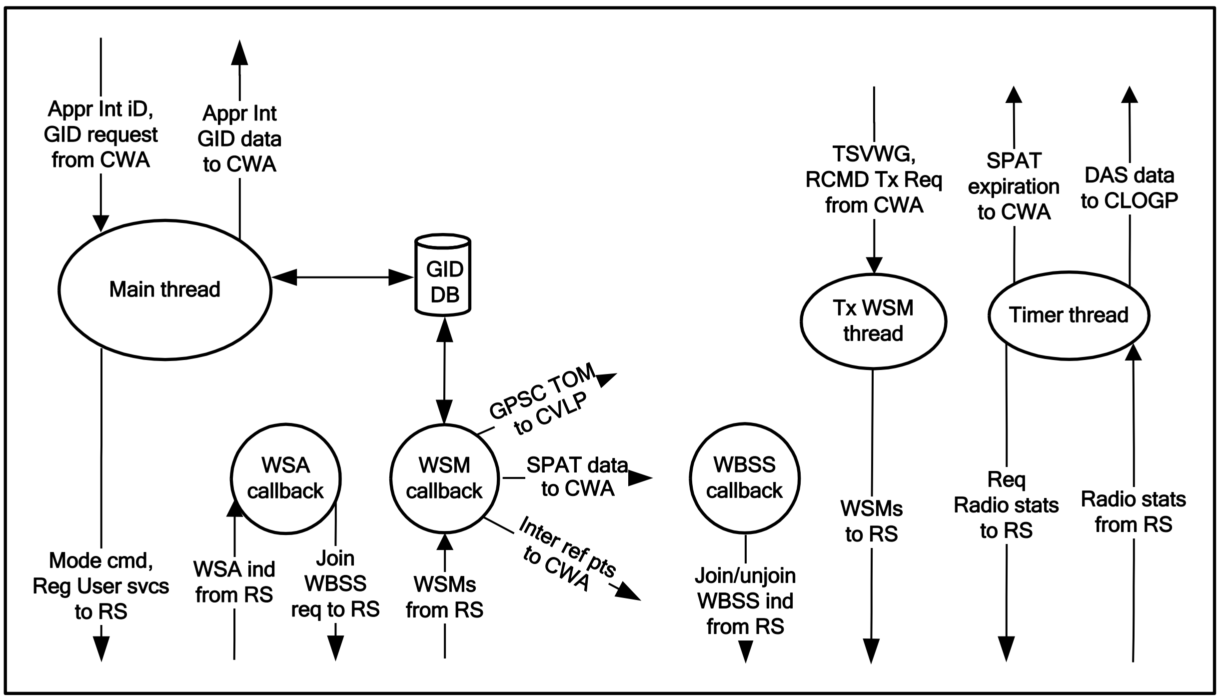

Figure 31: CDIP Thread Decomposition

Functional Description

The following subparagraphs describe the processing for each of the threads.

A.5.1.1 CDIP Main Thread

The CDIP Main thread performs the following processing at initialization.

1. Configure the WSU WAVE Radio to WAVE mode to support CICAS-V operation.
2. Register with the WSU Radio Services as a User of the CICAS-V service. The registration request includes the CICAS-V PSIDs and the WSA and WSM callback function pointers.

3. Register for WBSS join/un-join indications and identify the WBSS callback function.
4. Load the GID database into RAM. If any records have expired, delete the expired records and update the database in flash memory. If the current database size exceeds the configurable storage limit (i.e., the limit was reduced since the previous execution), delete the oldest records to meet the storage limit.
5. Create sockets for communication with other processes.
6. Create a Tx WSM thread which processes TSVWG and RCMD Tx requests from CWA.
7. Send the current list of intersection reference points to the CWA. If no GID data exists, the list will be of zero length.
8. Starts the periodic timer to collect radio statistics, check SPaT expiry, and to send DAS data to CLOGP.
9. Service CWA inputs.

Figure 32 illustrates the CDIP Main thread initialization processing.

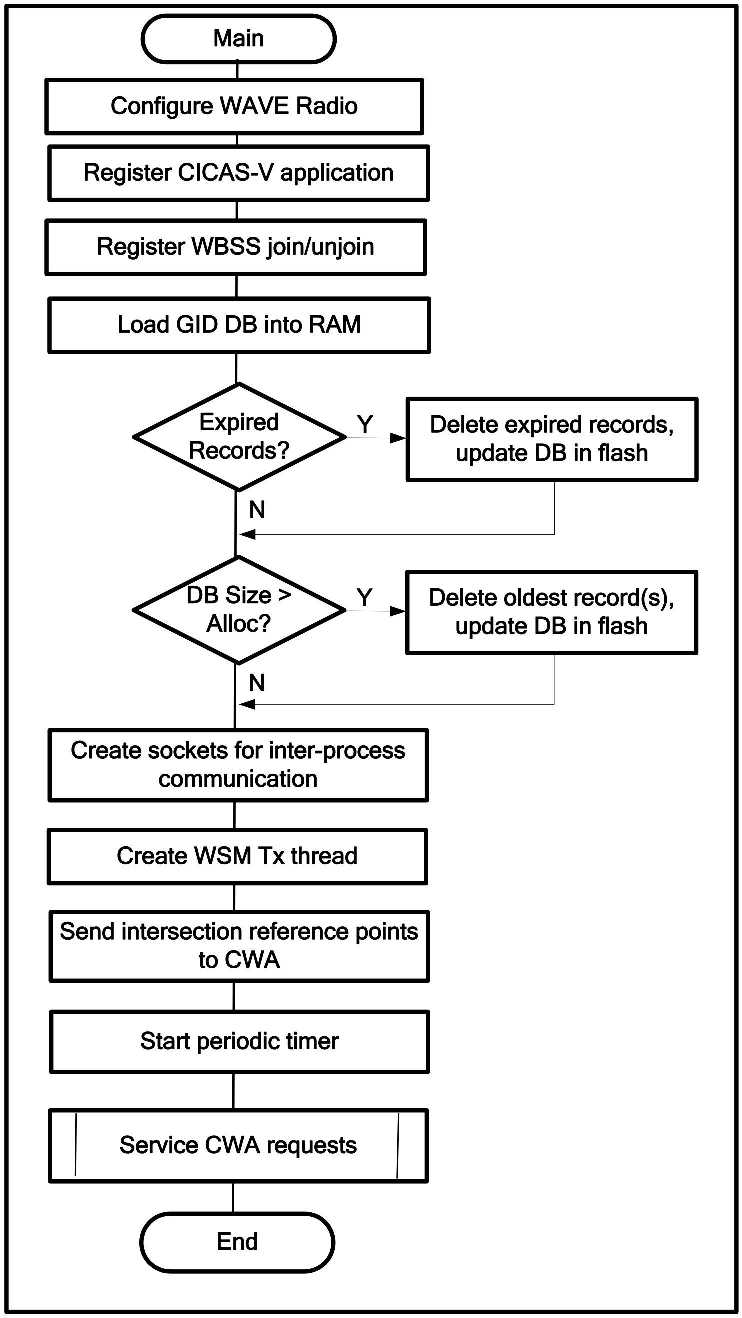

Figure 32: CDIP Main Thread Initialization Processing

Figure 33 illustrates the Main thread CWA servicing. CDIP accepts a command from the CWA to set the SPaT filter to the approaching intersection ID or no intersection available. If an intersection is available, it sets the SPaT expiration timer if it is not currently active or resets the timer if it is active. It also responds with the complete GID data for the intersection. If no intersection is available, it cancels the SPaT expiration timer.

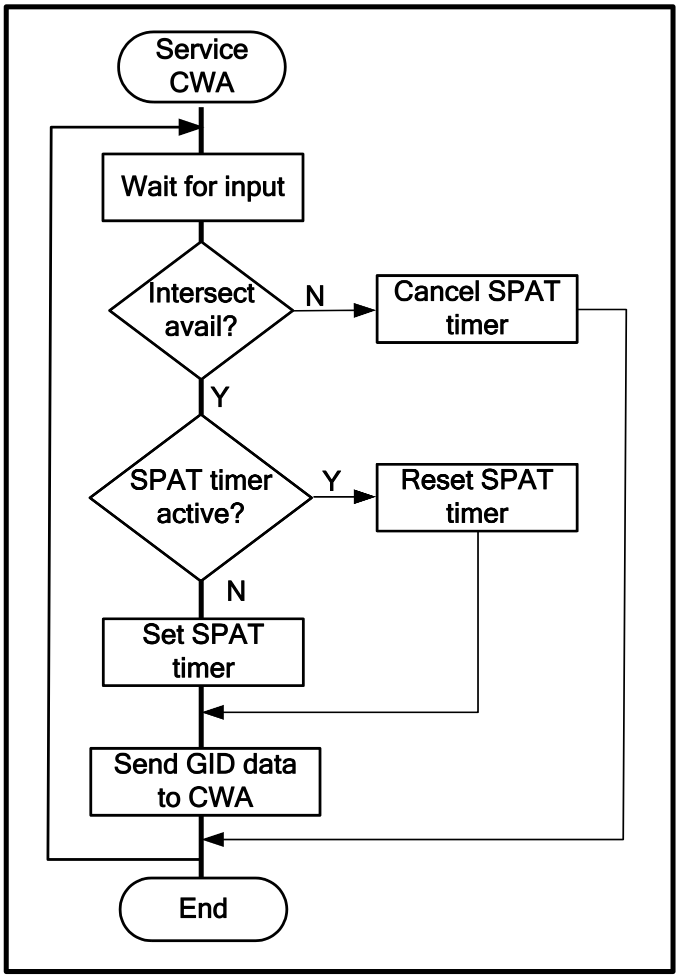

Figure 33: Main Thread CWA Servicing

A.5.1.2 WSA Callback

The WSU RS calls the WSA callback function when it receives a WSA for a CICAS-V service (i.e., the WSA contains the CICAS-V Provider Service ID). CDI determines whether to join the WBSS after further processing of the received GID or GPSC WSA.

Upon receipt of a GID WSA:

- If WSA contains area GID information, identify the intersection(s) corresponding to the area ID.
- Determine if the GID information has already been received and stored in the local database based on the Area or Intersection ID and content version in the Provider Service Context (PSC) data. If so, update the load time for the corresponding intersection(s) and discontinue processing the WSA.
- If a GID WSA has been received from only one RSE where the GID information is not in the database, request the RS module to join the corresponding WBSS.

- If a GID WSA has been received from more than one RSE where the GID information is not in the database, select the closest RSE based on the PSC Reference Point information and request the RS to join the corresponding WBSS.

Upon receipt of a GPSC WSA:

- If (a) no WBSS has been joined based on GID WSAs and (b) the GPSC is for the approaching intersection or no intersection has been selected and (c) the status is healthy and monitored, request the RS to join the WBSS.

Figure 34 illustrates the WSA processing.

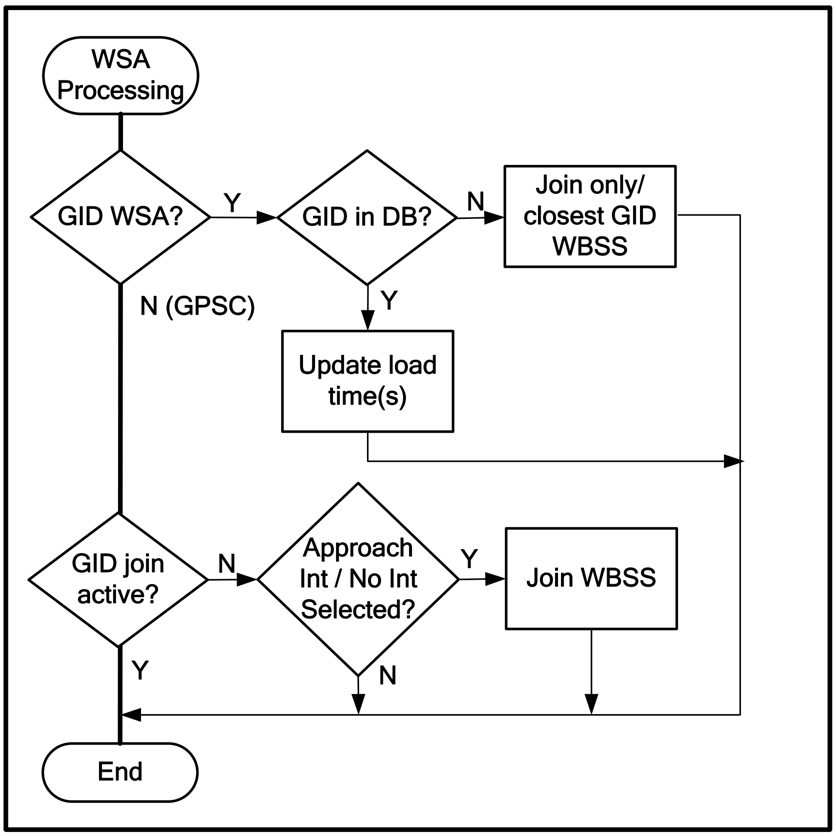

Figure 34: WSA Callback Processing

A.5.1.3 WSM Callback

The WSU RS calls the WSM callback function when it receives a WSM for the CICAS-V service (i.e., the WSM contains the CICAS-V Provider Service ID). The callback function performs the following processing:

1. Verifies the following in the TOM header:
    a. The message type indicates a TOM.
    b. The software supports the TOM framework and layer versions.
    c. The TOM length, TOM footer, and CRC are valid.
    d. The TOM contains GPSC, GID, or SPaT data.

2. If these checks are OK, perform TOM processing. Otherwise, discard the data.
3. For GPSC TOMs, send the data to the CVLP.
4. For SPaT TOMs:
    a. Parse and convert the data to a format usable by other processes.
    b. If the SPaT is for the current approaching intersection (provided by the CWA):
        - If the SPaT content version indicates it is a duplicate message (i.e., the content version is the same as the previous SPaT message), discard the data.
        - If the SPaT contains a Metric Object, output the elapsed time to CLOGP.
        - If the SPaT expiration timer is not active (i.e., this is the first SPaT for the current approaching intersection or the first SPaT following a previous timer expiration), set the SPaT timer to the configurable expiration period.
        - If the SPaT expiration timer is active, reset the timer.
        - Output the required data to CWA.
5. If the SPaT is not for the approaching intersection, discard the data.
6. For GID TOMs:
    a. Parse and convert the data to a format usable by other processes. If the GID contains an Area object, process individual intersection objects within the Area object.
    b. If the GID contains a Metric Object, output the elapsed time to CLOGP.
    c. If the GID data for the intersection does not exist in the database, check if the database is full (i.e., will exceed the configurable allocation with the addition of another record). If so, delete the record with the oldest load time. Add a new record with the GID data and load time and update the database in flash. Output the reference point to the CWA.
    d. If the GID data for the intersection already exists in the database:
        - If the GID content version is different, update the existing record with the new GID data and load time, and update the database in flash. Output the reference point to the CWA.
        - If the GID content version is the same, update the load time of the existing record in RAM, but do not update the database in flash. This avoids excessive writes to flash during operation (i.e., flash is not written every time a GID TOM is received).
    e. Write the updated database to flash periodically to store the latest load times.

f. If a WBSS was joined to obtain the GID, set the join request to inactive.

Figure 35 illustrates the WSM callback processing. Figure 36 illustrates the SPaT Handler processing and Figure 37 illustrates the GID DB Handler processing.

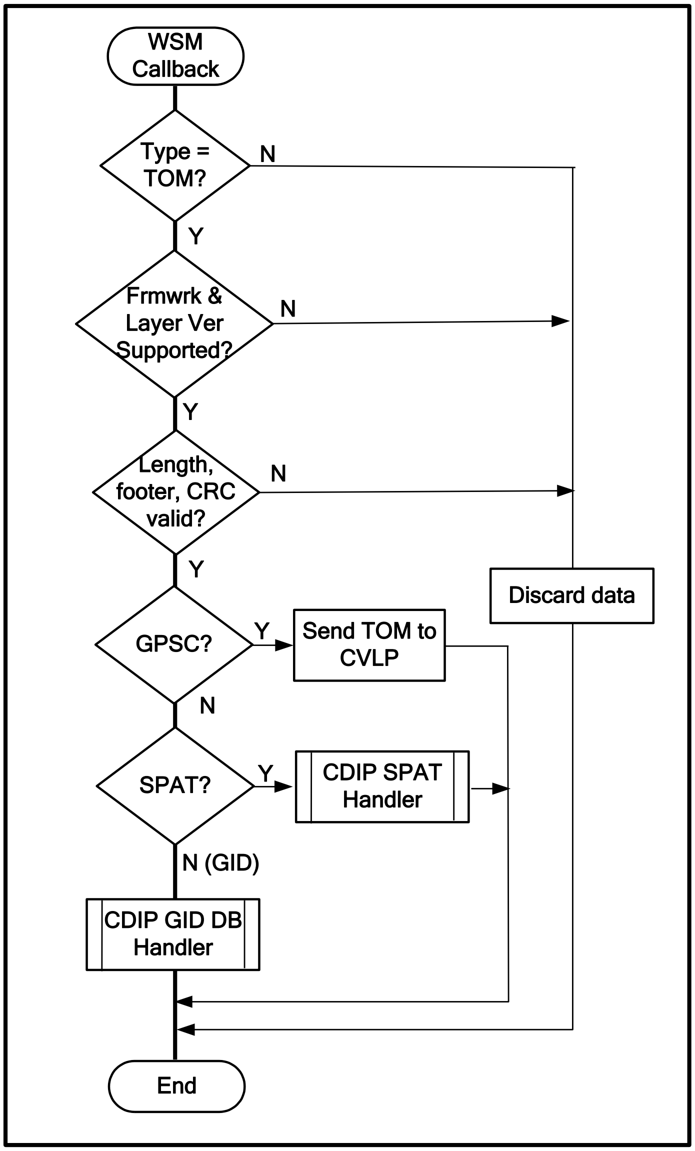

Figure 35: WSM Callback Processing

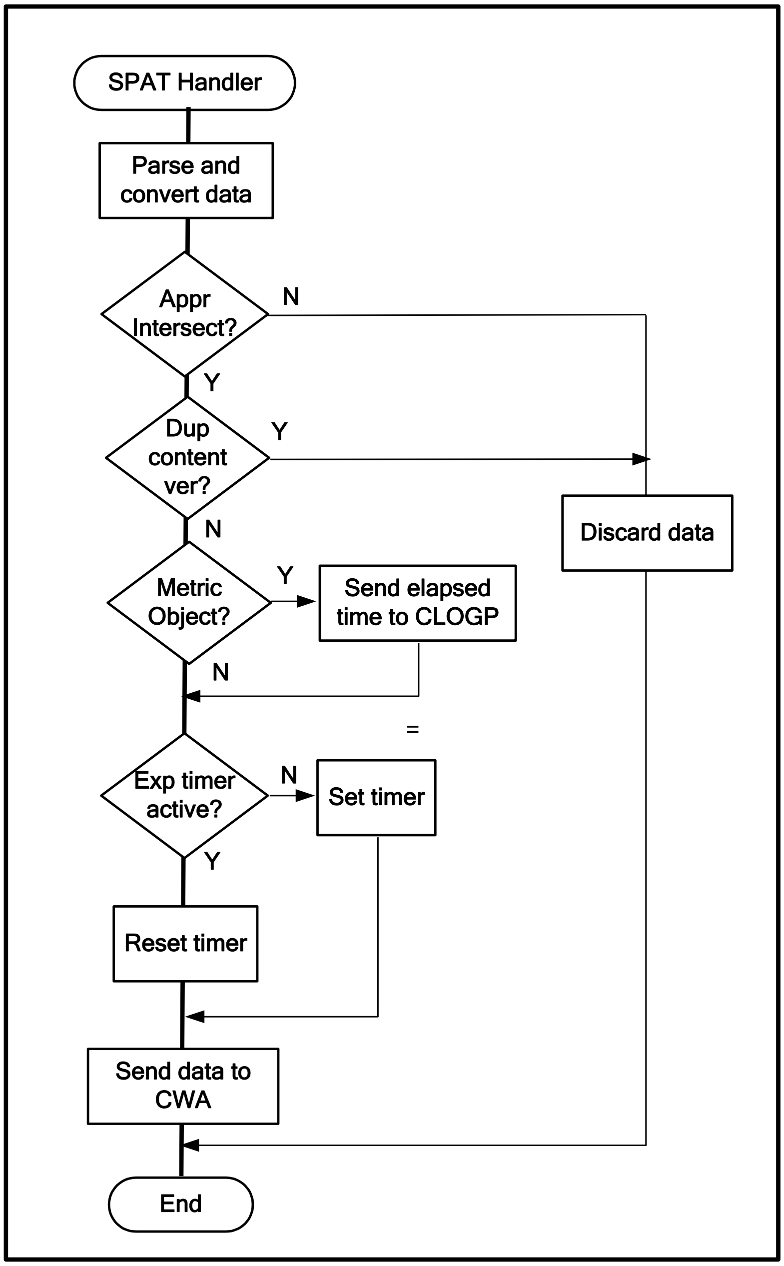

Figure 36: CDIP SPaT Handler Processing

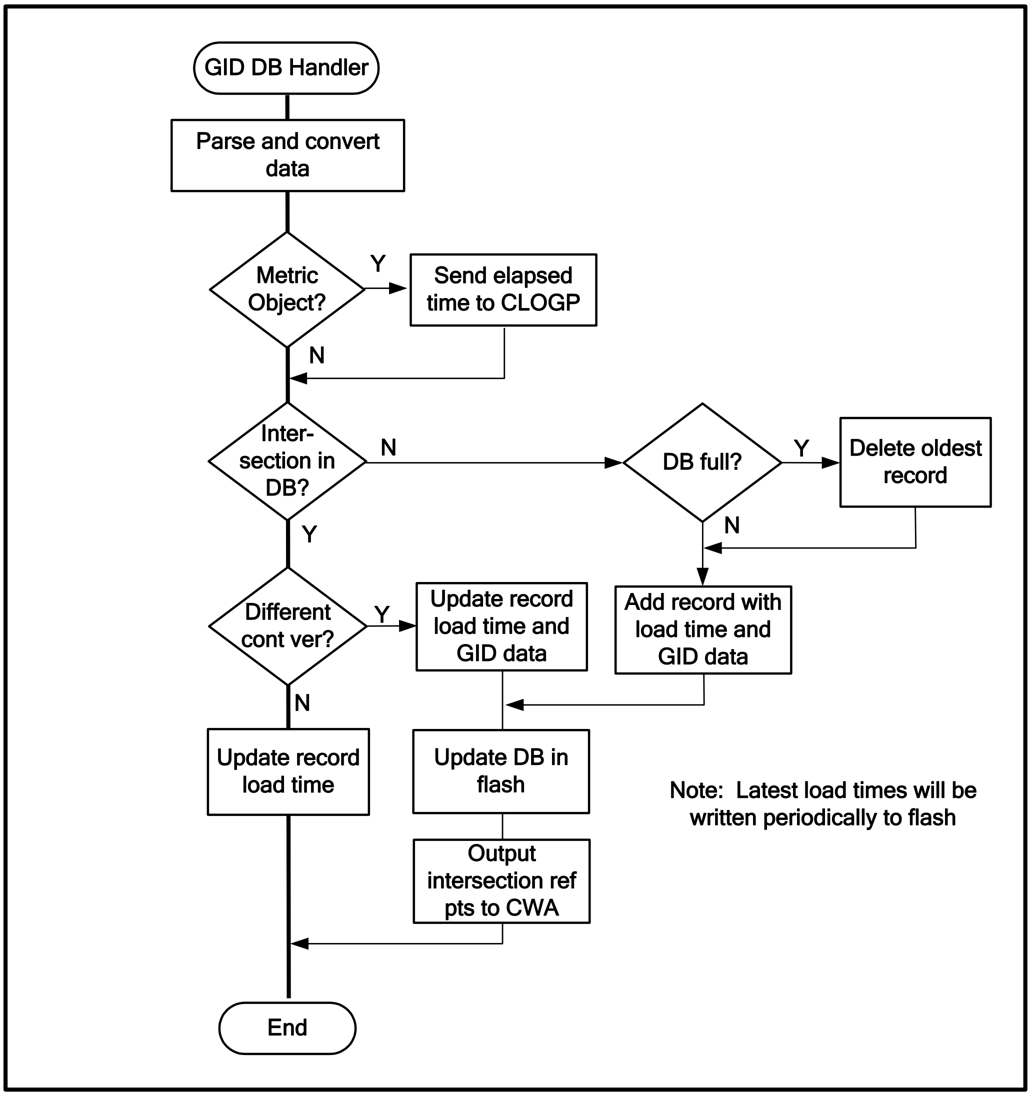

Figure 37: CDIP GID DB Handler Processing

A.5.1.4 WBSS Callback

The WSU RS provides an indication to CDIP when any WBSS join or un-join events occur. CDIP uses the information provided in the indication to log details about the join and un-join events.

A.5.1.5 Tx WSM Thread

The Tx WSM thread processes TSVWG and RCMD Tx requests from CWA. The Tx rate, power, and frequency for the Tx WSMs are configuration parameters. The WSM thread constructs the TOM messages and sends them to RS to be transmitted.

A.5.1.6 Timer Thread

The timer thread handles polling the radio statistics, detecting a SPaT timeout, and periodically sending DAS data to CLOGP. When the timer thread executes, it sends a message to the CWA, if the previous SPaT data has expired and is no longer valid. The SPaT timeout occurs when no new SPaT data has been received for the current intersection for a configurable period of time. The thread also polls the radio driver for statistics. The DAS data includes CDIP status and error indications.

A.6 CICAS-V Vehicle Location Process (CVLP)

Overview

The CICAS-V Vehicle Location Process (CVLP) performs the following functions:

1. Interface to the GPS receiver through the WSU Time/Position Services (TPS).
2. Register with the WSU TPS to receive GPS data updates and error status (e.g., NMEA checksum error).
3. Upon receiving vehicle location data (parsed GPS National Marine Electronics Association (NMEA) data) from TPS, send the required data to CWA. Also detect when the GPS updates have been missed, and the previous data has expired.
4. Upon receiving a GPSC TOM from the CDIP, validate the data and output the corrections to the GPS receiver through TPS.
5. Periodically output a message to CLOGP containing data required for the DAS.

Interfaces

Figure 38 illustrates the CVLP interfaces with the WSU TPS and other CICAS-V processes.

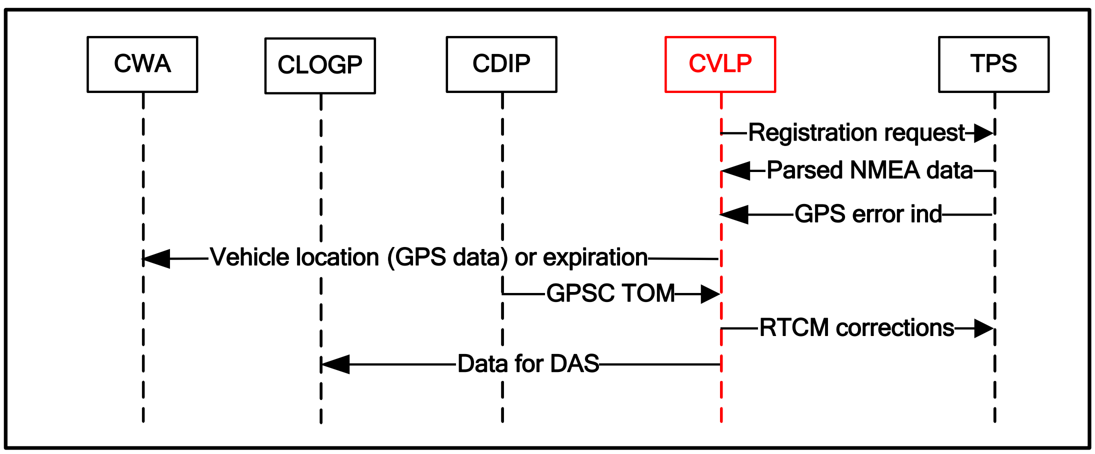

Figure 38: CVLP Interface Diagram

Process Structure

Figure 39 illustrates the CVLP thread decomposition and data flow. The CVLP Main thread registers with TPS to receive GPS data updates and error indications. The registration requests identify the associated callback functions. The Main thread also starts a timer to detect when the GPS data has expired. When the timer expires, the associated timer function executes in a separate thread.

TPS calls the GPS callback function when it receives and parses NMEA data. Upon receiving data, the callback function outputs the required vehicle location data to the

CWA and DAS data to CLOGP. The callback function executes as part of a TPS API thread.

TPS calls the GPS error callback function when it detects an NMEA checksum error or other GPS error. The callback function sets an error flag that is output to CLOGP the next time DAS data is output. The callback function executes as part of a TPS API thread.

The GPS data expiration timer is set/reset so it expires when no GPS data has been received for a configurable period of time and, therefore, the previous data has expired. If the timer expires, the GPS expiration timer thread executes and sends a vehicle location expiration notification to the CWA and DAS data to CLOGP.

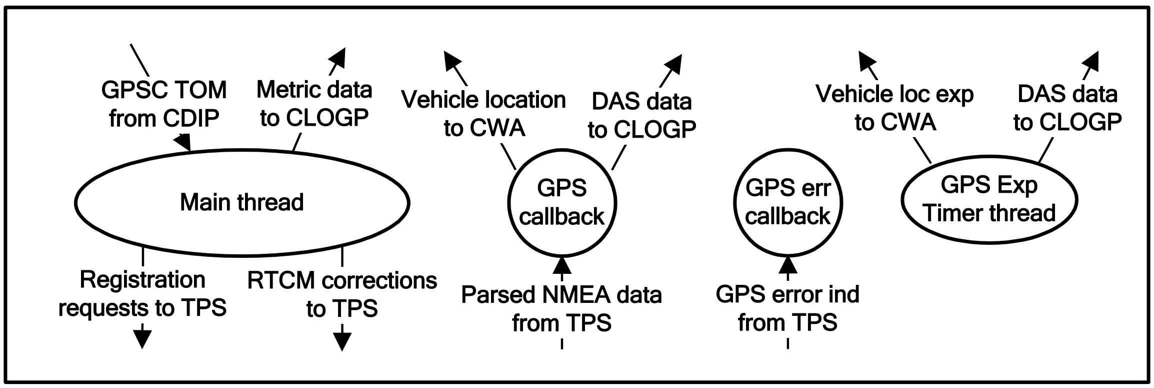

Figure 39: CVLP Thread Decomposition

Functional Description

The following subparagraphs describe the processing for each of the threads.

A.6.1.1 CVLP Main Thread

The CVLP Main thread performs the following processing:

1. Register with TPS to receive GPS data updates and error status. The registration requests include the corresponding callback function pointers.
2. Create sockets for communication with other processes.
3. Set the GPS expiration timer to the configurable expiration period.
4. Wait for a GPSC TOM from CDIP. Upon receipt of the TOM:
    a. If the TOM includes a Metric Object, output the elapsed time to CLOGP.
    b. Check if the GPS Status indicates healthy (i.e., the unhealthy and unmonitored indicators are not set), and verify the Radio Technical Commission for Maritime Services (RTCM) 1005 and 1001 message lengths are non-zero and the checksums are valid.
    c. If these conditions are true, output the RTCM messages to the GPS receiver. Otherwise, discard the data.
5. Return to Step 4.

Figure 40 illustrates the CVLP Main thread processing.

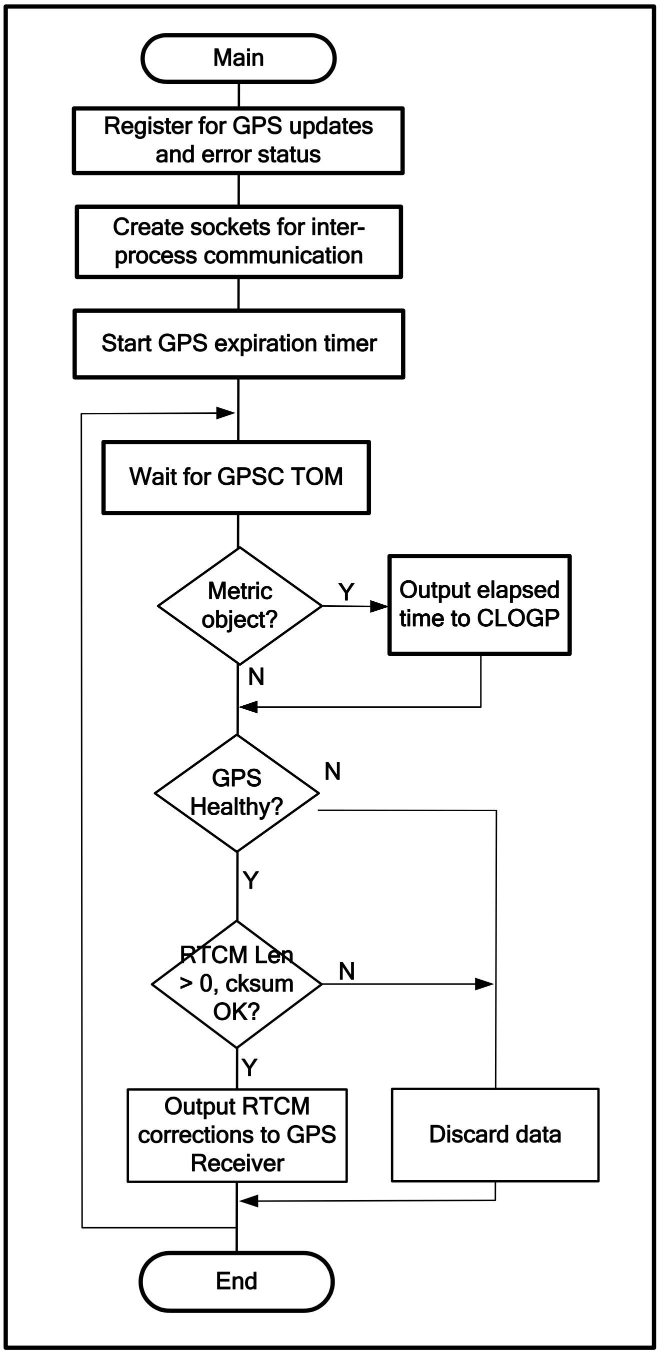

Figure 40: CVLP Main Thread Processing

A.6.1.2 GPS Callback

The WSU TPS calls the GPS callback function when it receives and parses NMEA data. The callback resets the GPS expiration timer and sends the required vehicle location data to the CWA and DAS data to CLOGP.

A.6.1.3 GPS Error Callback

The WSU TPS calls the GPS error callback function when it detects an NMEA checksum error or other GPS error. The callback function sets an error flag that is output to CLOGP the next time DAS data is output. The callback function executes as part of a TPS API thread.

A.6.1.4 GPS Data Expiration Timer Thread

When the GPS data expiration timer thread executes, it sends a message to the CWA that the previous vehicle location data has expired and is no longer valid. The expiration may occur because no data is being received (e.g., due to a faulty GPS or cable), the GPS receiver does not have a fix, or failure of NMEA checksums. If the log events are enabled, the CLOGP log data distinguishes among these cases. The thread also outputs DAS data to CLOGP.

A.7 CICAS-V Vehicle Information Process (CVIP)

Overview

The CICAS-V Vehicle Information Process (CVIP) performs the following functions:

1. Interface to the CAN bus through the WSU Vehicle/CAN Interface Services (VIS). Appendix A.22 defines the format and content for Vehicle-OBE CAN Interface for CICAS-V.
2. Register with the WSU VIS to receive CAN message updates.
3. Periodically transmit CAN message $704 to the Netway box.
4. Upon receiving the requested CAN messages, send the required data to CWA (i.e., send only the parameters that are needed by the destination process). Also detect when CAN updates have been missed and the previous data has expired.
5. Output a message to CLOGP containing data required for the DAS upon receipt of CAN data.

Interfaces

Figure 41 illustrates the CVIP interfaces.

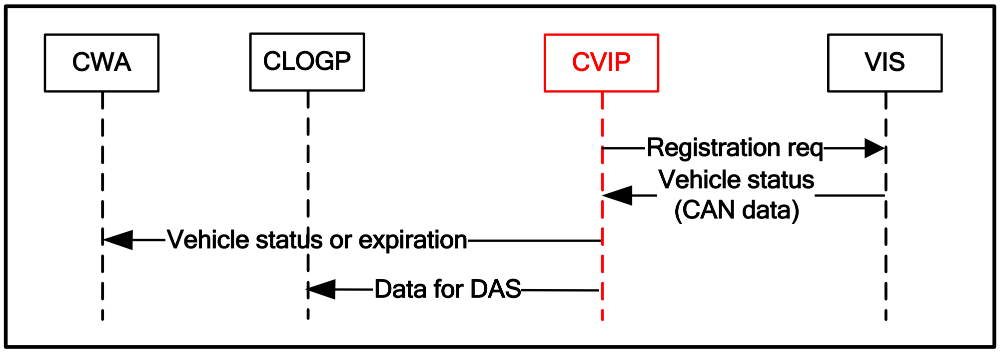

Figure 41: CVIP Interface Diagram

Process Structure

Figure 42 illustrates the CVIP thread decomposition and data flow. The CVIP Main thread registers with VIS to receive CAN message $600-605, $606, and $650 and provides separates callback function pointers for each of the three requests. It also starts a timer to detect when the CAN data has expired and a second timer for periodic transmission of the CAN $704 message.

VIS calls the corresponding callback function when it receives the requested CAN message data. Upon receiving the data, the callback function processes the CAN message(s), and outputs the required vehicle status data to the CWA and CLOGP. The callback functions execute as part of a VIS API thread.

The CAN data expiration timer is set/reset so it expires when CVIP has not received valid CAN data $600-$605 for the configurable expiration time, and therefore the previous data has expired. If the timer expires, the CAN expiration thread executes and sends a vehicle status expiration notification to the CWA and CLOGP.

CAN message $704 expires periodically and causes the CVIP to transmit a $704 message to the Netway box containing heartbeat sequence information.

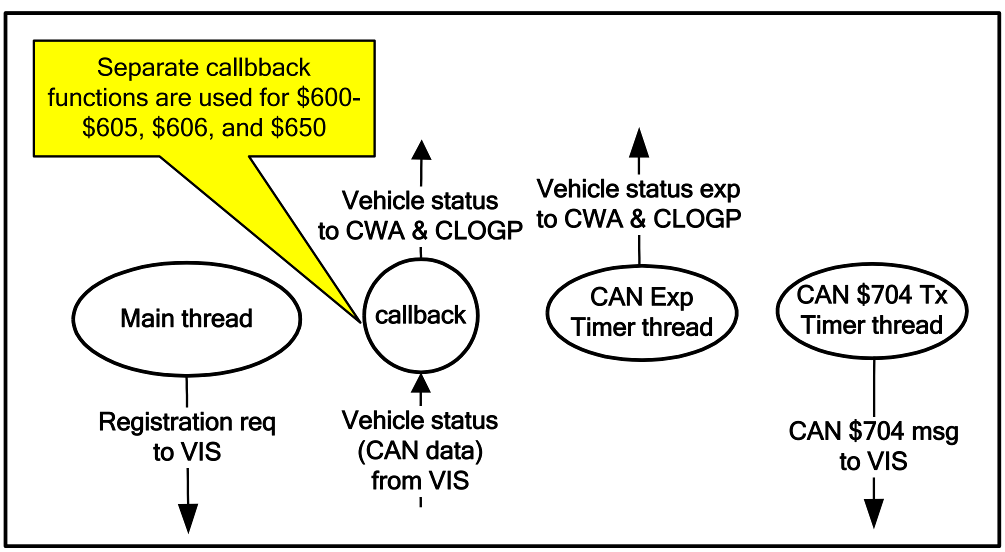

Figure 42: CVIP Thread Decomposition

Functional Description

The following subparagraphs describe the processing for each of the threads.

A.7.1.1 CVIP Main Thread

The CVIP Main thread performs the following processing:

1. Register with VIS to receive CAN messages $600-605, $606 and $650. The registration requests include the requested message IDs and the associated CAN callback function pointers.
2. Create sockets for communication with other processes.
3. Set the CAN expiration timer to the specified expiration period.
4. Set the CAN $704 Tx timer to the specified transmit interval.
5. Wait for the Linux process termination signal. The main thread does not perform any additional processing (all processing is performed in the callback and timer functions).

Figure 43 illustrates the processing.

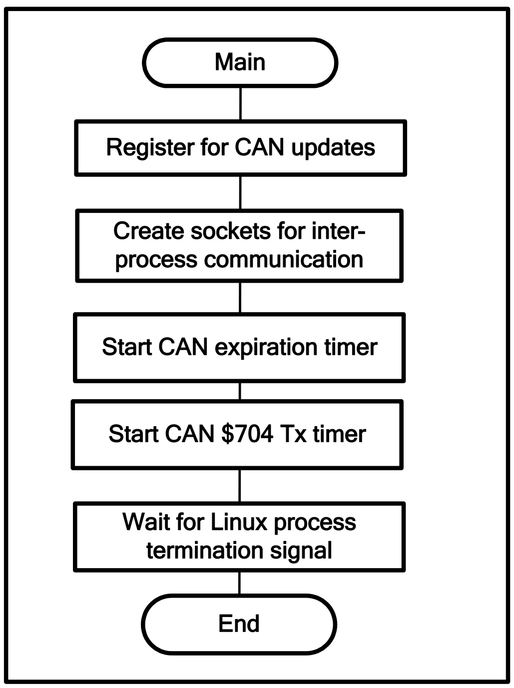

Figure 43: CVIP Main Thread Processing

A.7.1.2 CAN $600-605 Callback

The WSU VIS calls the CAN $600-605 callback function when it receives the requested CAN data. The callback function performs the following processing:

1. Check if all of the requested message IDs were received. If not, discard the data.
2. Reset the CAN expiration timer.
3. Send the raw message data to CLOGP and the required vehicle status data to CWA.

Figure 44 illustrates the processing.

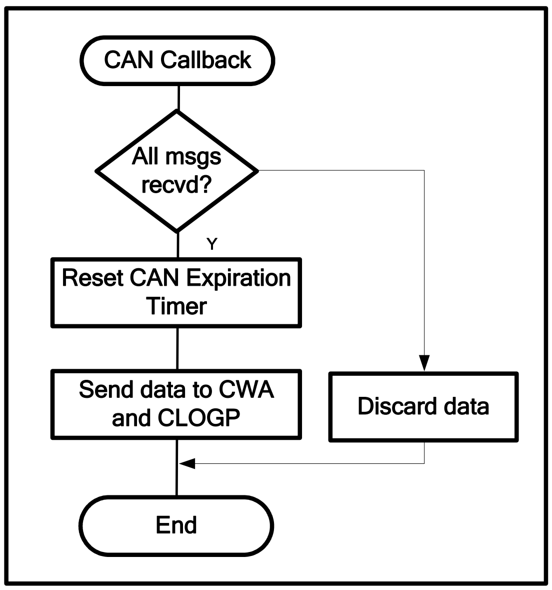

Figure 44: CAN $600-605 Callback Processing

A.7.1.3 CAN $606 Callback

The WSU VIS calls the CAN $606 callback function when it receives the requested CAN data. The callback function performs the following processing:

1. Check if the Netway to OBE heartbeat sequence counter is correct. If not, set an error flag that is forwarded to CLOGP when the next group of $600-605 messages is received.
2. Check for a vehicle CAN timeout indication. If so, the $600-605 data is considered invalid.
3. Send the raw $606 message data to CLOGP.

A.7.1.4 CAN $650 Callback

The WSU VIS calls the CAN $650 callback function when it receives the requested CAN data. The callback function performs the following processing:

1. Send the raw $650 message data to CLOGP.

A.7.1.5 CAN Data Expiration Timer Thread

When the CAN data expiration timer thread executes, it sends a message to the CWA and CLOGP that the previous vehicle status data has expired and is no longer valid.

A.7.1.6 CAN $704 Tx Timer Thread

When the CAN $704 timer thread executes, it checks if a CAN $606 message was received since the previous $704 transmission. If not, it sets an error flag that is forwarded to CLOGP when the next group of $600-605 messages is received. It then increments the OBE -> Netway sequence number and sends a $704 message to the Netway box through VIS.

A.8 CICAS-V Warning Application (CWA)

Overview

The CICAS-V Warning Application (CWA) is the violation detection process, which consists of the Intersection Identification module (IIM), Map Matching / Lane Identification module (MM-LIM), and Warning Algorithm (WA) module. The CWA performs the following functions:

1. Identify the most likely approaching intersection.
2. Determine the approach ID(s) of the probable lane(s) of travel and the distance to the stop bar.
3. Detect when a red light or stop sign violation is likely to occur.
4. Provide warning algorithm status to the DVIN.

The details of each module follow.

Interfaces

Figure 45 illustrates the CWA interfaces with the other CICAS-V processes.

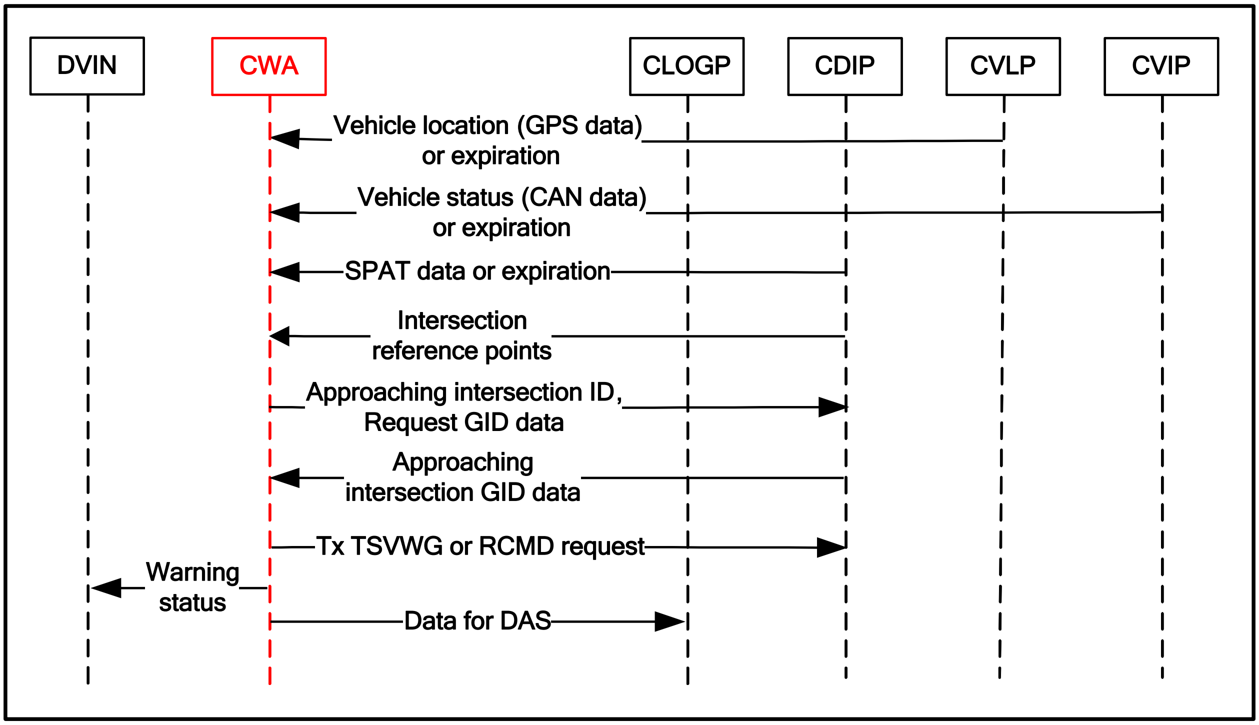

Figure 45: CWA Interface Diagram

Process Structure

Figure 46 illustrates the CWA thread decomposition and data flow. The CWA Main thread initializes, sets the configurable timer for the WA, and then waits to receive new intersection reference point data or new vehicle location (GPS) data. When new intersection reference point data is received, it stores the data in a local buffer. When new vehicle location data is received, it executes the IIM and MM-LIM algorithms. The Data Input thread waits to receive SPaT and vehicle status (CAN) data, and when received, stores the data in their respective buffers. When the timer expires, the Timer thread executes the WA algorithm. To avoid the situation where the Data Input thread is storing data while the WA Timer thread is reading it, both threads access the data using a mutex.

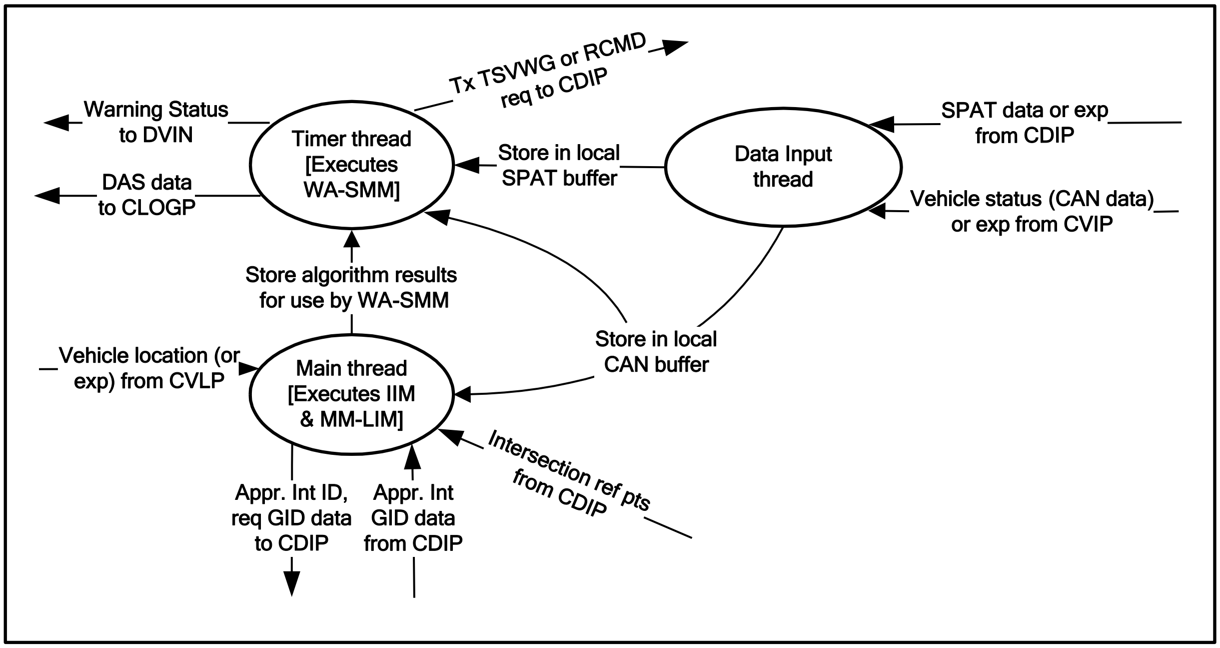

Figure 46: CWA Thread Decomposition

A.8.1.1.1 Functional Description

The following subparagraphs describe each of the threads and the processing of the modules executed as part of the threads.

A.8.1.2 CWA Threads

A.8.1.2.1 CWA Main Thread

The CWA Main thread performs the following functions:

1. Initialize the inputs to show no Intersection Reference Points have been received from CDIP. Initialize the outputs from each of the modules (i.e., IIM – no intersection available, MM-LIM – no approach IDs or distance to stop bar available, WA-SMM – no status).
2. Create sockets for communication with other processes.
3. Create the Data Input thread.
4. Set a configurable (nominally 100 ms) timer, which triggers the execution of the WA processing.
5. Wait to receive updates to the intersection reference point list or new vehicle location data.
6. If an intersection reference point update is received (add or delete), update the local buffer for use by the IIM and return to step 4.
7. If vehicle location data is received, check if the location data is valid and a non-zero list of Intersection Reference Points has been received from the CDIP. If so, perform the IIM processing described in Section A.8.1.3 and the MM-LIM processing described in Section A.8.1.4.

8. Return to Step 5.

Figure 47 illustrates the Main Thread processing.

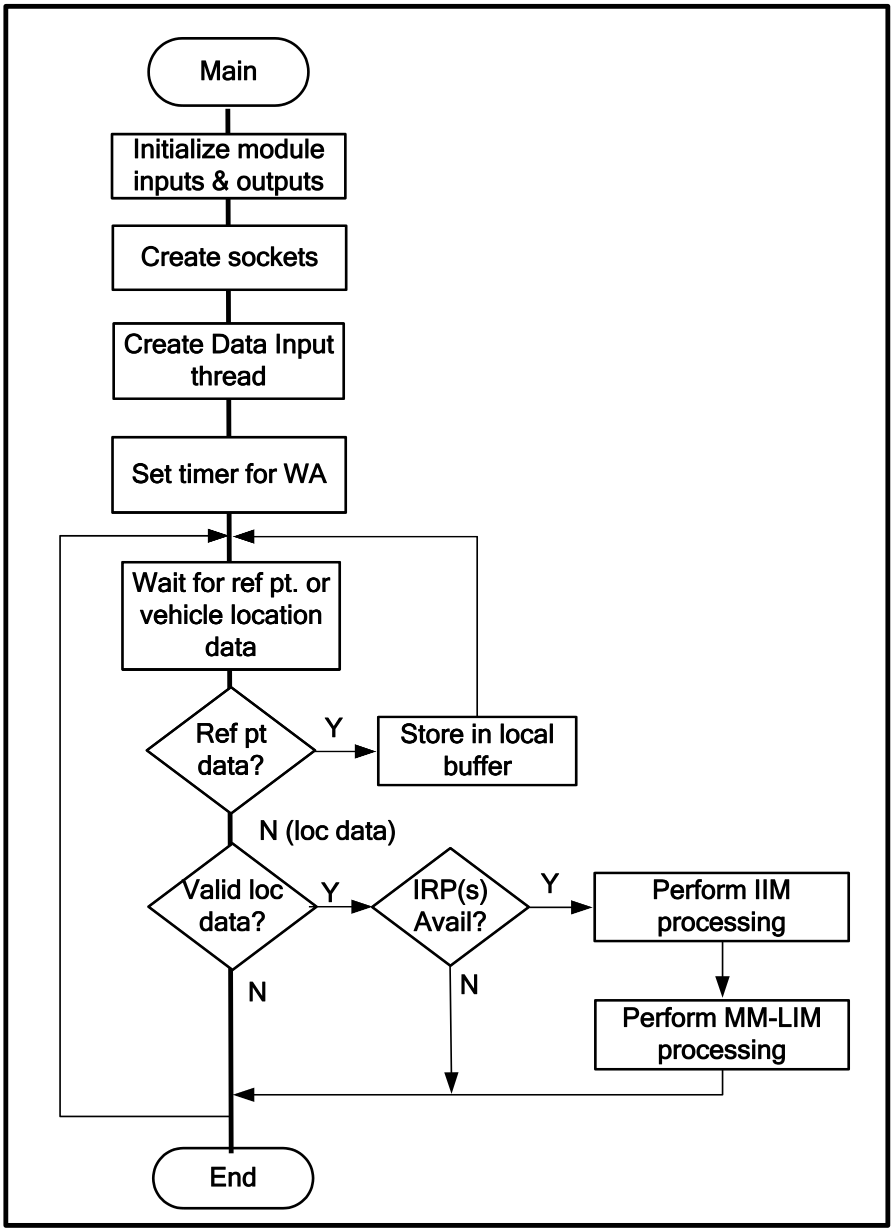

Figure 47: CWA Main Thread Processing

A.8.1.2.2 CWA Timer Thread

The CWA Timer thread executes the WA processing described in Section A.8.1.5.

A.8.1.2.3 CWA Data Input Thread

The Data Input thread waits for the following inputs:

1. SPaT data or SPaT expiration indication from the CDIP.
2. Vehicle status data (i.e., CAN data) or data expiration indication from the CVIP.

Upon receiving an input, the Data Input thread stores the data and waits for the next input.

A.8.1.3 Intersection Identification Module

A.8.1.3.1 Overview

The Intersection Identification Module (IIM) determines the nearby intersections and identifies the most likely approaching intersection(s) among them based on the vehicle location and the information in the GID database.

A.8.1.3.2 Functional Description

Figure 48 and Figure 49 illustrate the IIM processing.

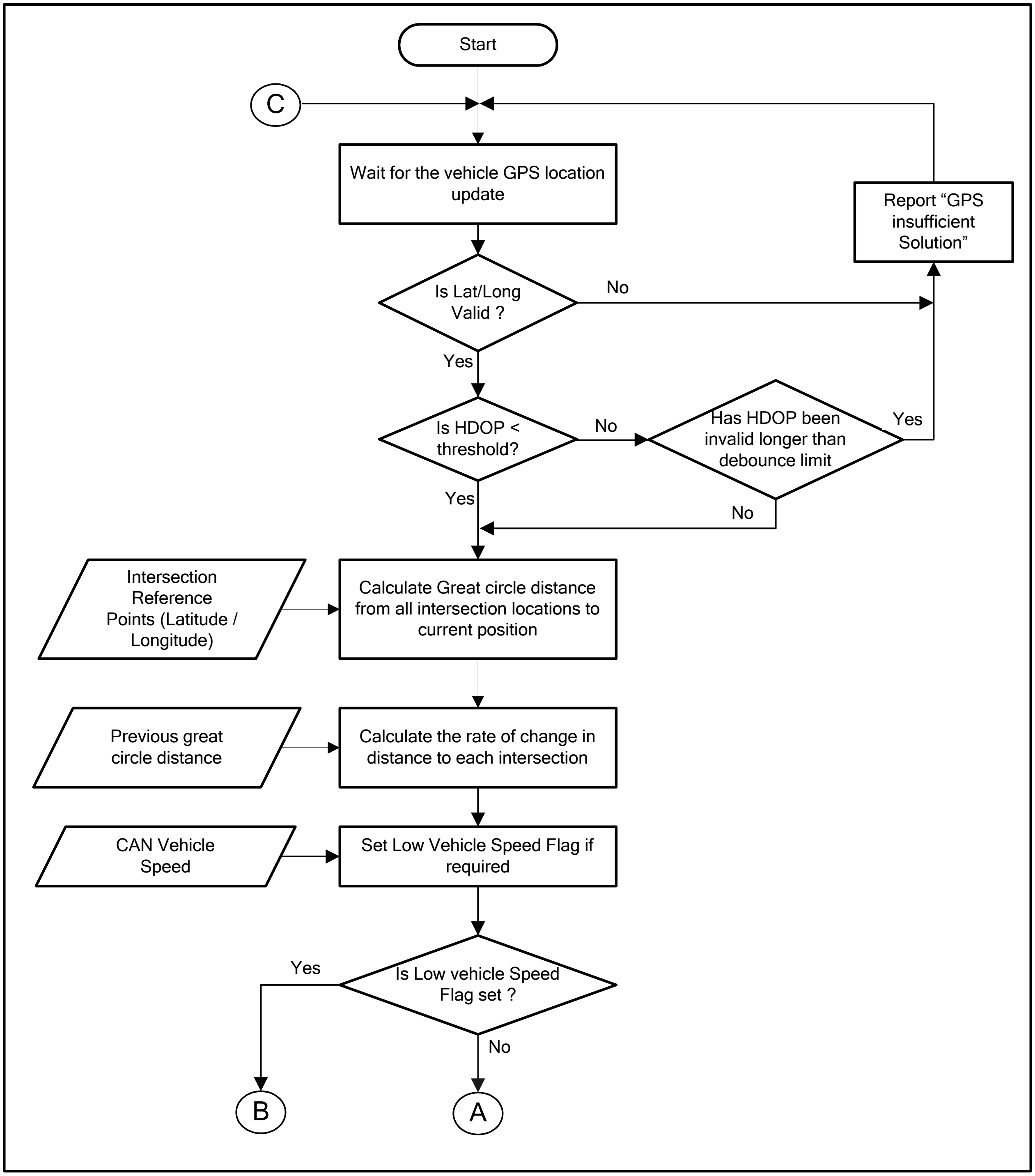

Figure 48: Intersection Identification Module Processing

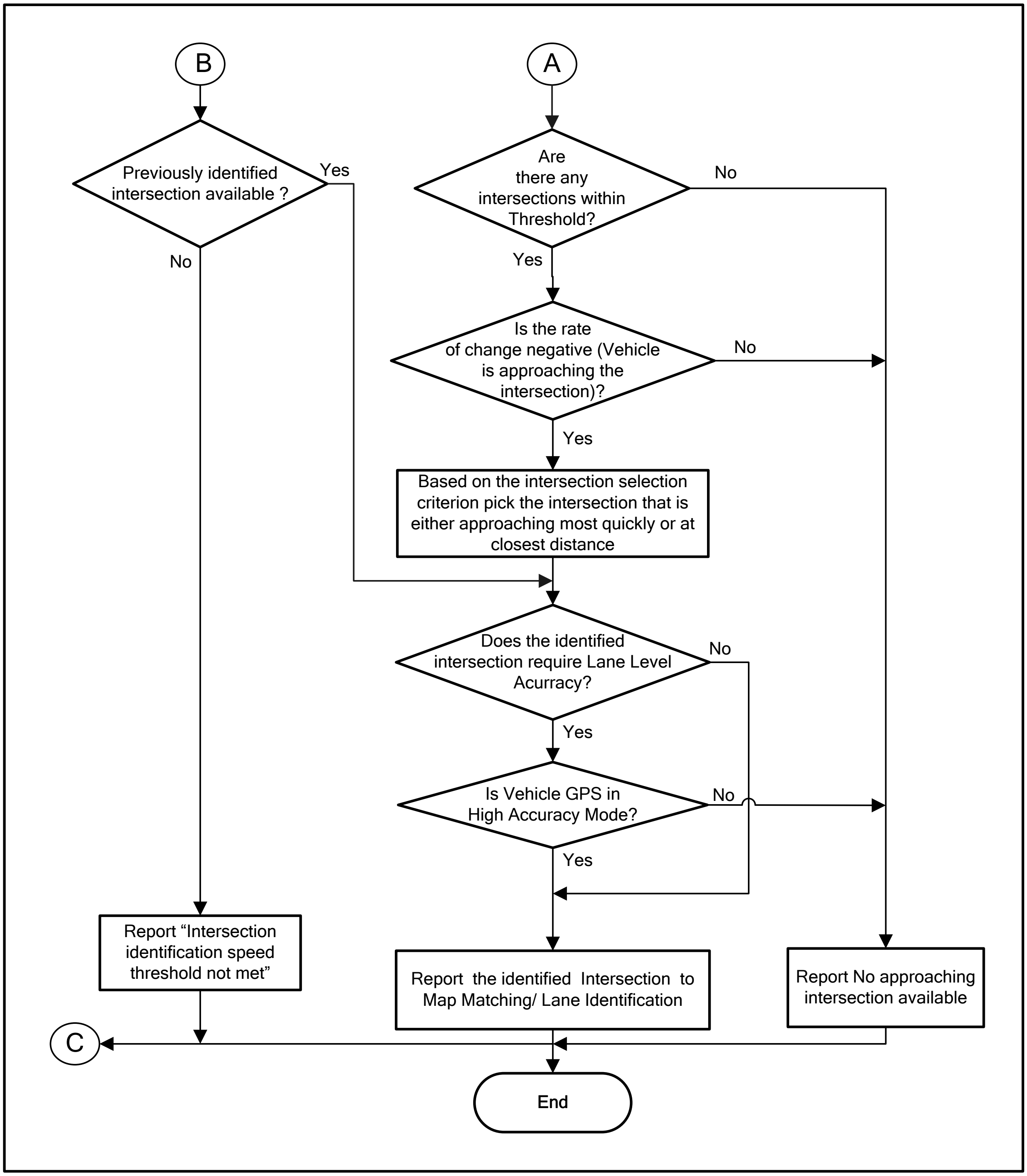

Figure 49: Intersection Identification Module Processing (continued)

The IIM algorithm validates the GPS solution update received from CVLP as follows:

1. Check latitude and longitude values for valid range. The valid range for latitude is 0 to 75 degrees north and for longitude 51 to 177 degrees west. This range covers all of North American land mass. If the latitude/longitude is out of range, then

IIM sets its status to "GPS invalid position" and aborts further processing in the current solution period.

2. Check if the HDOP value is less than the configurable threshold (nominally 3.0). If the HDOP exceeds the threshold for a configurable period of time, IIM sets its status to "GPS insufficient solution" and aborts further processing in the current solution period.

The CDIP provides the CWA with all of the intersection reference points (latitude and longitude information) from the GID database at startup and whenever any updates occur. IIM calculates the great circle distance from all intersection reference points found in the local database to the current vehicle location using the Haversine formula.

If vehicle location is ($lat_1, long_1$), intersection reference point is ($lat_2, long_2$) and R is the earth's radius as per WGS84, then Haversine formula can be used as:

$$\Delta lat = lat_2 - lat_1$$
$$\Delta long = long_2 - long_1$$
$$a = \sin^2(\Delta lat/2) + \cos(lat_1) * \cos(lat_2) * \sin^2(\Delta long/2)$$
$$c = 2 * atan2(\sqrt{a}, \sqrt{(1-a)})$$

Great circle distance between the vehicle location and intersection reference point is given by, $d = (R * c)$.

IIM prepares a list of nearby intersections that are less than a certain configurable threshold distance (nominally 300 m) away from the vehicle location. IIM calculates the rate of change in distance to each nearby intersection location and maintains a history of these values. The rate of change is used to determine if the vehicle is approaching or moving away from the intersection.

IIM supports three methods for calculating rate of change. Only the configured method is used in further processing but the results of all three methods may be logged for performance comparison. The following paragraphs describe the three methods:

1. **CAN vehicle speed and GPS heading method:**

   Figure 50 illustrates the GPS and calculated heading to intersection.

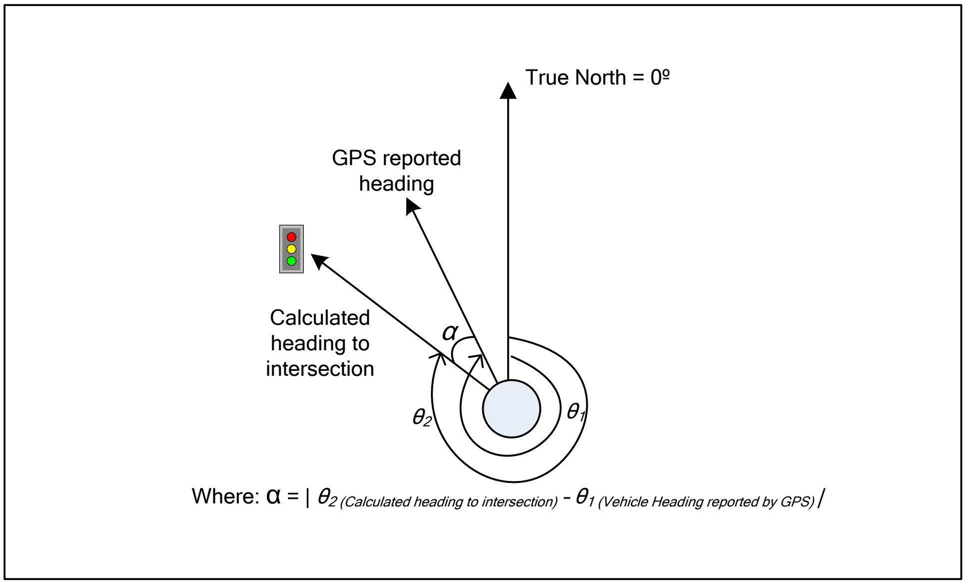

Figure 50: Calculated and GPS Heading

This method calculates the vehicle rate of change based on the equation:

Velocity made good = CAN reported speed * Cosine (| θ_2 (Calculated heading to intersection) – θ_1 (vehicle heading reported by GPS |)

Where:

Velocity made good	= vehicle rate of change.
CAN reported speed	= speed reported from vehicle CAN bus.
θ_1 (vehicle heading reported by GPS)	= true heading reported by GPS.
θ_2 *(Calculated heading to intersection):*	= calculated heading relative to true north.

The following formula gives the initial heading for a great-circle route between two latitude and longitude points:

$\Delta lat = lat_2 - lat_1$
$\Delta long = long_2 - long_1$

$\theta 2$ = mod(atan2(sin(Δlong)*cos(lat_2), cos(lat_1)*sin(lat_2) – sin(lat_1)*cos(lat_2).cos(Δlong)),2 * PI)

Where: PI= 3.141592...

2. GPS location data method:

This method calculates the rate of change by determining the change in distance to a particular intersection since the last vehicle location update, and calculating a moving average of this value for a configurable duration (nominally 1 second).

If d_n is the distance of a particular intersection from the current vehicle location, d_{n-1} was the distance t_n seconds prior, then rate of change in distance is given by,

$r = [\sum (d_n - d_{n-1}) / t_n] / npoints$

Where: $\sum$ is summation of n = 1 to the number of points (npoints) corresponding to the duration of the moving average calculation.

The moving average is not considered valid until it has accumulated data points over the specified duration (i.e., the intersection will not be considered for selection until this occurs).

3. **GPS location data method with filter:**

This method calculates the rate of change by first calculating distance to intersection repeatedly and applying a filter to the results.

The distance between the intersection and vehicle location (based on GPS position updated) is calculated by using the great circle method.

Once the first two GPS position updates are available, the calculated great circle distances to the intersection are used in the following equation to create initial conditions for the filter:

Rate of change in distance$_{initial}$ =

$$\frac{\text{distance to intersection}_2 - \text{distance to intersection}_1}{T}$$

Where:

Rate of change in distance$_{initial}$	= rate of change in distance value for initialization.
distance to intersection$_n$	= the *n*th distance to the intersection.
T	= Sampling time (e.g., 10Hz GPS, T = 0.1s)

Once the third GPS location update is provided, the third distance to the intersection is calculated, and this distance (*distance$_n$*) along with the previous rate of change (Rate of change in distance$_{initial}$) are used in the following filter while calculating rate of change:

Rate of change in distance$_n$ =

$$\frac{1}{1 + (2\pi FT)} \left((\text{rate of change in distance}_{n-1}) + 2\pi F(\text{distance}_n - \text{distance}_{n-1})\right)$$

Where:

Rate of change in distance$_n$	= the n^{th} filtered vehicle speed.
rate of change in distance$_{n-1}$	= the $(n - 1)^{th}$ filtered vehicle speed.
distance$_n$	= the n^{th} distance from the intersection.
T	= sampling time (e.g., 10Hz GPS, T =0.1s).
F	= the cut off frequency for the filter. This is a configurable parameter with a valid range from 0.5 to 5 Hz.

In order to prevent normal GPS inaccuracies from impacting the performance of IIM, while the vehicle is moving at low speed, IIM maintains a flag to indicate this condition.

IIM sets the low vehicle speed flag when the following conditions are true:

1. If the CAN vehicle speed is below the configurable threshold value (nominally 3 mph).
2. If the configurable "enable low vehicle speed filtering" flag is set.

If the low vehicle speed flag is set, then IIM identifies the approaching intersection as the one which is identified previously. If there has been no intersection identified previously, IIM status is set "IID speed threshold not met" and aborts further processing.

If the low vehicle speed flag is not set, IIM identifies the most likely approaching intersection as the one:

1. With the rate of change in distance indicating the vehicle's direction of travel is toward the intersection. A negative rate of change indicates traveling towards the intersection and positive away from the intersection.
2. Meets the configurable intersection selection criterion which is either approaching most quickly or closest distance.

IIM reports the identified approaching intersection ID or none, if no intersection is identified to MM-LIM algorithm and sets its status accordingly.

A.8.1.4 Map Matching / Lane Identification Module

A.8.1.4.1 Overview

The Map Matching/Lane Identification module (MM-LIM) determines the approach ID(s) of the probable lane(s) of travel, the distance to the stop bar and likelihood of approach based on the vehicle location and the information in the GID database. Depending upon number of approaches matched, this module may report information for more than one approach.

A.8.1.4.2 Functional Description

Figure 51 and Figure 52 illustrate the MM-LIM logic flow.

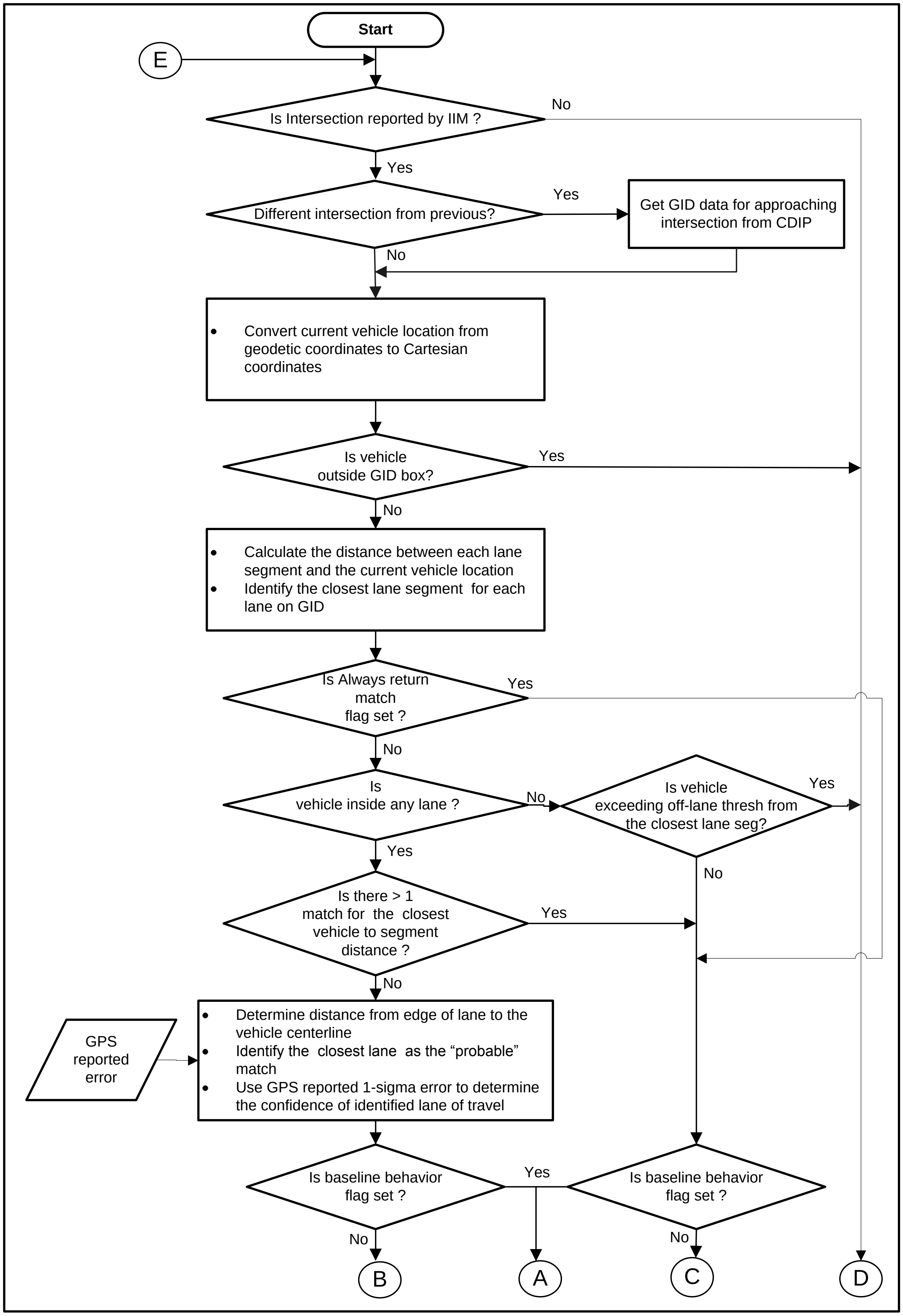

Figure 51: Map Matching/Lane Identification Module Processing

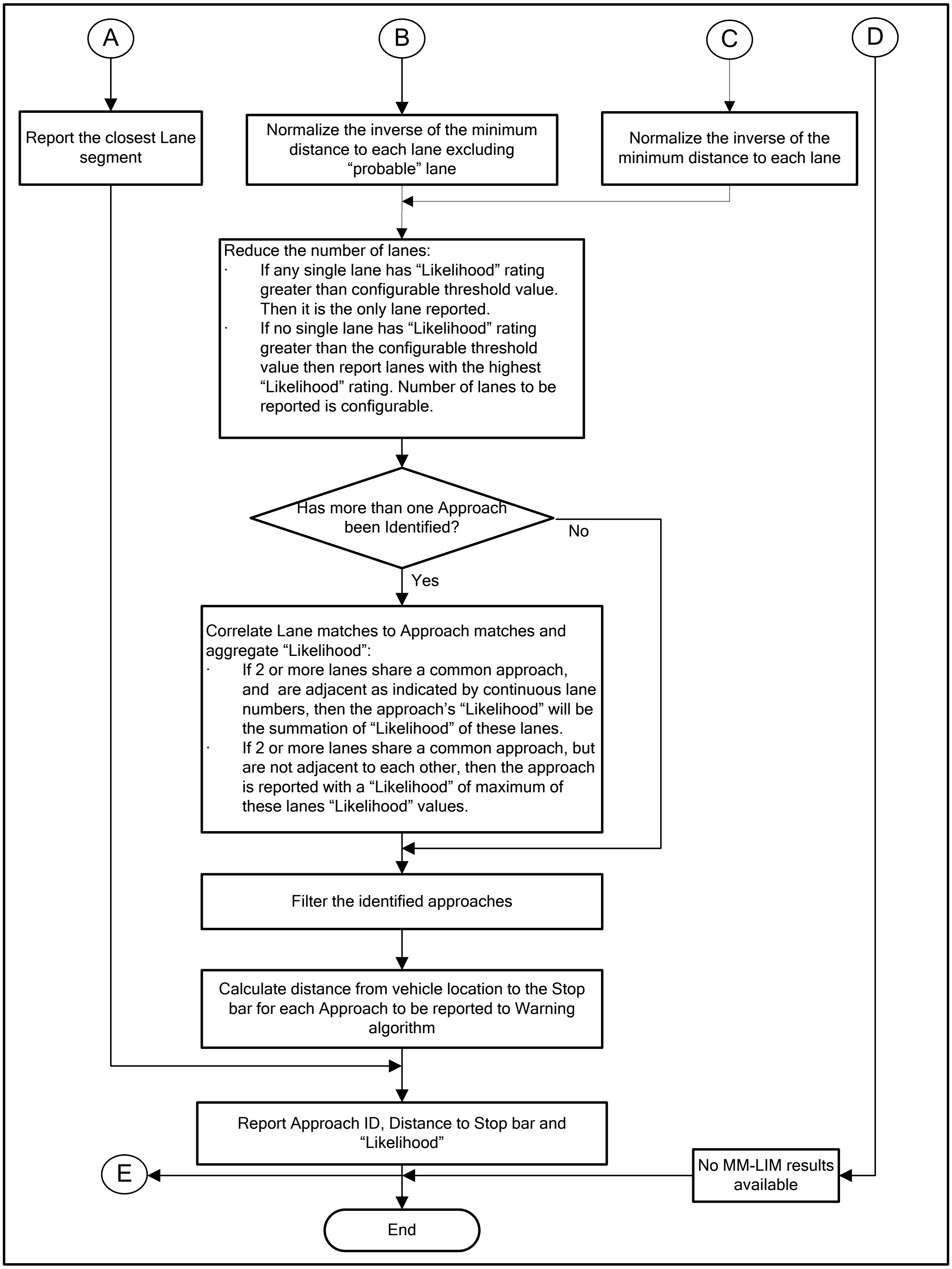

Figure 52: Map Matching/Lane Identification Module Processing (continued)

The MM-LIM algorithm starts its processing by receiving identified intersection ID from IIM.

If the approaching intersection has changed from the previous execution, then MM-LIM sends the approaching intersection (intersection ID or no intersection available) to the CDIP. CDIP sets the SPaT filter based on the approaching intersection. If an intersection ID has been selected, CDIP also returns the GID data for the intersection to CWA. If the approaching intersection is the same as the previous execution, then MM-LIM starts processing the previous GID data. If IIM reports no intersection, then MM-LIM sets its status to intersection unavailable and aborts further processing for current GPS solution period.

MM-LIM maps the location of the vehicle on to the GID by converting the current vehicle location from geodetic coordinates to Cartesian coordinates by finding the linear distances, x (EW distance), y (NS distance), d (shortest "direct" distance) from the intersection reference point (RP), as illustrated in Figure 53. The Vincenty ellipsoid formula for geodesic distance between two latitude / longitude points is used for this distance calculation. Vincenty's ellipsoid formula is claimed to be accurate to within 0.5mm based onan approximated WGS84 reference ellipsoid. Calculations based on a spherical model, such as the much simpler Haversine formula, are accurate to around 0.3%.

The Vincenty equations described below are based on Vincenty's original publication [8] and scripts from http://www.movable-type.co.uk/scripts/latlong-vincenty.html. This web site allows full re-use of the scripts with retention of the following copyright notice: *"© 2002-2006 Chris Veness*" and a reference to the web site.

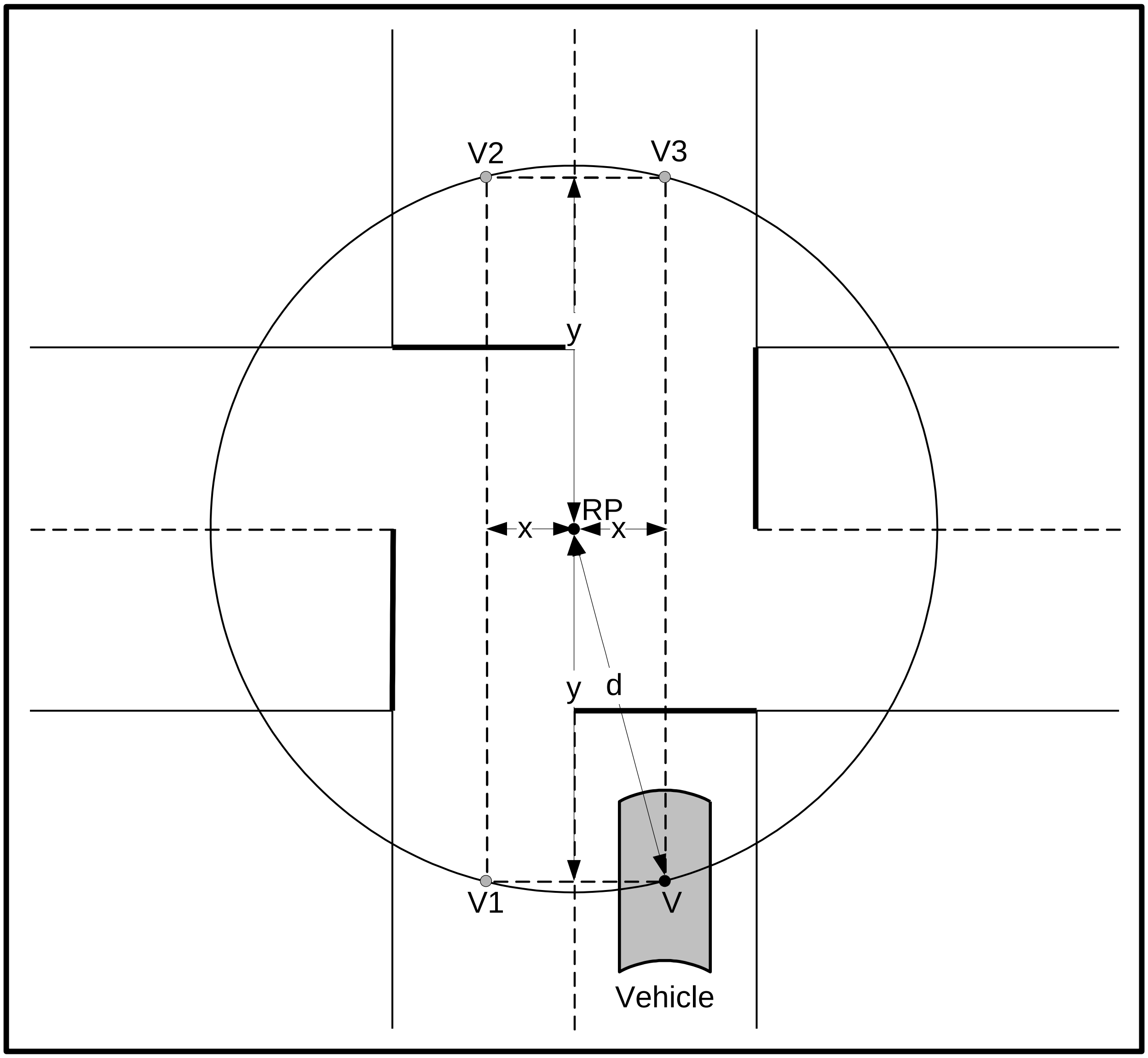

Figure 53: Map Vehicle Location on to GID

Shortest “direct” distance d indicates the vehicle location to be anywhere on the circle with radius d. EW distance, x and NS distance, y narrows down the vehicle location to V(x, -y), V1(-x, -y), V2(-x, y) or V3(x, y). Considering the relative location of the reference point latitude / longitude with respect to the vehicle location latitude / longitude allows the real vehicle location, V, to be identified.

Vincenty’s formula as it is used here is as follows:

a, b = major & minor semi-axes of the ellipsoid, f = flattening is given by $(a - b) / a$.

As per WGS84, a = 6,378,137m; b = 6,356,752.3142m; f = 1 / 298.257223563

$\varphi 1$, $\varphi 2$ = geodetic latitude, L = difference in longitude, U = reduced latitude

$U1 = atan((1 - f) * tan\varphi 1)$

$U2 = atan((1 - f) * tan\varphi 2)$

$\lambda = L$, $\lambda' = 2\pi$

while abs($\lambda - \lambda'$) > 10^{-12} { (i.e., 0.06mm)

$\sin\sigma = \sqrt{[(\cos U2 * \sin\lambda)^2 + (\cos U1 * \sin U2 - \sin U1 * \cos U2 * \cos\lambda)^2]}$

$\cos\sigma = \sin U1 * \sin U2 + \cos U1 * \cos U2 * \cos\lambda$

$\sigma = \text{atan2}(\sin\sigma, \cos\sigma)$

$\sin\alpha = \cos U1 * \cos U2 * \sin\lambda / \sin\sigma$

$\cos^2\alpha = 1 - \sin^2\alpha$

$\cos 2\sigma_m = \cos\sigma - 2 * \sin U1 * \sin U2 / \cos^2\alpha$ (Note)

$C = f / 16 * \cos^2\alpha * [4 + f * (4 - 3 * \cos^2\alpha)]$

$\lambda' = \lambda$

$\lambda = L + (1 - C) * f * \sin\alpha * \{\sigma + C * \sin\sigma * [\cos 2\sigma_m + C * \cos\sigma * (-1 + 2 * \cos^2 2\sigma_m)]\}$

}

$u^2 = \cos^2\alpha * (a^2 - b^2) / b^2$

$A = 1 + u^2 / 16384 * \{4096 + u^2 * [-768 + u^2 * (320 - 175 * u^2)]\}$

$B = u^2 / 1024 * \{256 + u^2 * [-128 + u^2 * (74 - 47 * u^2)]\}$

$\Delta\sigma = B * \sin\sigma * \{\cos 2\sigma_m + B / 4 * [\cos\sigma * (-1 + 2 * \cos^2 2\sigma_m) - B / 6 * \cos 2\sigma_m * (-3 + 4 * \sin^2\sigma) * (-3 + 4 * \cos^2 2\sigma_m)]\}$

$s = b * A * (\sigma - \Delta\sigma)$, the distance (in the same units as a & b)

Note: Vincenty observes this equation becomes indeterminate over equatorial lines (since $\cos^2\alpha \rightarrow 0$). In this case, $\cos 2\sigma_m$ should be set to 0 and then the result is computed correctly. Also the formula might have no solution for the case when the points are between two nearly antipodal points. An iteration limit should be used in such a case.

Vehicle V1 is within the intersection box boundary (I0-I1-I2-I3-I0) as illustrated in Figure 54. The MM-LIM does not have any processing to detect when the vehicle is within the intersection box, since the GID does not contain intersection box boundary information. However, the WA checks if the distance to the stop bar is increasing and does not issue a violation warning when this condition is true.

Vehicle V2 is outside the GID box as illustrated in Figure 54. MM-LIM aborts the algorithm execution if it determines that the vehicle location is outside the GID box. The minimum and maximum of all node segment coordinates, N1 – N8 define the boundary and thus any location beyond this boundary can be considered outside the GID box.

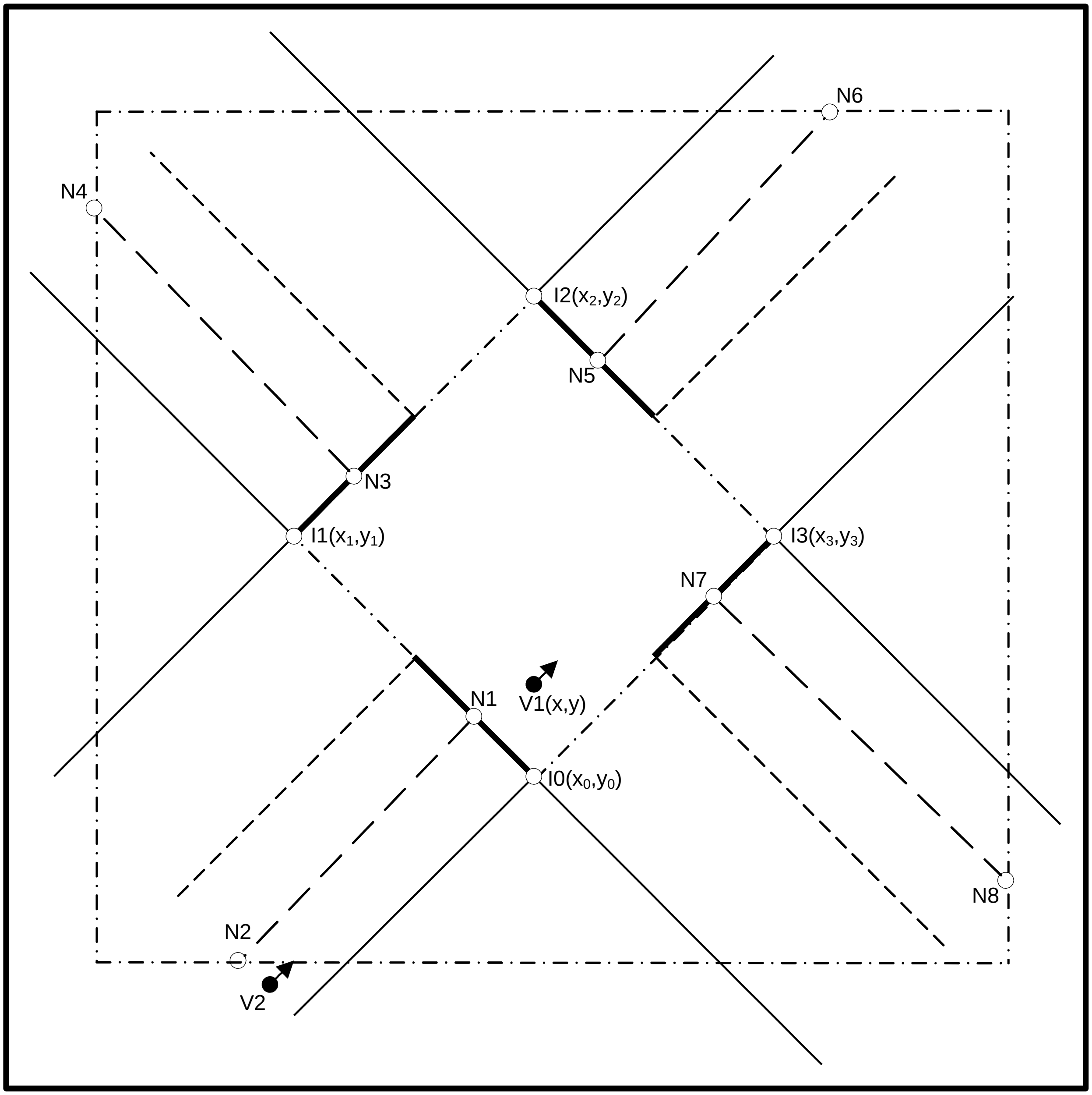

Figure 54: Intersection Box / Off GID Box

In order to resolve location in terms of actual lane of travel, MM-LIM calculates the shortest distance between the current vehicle location and each lane segment in the GID data. Figure 55 illustrates an example of the distances calculated. d_0, d_1 and d_2 are the distances calculated from the current vehicle location, V, to lane segments, N0-N1, N1-N2 and N2-N3 respectively. Note that d_1 is the perpendicular distance to lane segment N1-N2 but d_0 and d_2 are the shortest distance to the edge of the lane segments N0-N1 and N2-N3 respectively. [Note that Figure 54 and Figure 55 are independent illustrations. The vehicle location and reference points in Figure 54 do not correspond to the points in Figure 55].

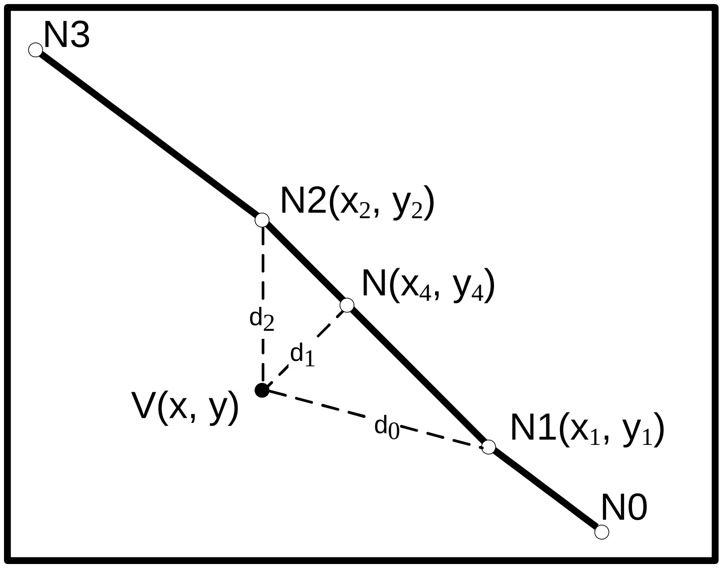

Figure 55: Distance to Lane Segment

The method to calculate the distance from a point to a lane segment as used here is as follows.

Considering line segment N1(x_1, y_1) – N2(x_2, y_2) and vehicle location, V(x, y), the distance d_1 is the length of the line segment, V – N that is normal to the line segment, N1 – N2 and intersects at the point N(x_4, y_4) given by,

$t = ((x - x_1) * (x_2 - x_1) + (y - y_1) * (y_2 - y_1)) / ((y_2 - y_1)^2 + (x_2 - x_1)^2)$

Limit t between 0 and 1. This takes care of the condition applicable to distances d_0 and d_2.

$x_4 = x_1 + t * (x_2 - x_1)$; $y_4 = y_1 + t * (y_2 - y_1)$

$d_1 = \sqrt{((y_4 - y)^2 + (x_4 - x)^2)}$

MM-LIM identifies a lane as vehicle probable lane of travel (“probable” lane), if the vehicle is inside that lane and the lane has the minimum vehicle to segment distance. The vehicle is considered inside a lane when the distance from the vehicle to lane segment is less than the reported lane width for that segment. If a GID is constructed in such way that 2 lanes overlap (e.g., two nodes are identical because a right or left turn lane is branching from an existing lane), then it is possible for the vehicle to exist in two lanes simultaneously. If this occurs, MM-LIM sets the vehicle actual/probable lane of travel to none.

MM-LIM determines the “Likelihood/Confidence” value for all lanes in the GID. If a lane has higher “Likelihood” rating than others then it is more certain that the vehicle is going to follow the approach of that particular lane. Depending upon the probable lane match the lane likelihood calculation differs. The following sections describe both cases:

a. **When there is a probable lane match:**

In order to determine the confidence value of probable lane of travel, MM-LIM calculates the distance to the vehicle centerline from the nearest lane edge, d_c as illustrated in Figure 56.

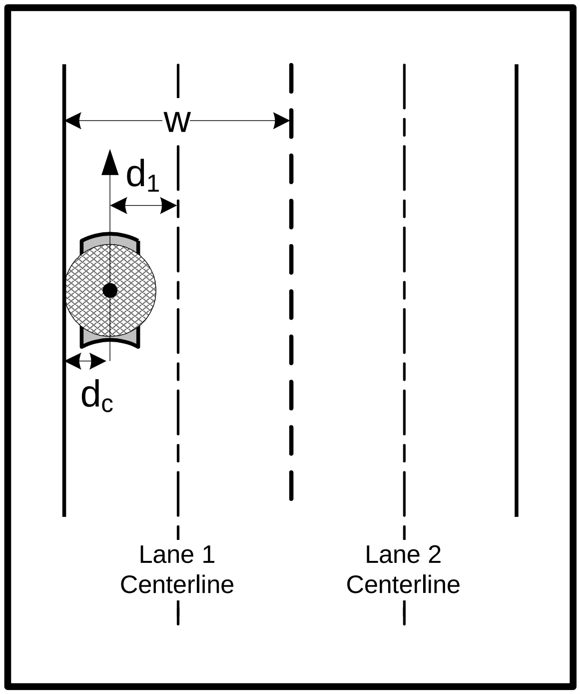

Figure 56: Distance to Vehicle Centerline

If w is the lane width of the nearest lane segment in GID and d_1 is the vehicle to segment distance, then distance to the vehicle centerline from the nearest lane edge, $d_c = (w / 2) - d_1$.

GPS data provides an estimate of error in reported latitude and longitude. The 1-sigma error ellipsoid is simplified to an error circle with radius, $d_e = \sqrt{(\text{latitude error})^2 + (\text{longitude error})^2}$. The following Figure 57 illustrates confidence interval around the vehicle:

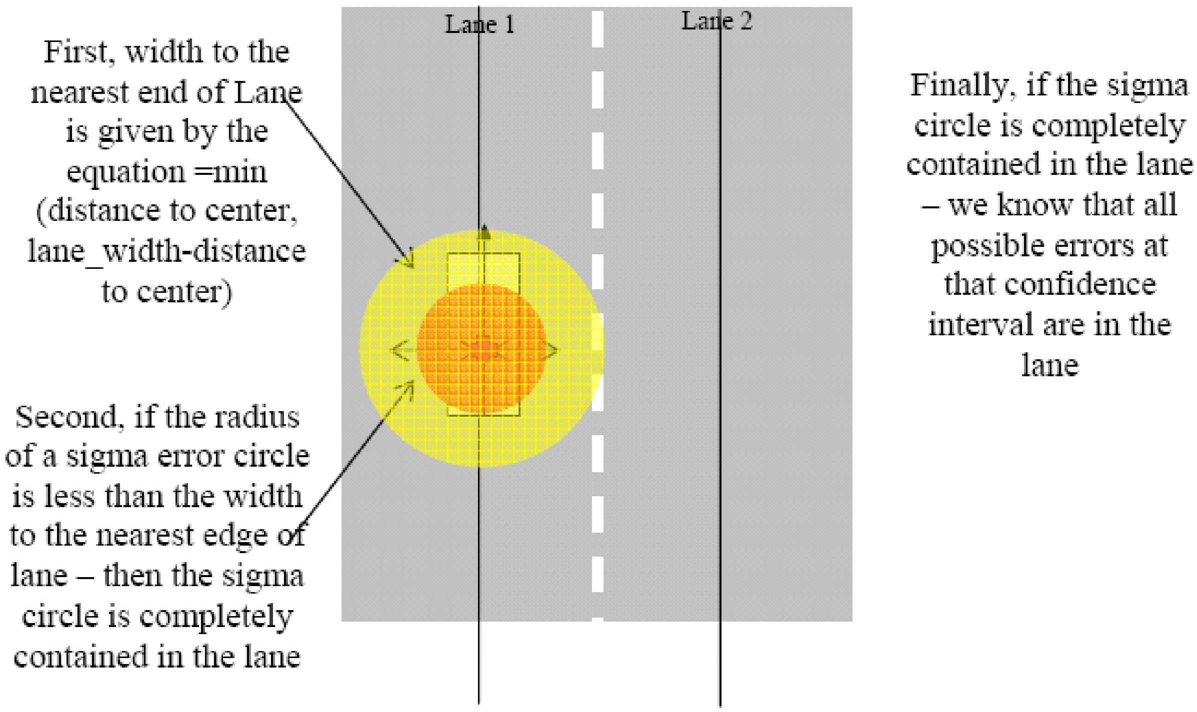

Figure 57: Confidence Interval Around Vehicle

If the distance to the edge of the lane is greater than the results from the error circle equation, then the vehicle is in the lane segment envelope defined by the lane segment being tested. In this case, the probable lane confidence should be reported as zero.

The confidence of the probable lane of travel in terms of a sigma number is, $\sigma = (d_c / d_e)$, it can also be represented in terms of percent by using the following equation:

$$\text{Confidence\_probable\_lane\_of\_travel} = (0.5 * (1 + \text{erf}(\sigma / \text{pow}(2.0,0.5)))) * 100$$

MM-LIM determines the confidence/Likelihood of all the lanes in the GID excluding the "probable" lane of travel through normalizing the inverse of the minimum distance to each lane:

Typically a set of values are normalized by using the method:

$$\frac{\text{Partial Value}}{\text{Sum of all partial values}} * (100)$$

Calculate the reciprocal for each lane because of when the distance to each lane segment becomes smaller then the probability of the lane match being correct becomes greater.

$$R_n = \frac{\text{Sum of distances to nearest lane segment for each lane} - \text{Distance to nearest segment}}{\text{Distance to nearest segment of lane}_n}$$

Where:

Rn = Reciprocal of $lane_n$.

Distance to nearest segment = Distance to the nearest segment for "probable" lane of travel.

Once the reciprocal is calculated for all the lanes excluding "probable" lane of travel for which confidence interval is already calculated, the values can be normalized over the percent range that has not already been assigned to the confidence interval associated with "probable" lane of travel.

$$Lk_n = \frac{\text{Calculated reciprocal for lane}_n}{\text{Sum of calculated reciprocals for all lanes}} \left(100 - \text{Confidence value}_{\text{Probable Lane of Travel}}\right)$$

Where:

Lk_n = is the "Likelihood" of $lane_n$

b. When there is no probable lane match:

MM-LIM considers the vehicle to be off GID, if the closest lane segment is greater than the configurable off GID threshold (nominally 4.0m). MM-LIM supports two methods for determining off-GID, selectable by a configuration parameter.

- Distance from centerline method – the minimum distance to any lane segment is greater than a configurable off GID threshold.
- Distance from lane edge method – the minimum distance to nearest lane edge is greater than a configurable off GID threshold.

If the vehicle is off GID, MM-LIM aborts further processing and sets its status to "off lane but on GID."

If the vehicle is not off GID, then MM-LIM determines the "Likelihood" for all lanes in the GID through normalizing the inverse of the minimum distance to each lane segment.

Calculate the reciprocal for all the lanes in the GID as the probability of lane match being correct is greater when the distance to each lane segment becomes smaller.

$$R_n = \frac{\text{Sum of distances to nearest lane segment for each lane}}{\text{Distance to nearest segment of lane}_n}$$

Where:

Rn = Reciprocal of $lane_n$.

Once the reciprocals have been calculated, the values can be normalized over 50 percent using the equation:

$$Lk_n = \frac{\text{Calculated reciprocal for lane}_n}{\text{Sum of calculated reciprocals for all lanes}} (50)$$

Where:

Lk_n = "Likelihood" of $lane_n$

Consolidate lanes based on "Likelihood" rating:

Once the "Likelihood" rating is available for all lanes in the GID, report the lane which has the "Likelihood" rating greater than the configurable threshold value (nominally 95%). If none of these lanes qualify, then report lanes with the maximum "Likelihood" rating. The number of lanes to report is based on the configurable parameter and a negative value indicates to report all lanes. Mapping between lanes and corresponding approach ID is done from information in GID database.

The following discussion depends upon these assumptions:

1. All lanes in every approach are parallel and can be traversed by the vehicle (no physical barriers between lanes).
2. All lanes in every approach have the same direction of travel.

Determine the number of approaches to be reported based on the lane matches:

- If a single lane has been identified with a greater "Likelihood" rating, the configurable threshold value or all lanes that have been identified share a common approach then report the approach.
- Report multiple approaches and corresponding "Likelihood" values if all the identified lanes correspond to different approaches. The number of lanes to be considered for reporting depends up on the configurable parameter.
- If two or more lanes identified share the same approach and these lanes are adjacent to each other as indicated by having continuous lane numbers (either increasing or decreasing), the reported "Likelihood" for this approach will be summation of the "Likelihood" of these lanes.
- If two or more lanes share the same approach and these lanes are not adjacent to each other as indicated by having discontinuous lane numbers, the reported "Likelihood" for this approach will be the maximum of these lanes "Likelihood" rating.

Filtering the approaches to be reported to the Warning Algorithm module:

1. If the highest likelihood approach is not above a configurable minimum likelihood approach threshold, do not report any approaches.
2. Report additional approaches if the difference between the highest likelihood approach and their likelihood is less than or equal to a configurable likelihood difference threshold.
3. The minimum and maximum likelihood threshold values used above differ based on the availability of local GPS corrections.

For each selected approach to be reported to warning algorithm, MM-LIM calculates the distance from vehicle location to the stop bar for the most likely lane of travel as illustrated in Figure 58.

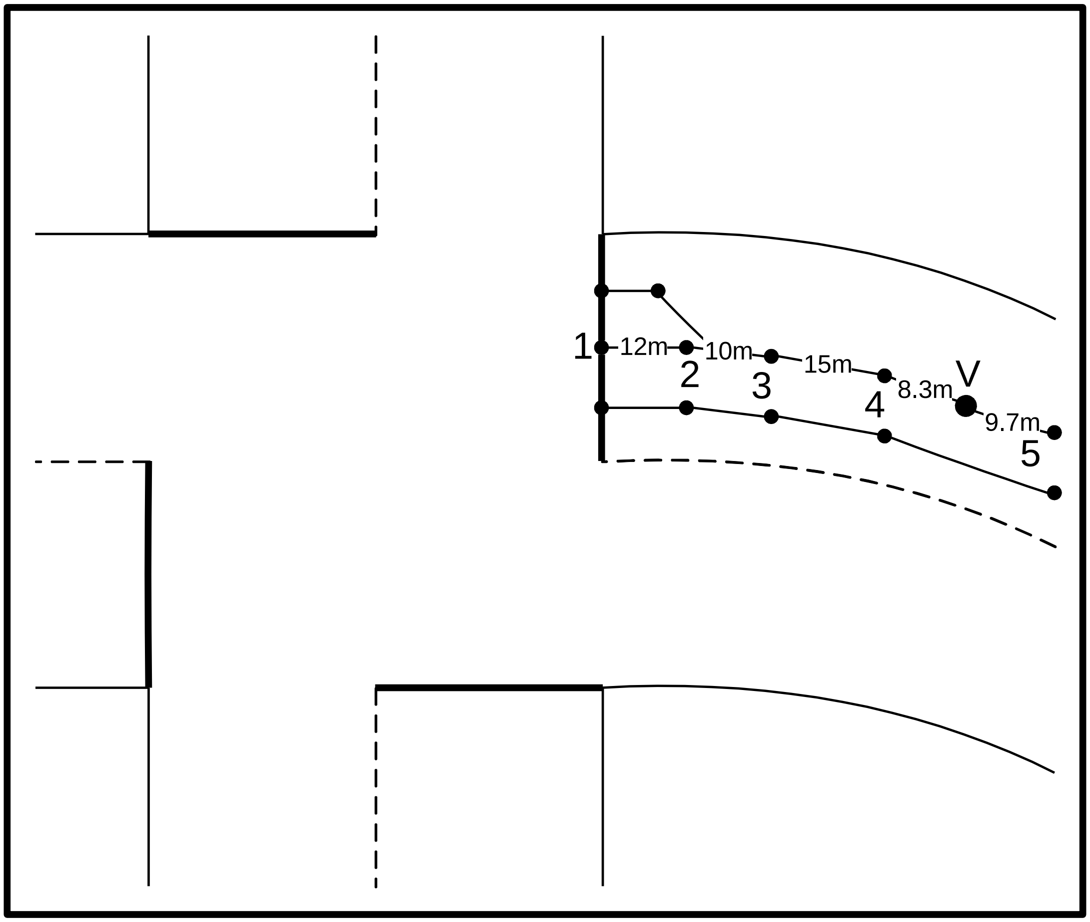

Figure 58: Distance to Stop Bar

The distance to stop bar is the distance along the most likely path from the vehicle location to the node point that represents the stop bar. The distance to stop bar in the illustration is V-4-3-2-1, which is 8.3+15+10+12 = 45.3m.

MM-LIM reports following information to Warning algorithm:

- For each approach reported:
    1. Approach ID
    2. Distance to stop bar
    3. "Likelihood"
- For the most likely lane:
    1. Lane ID
    2. Vehicle to segment distance (vsd)
    3. Distance to vehicle center line from nearest lane edge (dtvcl)
    4. Lane confidence/Likelihood
- Status of MM-LIM.

A.8.1.5 Warning Algorithm

A.8.1.5.1 Overview

The Warning Algorithm module (WA) executes periodically based on a configurable timer (nominally 100 ms).

A.8.1.5.2 Functional Description

Figure 59 illustrates the overall flow of the Warning Algorithm logic.

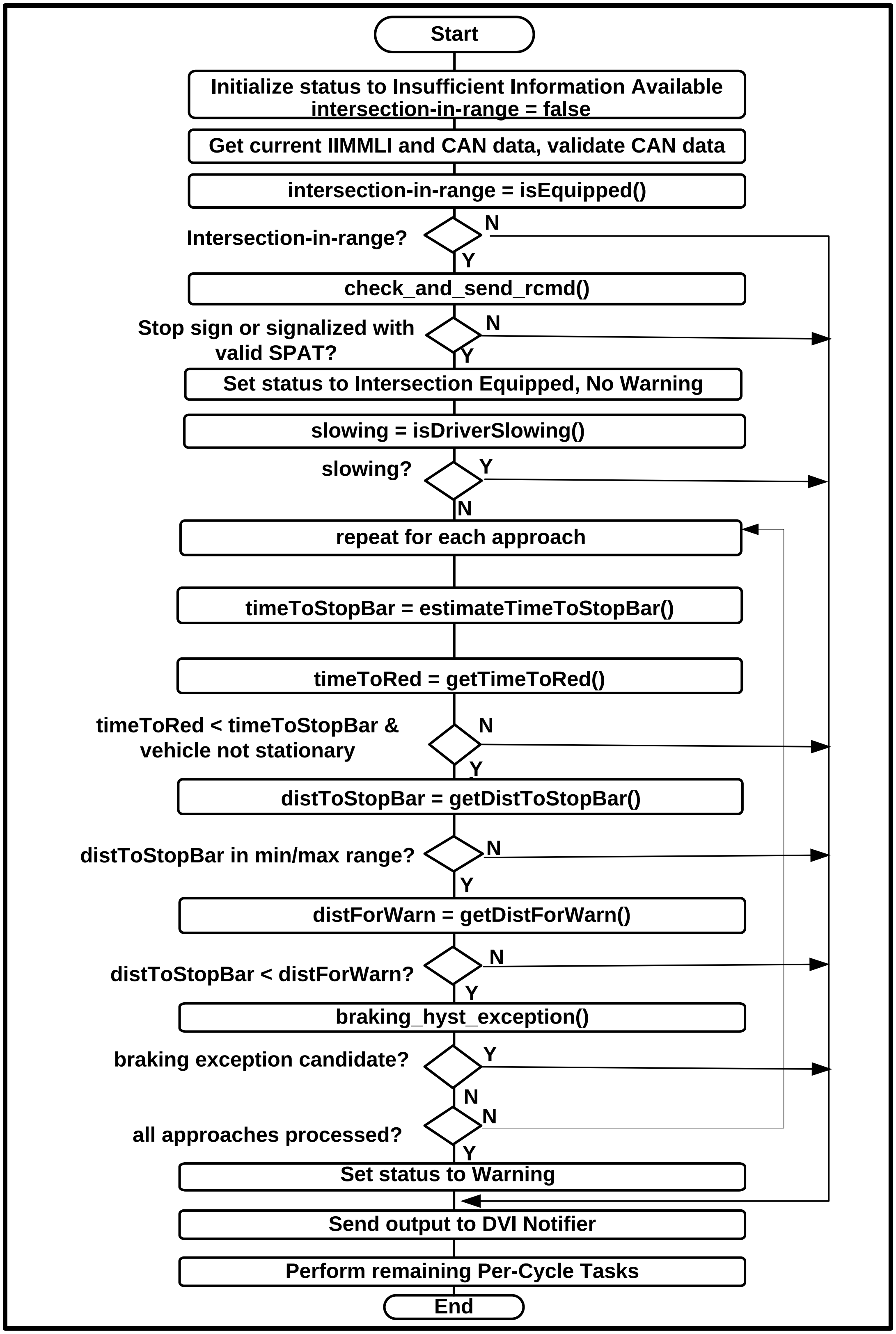

Figure 59: Warning Algorithm Processing

A.8.1.5.2.1 Intersection in Range Check

Figure 60 illustrates the processing for checking if an equipped CICAS-V intersection (signalized intersection or stop sign) is in range.

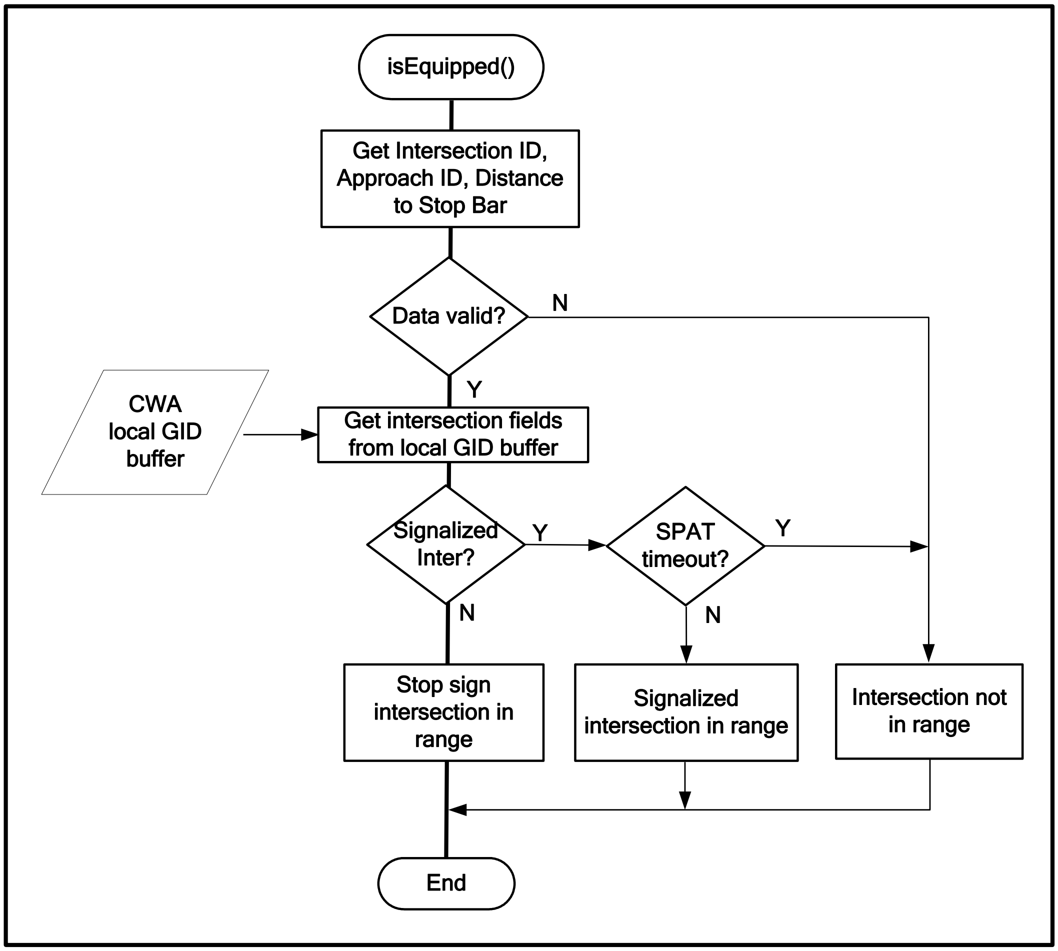

Figure 60: IsEquipped Processing

The isEquipped() function checks if the vehicle is approaching a CICAS-V equipped intersection. It performs the following processing:

1. Check if the IIM and MM-LIM have provided a valid approaching intersection ID and at least one approach. If not, set the intersection-in-range flag to false.
2. Otherwise, check the GID data to determine the type of intersection (signalized or stop sign).
3. For a stop sign intersection, set the intersection-in-range flag to true.
4. For a signalized intersection, check if SPaT data for the approaching intersection has been received from the CDIP within the configurable SPaT timeout applicable to the intersection-in-range indication. Note this timeout is different from and may be longer than the timeout used by the SPaT Handler to determine SPaT validity. Calculate the elapsed time since the last valid SPaT was received and compare it

with the timeout value. If the elapsed time is greater than the timeout value, set the intersection-in-range flag to false otherwise set it to true.

A.8.1.5.2.2 Check and Send RCMD Message

Figure 61 illustrates the processing for sending the RCMD message.

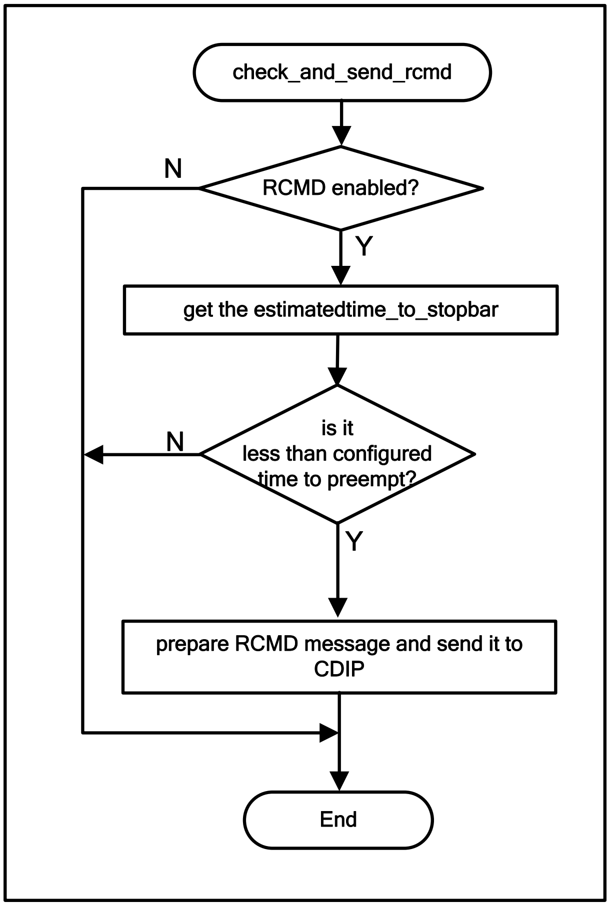

Figure 61: RCMD Transmit Processing

The check_and_send_rcmd function checks if the RCMD preempt is enabled, then checks the time to stop bar. If it is less than the configured time_to_preempt, then it sends the RCMD data to CDIP.

A.8.1.5.2.3 Vehicle Slowing Check

Figure 62 illustrates the processing for checking if the vehicle is slowing.

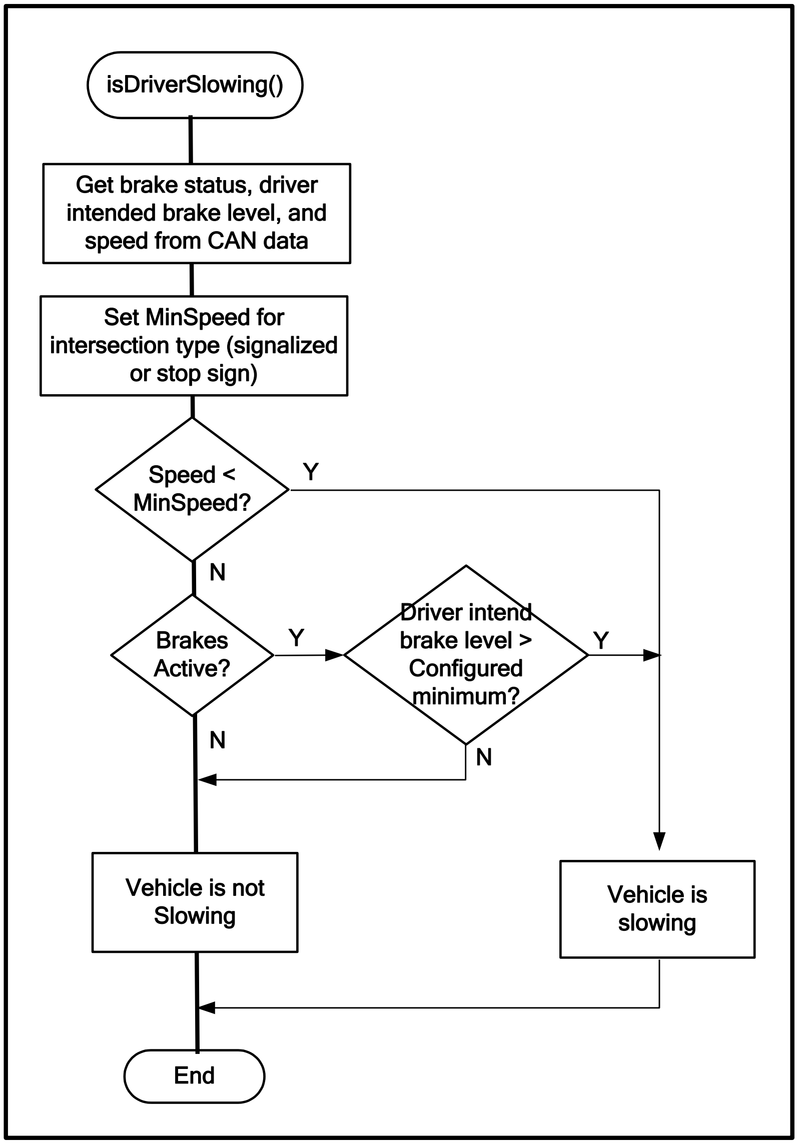

Figure 62: IsDriverSlowing Processing

The isDriverSlowing function checks if the vehicle is slowing down by checking the vehicle speed, brake status, braking time, and driver intended brake level. The vehicle is considered slowing if any of the following are true:

- The vehicle speed is below a minimum threshold. This check allows a CICAS-V vehicle to very slowly cruise through a red light without receiving a warning. This low speed will indicate the driver's attentiveness.
- The brakes are applied with a Driver intended brake level which is greater than the minimum configured value.

A.8.1.5.2.4 Repeating for each approach

The main Warning Algorithm Processing flow chart shows that the checks for "Time to Stop Bar" and beyond are done for each approach. This simply means that the Warning

Algorithm will process one or more approaches based on input from the Map Matching/Lane Identification process.

The Warning Algorithm processes the approaches given to it as above by the Map Matching/Lane Identification process. The first approach that does NOT result in a warning causes the process to quit from the loop without issuing a warning.

A.8.1.5.2.5 Time to Stop Bar Estimation

Figure 63 illustrates the logic for calculating the time it will take the vehicle to reach the intersection stop bar with unchanged dynamics (e.g., speed, acceleration, etc).

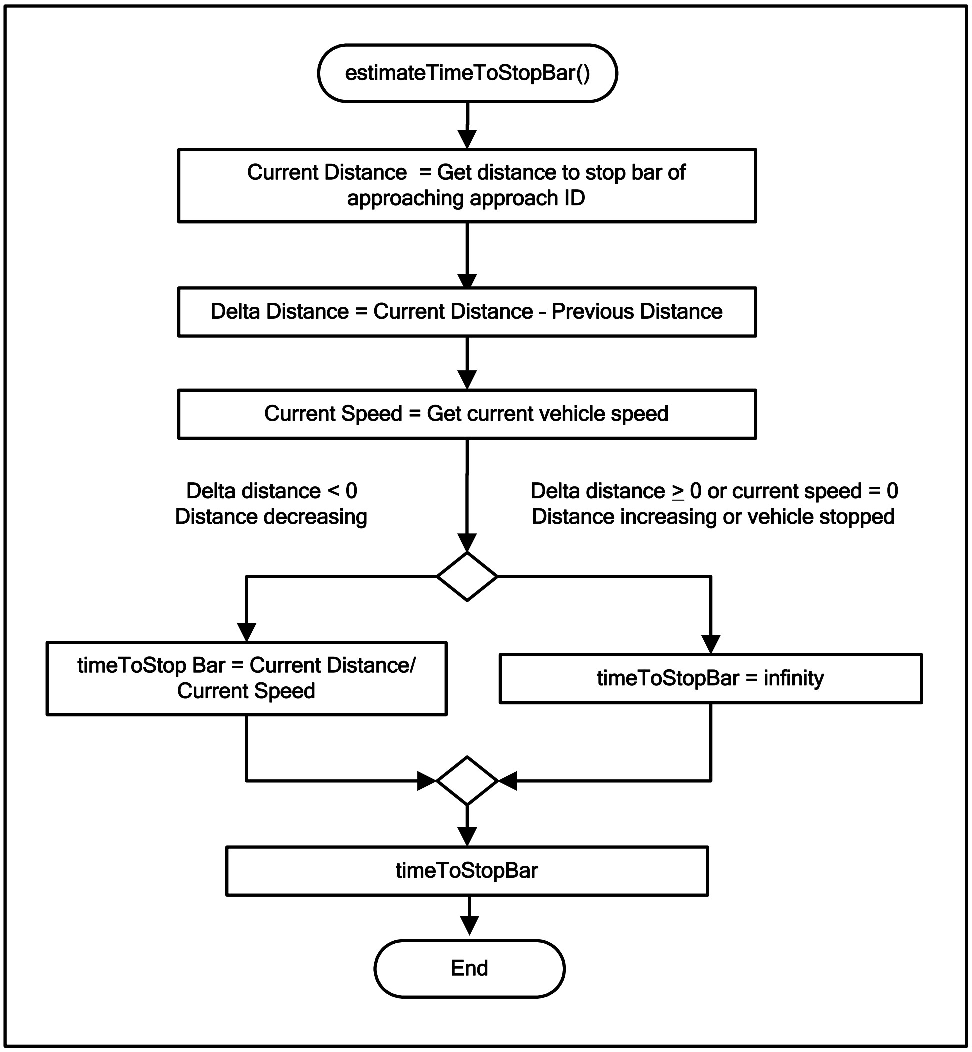

Figure 63: EstimateTimeToStopBar Processing

The estimateTimeToStopBar processing determines if the vehicle's distance to the stop bar is increasing or decreasing by calculating the difference between the current distance and the previous distance. If the distance is decreasing, the timeToStopBar is estimated as current distance divided by current speed. If it is increasing, not changing, or the vehicle's speed is 0, the timeToStopBar is estimated to be infinity since the vehicle is not approaching an equipped intersection.

A.8.1.5.2.6 Signal Time to Red Calculation

Figure 64 illustrates the logic for calculating the time it will take for the traffic signal to become red.

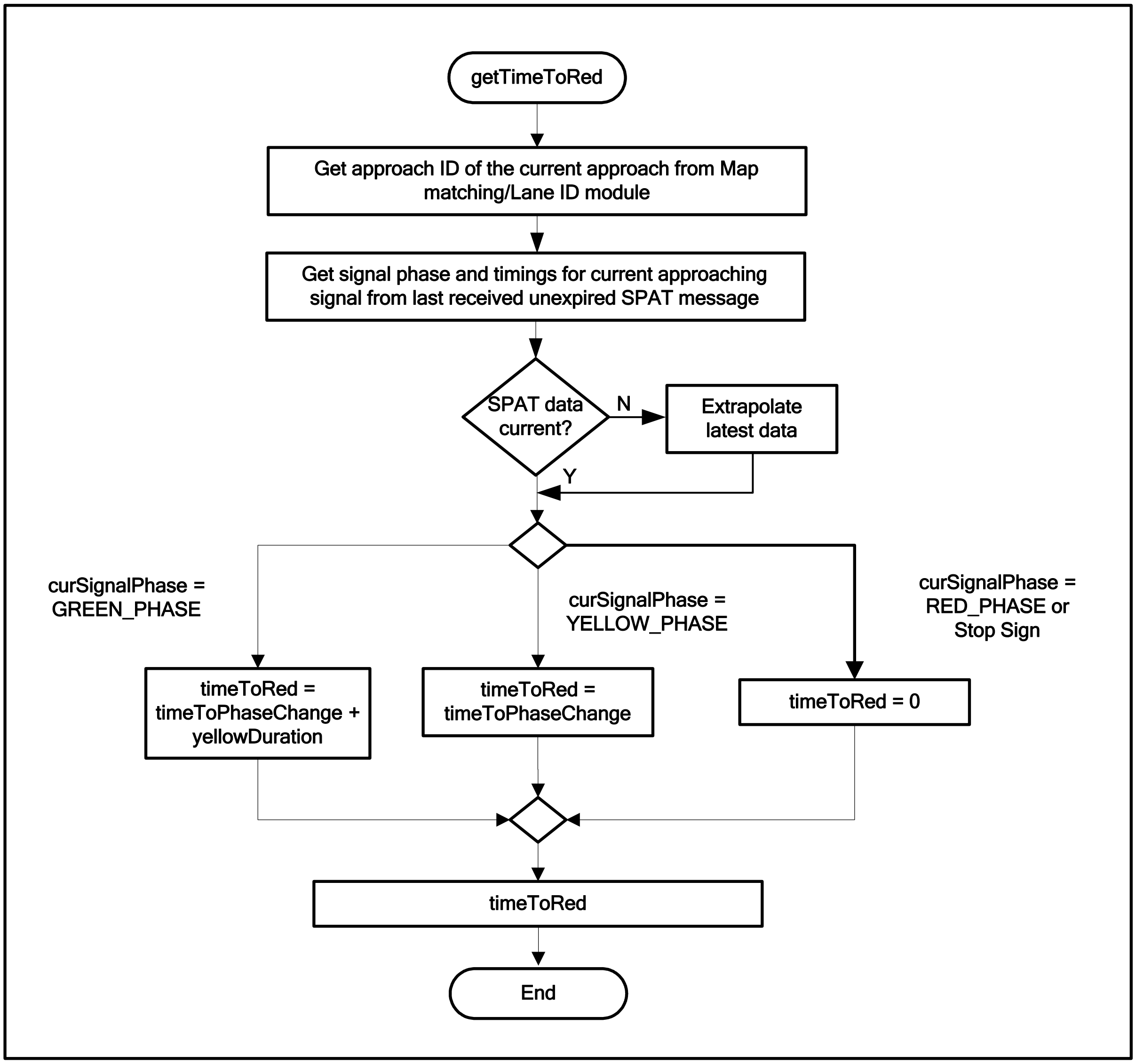

Figure 64: GetTimeToRed Processing

The getTimeToRed processing extracts the signal phase, yellow duration, yellow duration confidence, time to phase change and countdown timer confidence information for the current approach from the latest (and not yet expired) SPaT message. It ignores the yellow duration confidence and countdown timer confidence (since yellow duration and countdown time are know accurately by design). If the SPaT data is not current (i.e., some SPaT updates were missed), it estimates the current time to phase change by extrapolating from the latest information. It decrements the time to phase change by the elapsed time since the SPaT message was received.

TimeToRed is calculated as follows:

Signal Phase is GREEN

TimeToRed = timeToPhaseChange + yellowDuration

Signal Phase is YELLOW

TimeToRed = timeToPhaseChange

Signal Phase is RED

TimeToRed = 0

Approaching intersection is stop sign intersection

TimeToRed = 0

A.8.1.5.2.7 Distance to StopBar and DistanceForWarn Calculation

The distance to the stop bar is obtained from data provided by the Map Matching/Lane Identification module.

The distance for warn is calculated based on driver brake reaction time, vehicle stopping distance, and other system delay times (e.g., interface delay). These distances are provided in the form of a pre-defined array of minimum distances, indexed by the speed of the vehicle. There are separate arrays for stop sign intersections and for signalized intersections.

A.8.1.5.2.8 Braking Hysteresis Exception logic

Figure 65 illustrates the logic for calculating whether the vehicle is a candidate for the Braking Hysteresis exception of the warning algorithm.

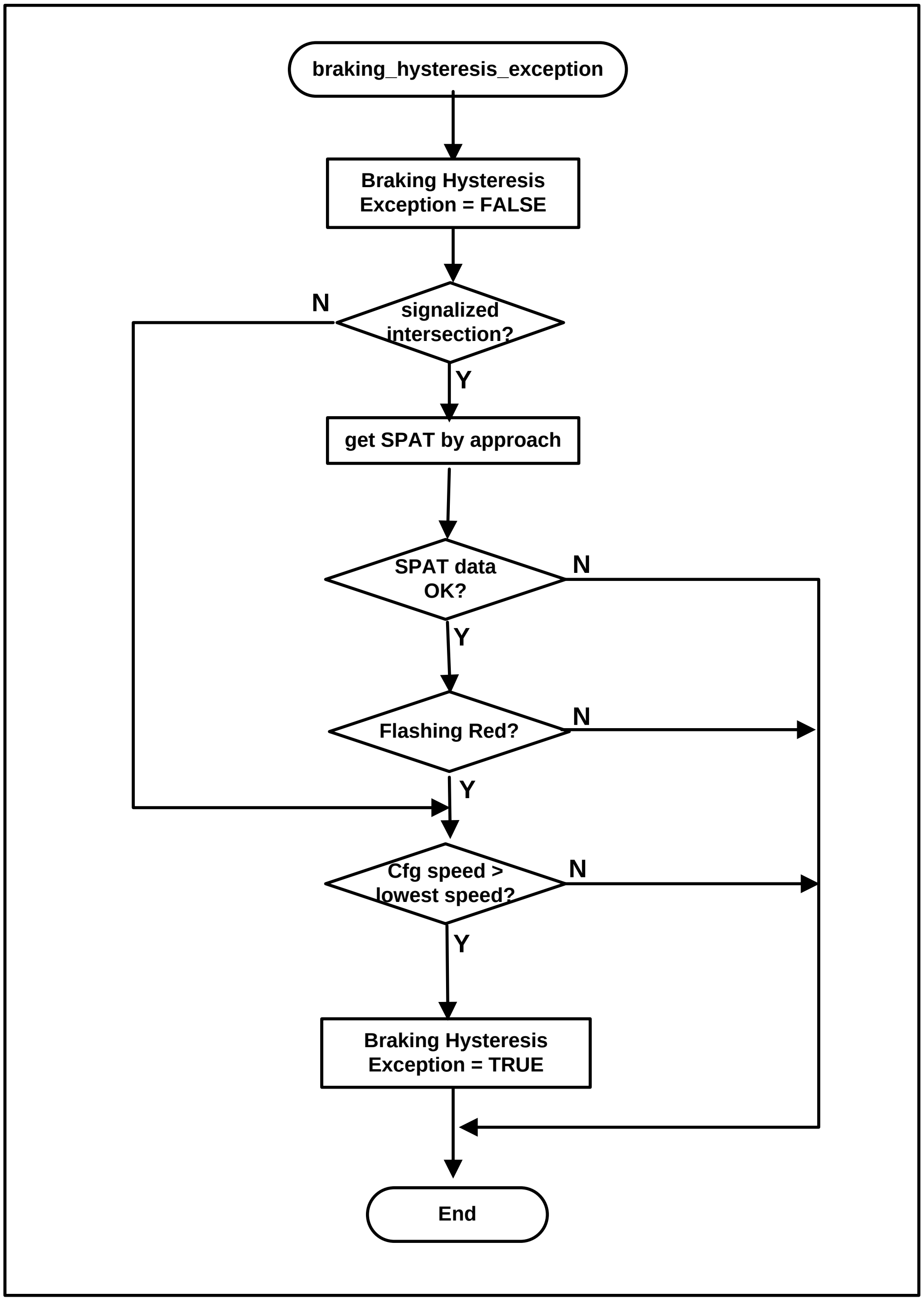

Figure 65: Braking Hysteresis Exception processing

A braking hysteresis exception causes a warning to be suppressed. For example, if the vehicle is approaching a flashing red light (a flashing red light is treated as a stop sign) and it has recently been below the configured speed threshold, the warning should not be given since the driver is aware of the situation around him/her. There is no added value in giving a warning.

The Warning Algorithm maintains an array of the recent speed of the vehicle. The array size is based on the configured hysteresis time and the execution interval of the Warning

Algorithm. On each run, the current speed is entered into an index position which is incremented for every run. When braking_hysteresis_exception() is called, if the intersection is a stop sign, or if it is signalized and has a flashing red light, the current lowest speed in the above array is checked against the configured hysteresis speed. If it is less, then the vehicle is given an exception.

A.8.1.5.2.9 Remaining Per-Cycle Tasks

At the end of the loop performed by the Warning Algorithm, several tasks are performed that are common to every execution of the loop:

1. If so configured and the status is Warning, send the TSVWG message.
2. Update the array containing the speed values for the hysteresis period.
3. Update the “previous” copy of the approach/distance data.
4. Update the local WA information data structure for this run of the algorithm loop.
5. Send data for the DAS to CLOGP.

A.9 DVI Notifier

Overview

The DVI Notifier performs the following functions:

1. Interface to the Driver Vehicle Interface (DVI) to control the DVI icon and input DVI status. Exchange heartbeat messages with the DVI.
2. Set the DVI icon based on the Warning Algorithm status, the OBE health status received from CLOGP and configuration parameters for each icon state (e.g., delay, keep high, keep low).
3. Upon receipt of a warning indication, set the flexible trigger outputs in a series of messages output to the CAN bus based on configuration parameters.
4. Output an audible warning upon a DVI state change based on user configurable parameters.

Interfaces

Figure 66 illustrates the DVIN interfaces with the WSU VIS and other CICAS-V processes.

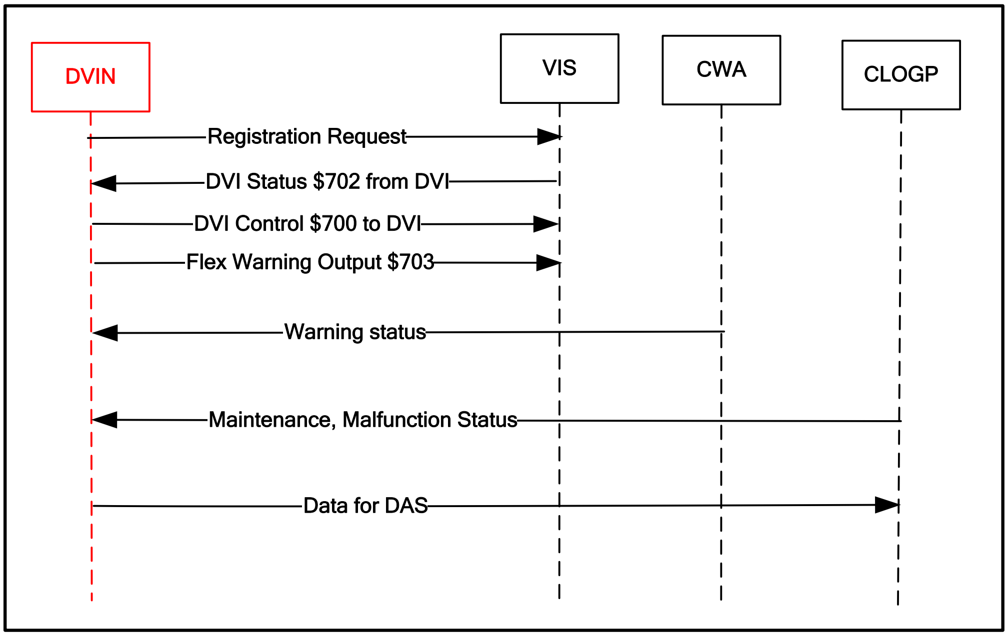

Figure 66: DVI Notifier Interface Diagram

Process structure

Figure 67 describes illustrates the DVIN thread decomposition and data flow. The Main thread executes upon receiving updated status from the Warning Algorithm (nominally every 100 ms). Based on the WA status, the OBE health status, and configuration parameters, it sets the DVI icon state and sends heartbeat status to the DVI through the WSU VIS. The Main thread also outputs data to CLOGP that is required for DAS output.

The CAN message listener thread waits for and processes heartbeat messages from the DVI (received via VIS).

The OBE health status listener thread waits for and processes OBE health status messages received from CLOGP. The messages indicate the current OBE maintenance and malfunction status.

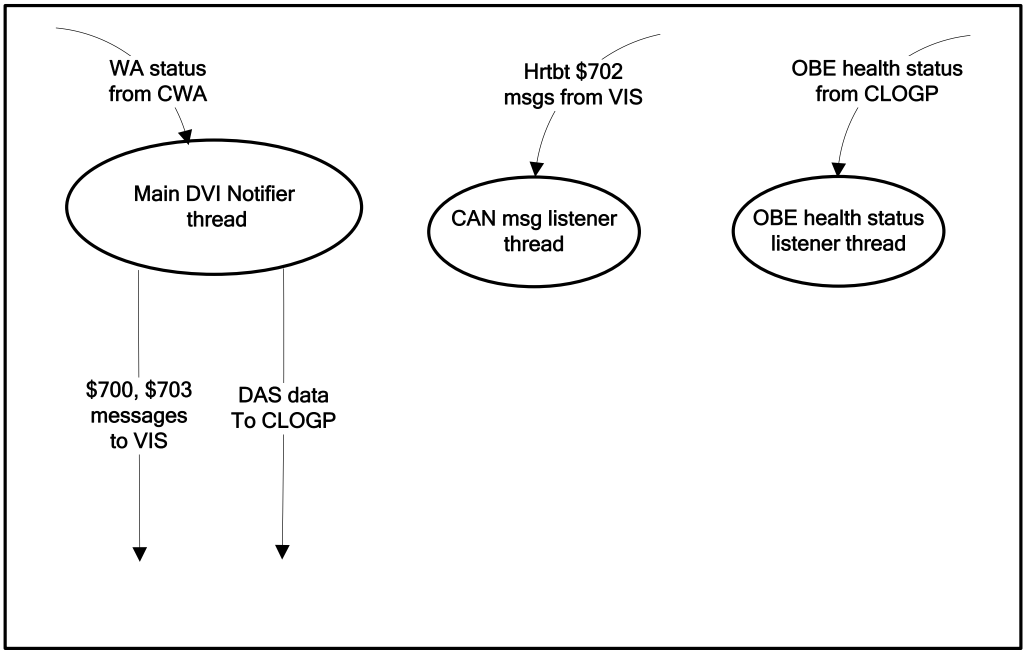

Figure 67: DVI Notifier Threads

Functional description

The following flowcharts describe the overall functionality of each of the above threads.

A.9.1.1 DVI Notifier Main/Warning Algorithm message handler thread

The DVI Notifier Main thread executes upon receiving a message from the Warning Algorithm. On initial startup, it does the following:

1. Reset the DVI state machine to its Standby state.
2. Set the flags for Malfunction, Maintenance, DVIN Reset, and Raw Warning to false.
3. Reset the counter for the OBE to DVI Heartbeat count, to 1.
4. Establish listeners for CAN and OBE health status messages.
5. Initialize message buffers for CAN message types $700 and $703.
6. Start the DVI heartbeat timeout timer of the configured duration.
7. Start to listen for messages from the Warning Algorithm.

Thereafter, the Main thread waits for incoming messages from the Warning Algorithm and processes state transitions as noted above. If there is an existing higher priority state transition ongoing, it will be the first to be completed.

At the end of the loop, the following common actions are done:

1. Increment the OBE to DVI heartbeat sequence number.
2. Update the time since the last warning.
3. Check for a transition to Maintenance and process it.

4. Decrement the DVI heartbeat countdown timer, and process if down to 0.
5. Complete prep of $700 message, and transmit it.
6. Play audio files that were triggered during this run, or are continuing.

Figure 68 illustrates the overall flow of the DVI Notifier Main thread.

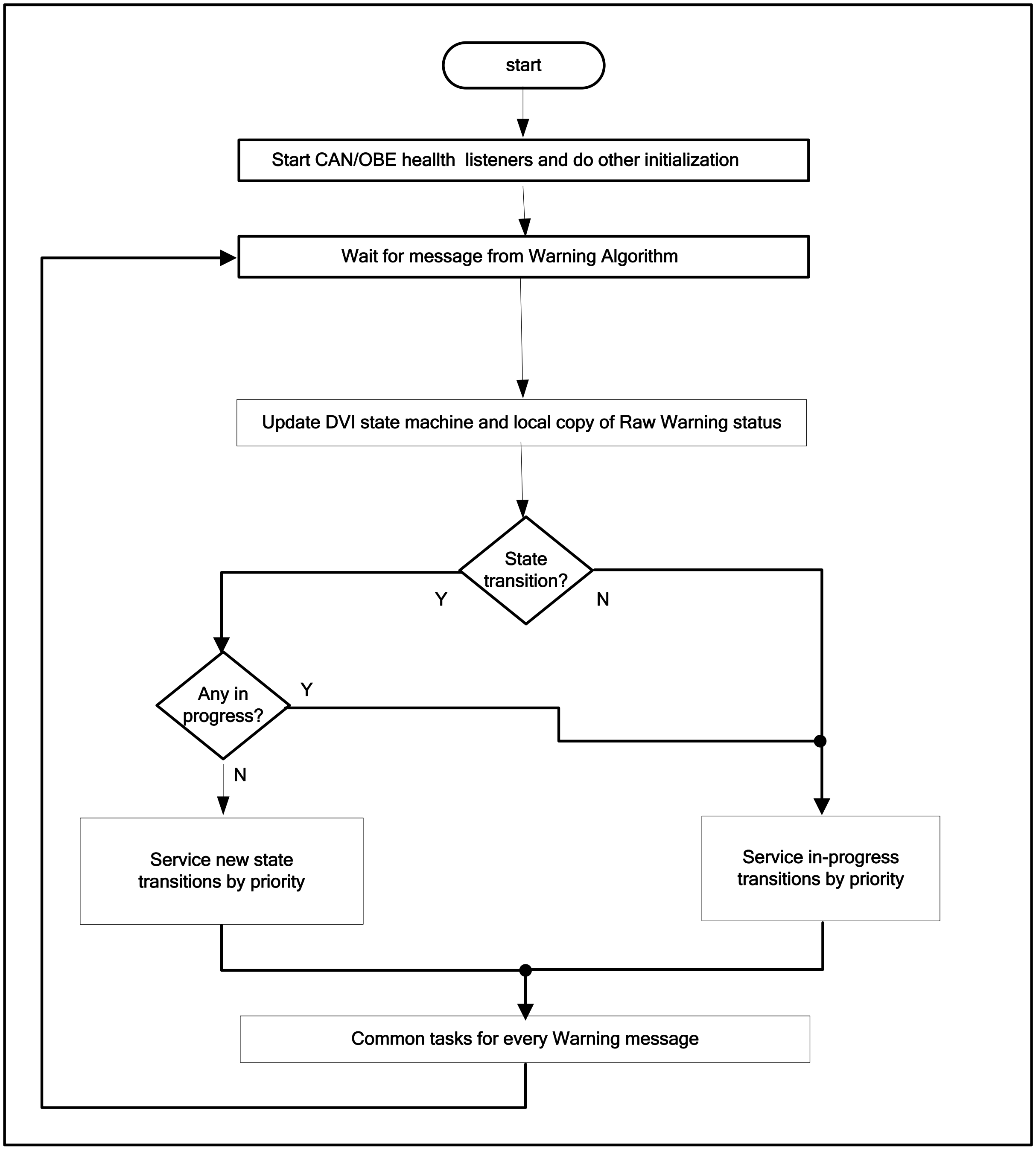

Figure 68: DVI Notifier Main Thread

A.9.1.2 CAN message $702 handler thread

On every CAN message $702 received by its listener, the CAN message handler thread performs the following processing:

1. Verify the heartbeat sequence number. If it is incorrect:
    - Send a message to the CLOGP to report the error status.
    - Set the DVI to OBE Heartbeat Error in the CAN message $700 buffer.
2. Else if the heartbeat sequence number is correct:
    - Clear the DVI to OBE Heartbeat Error in the CAN message $700 buffer.
3. Check the DVI System Error field in the message. If error:
    - Send a message to the CLOGP to report the error status.
4. Check the OBE to DVI Heartbeat Error flag in the message. If error:
    - Send a message to the CLOGP to report the error status.
5. Restart the heartbeat timeout timer with the configured duration.

Figure 69 illustrates the processing.

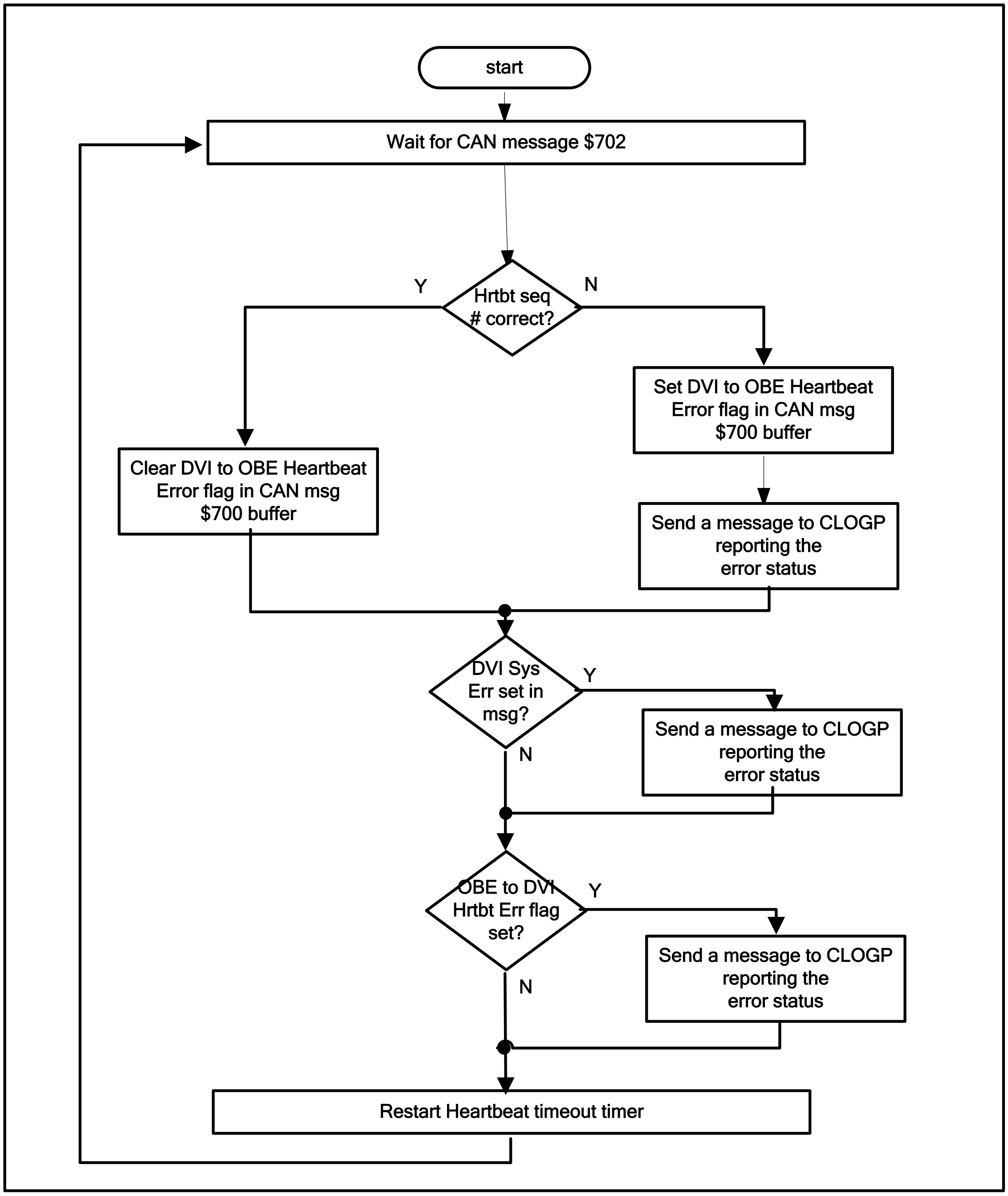

Figure 69: CAN Message $702 Handler Thread

A.9.1.3 OBE Health Status Listener thread

On every OBE health message received by its listener, the handler thread performs the following processing:

1. If the System Malfunction flag is set in the message:
    - Set the Malfunction status flag to true.
2. If the System Malfunction flag is cleared in the message:
    - Clear the Malfunction status flag.
3. If the System Maintenance flag is set in the message:
    - Set the Maintenance status flag to true.
4. If the System Maintenance flag is cleared in the message:
    - Clear the Maintenance status flag.

Figure 70 illustrates the processing:

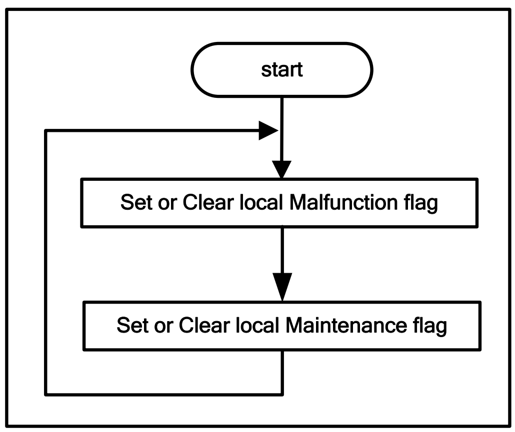

Figure 70: OBE Health Status Listener Thread

A.10 CICAS-V Logging Process (CLOGP)

Overview

The CICAS-V Log Process (CLOGP) performs the following functions:

1. Interface to other processes to receive data for logging and/or output to the DAS.
2. Output messages to the DAS (through the WSU VIS) on an event-driven basis for Metric Object data and at a 10 Hz rate for other data. The OBE-DAS CAN Interface Specification for CICAS-V (Appendix A.24) defines the format and content of the DAS messages.
3. Generate a log file in compact flash with the contents configurable by the user. Section 0 describes the format and content of the log file.

4. Maintain the OBE health status (normal, maintenance, or malfunction) based on error indications received from other processes.
5. Exchange heartbeat messages with the DAS and process DAS status.
6. Manage the hardware and software watchdog timers.

Interfaces

Figure 71 illustrates the CLOGP interfaces with the other CICAS-V processes.

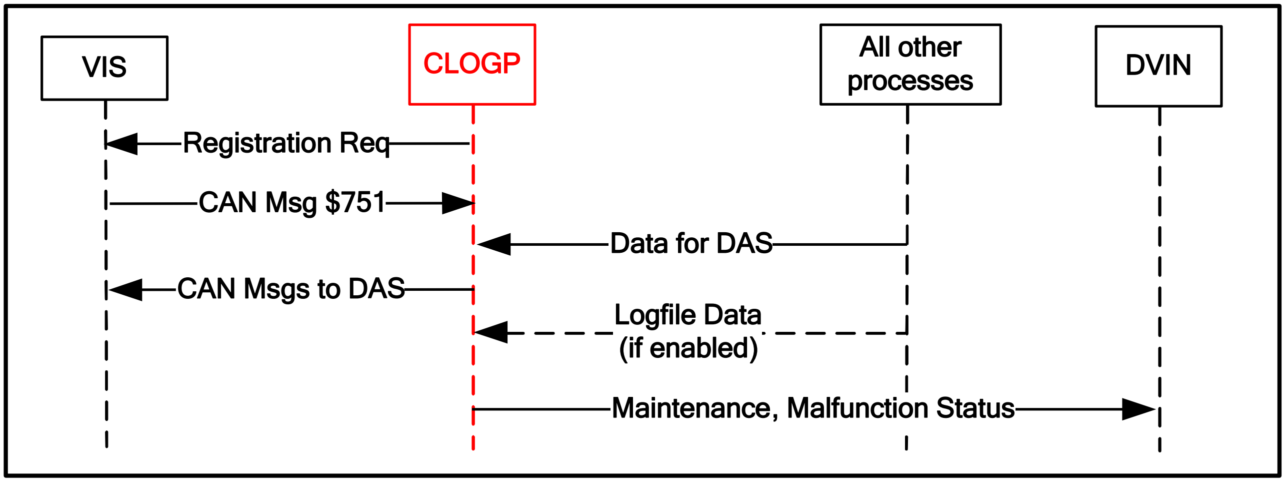

Figure 71: CLOGP Interface Diagram

Process Structure

Figure 72 illustrates the CLOGP thread decomposition and data flow. The CVIP Main registers with VIS to receive message $701 and provides a callback function pointer. It starts a 10 Hz DAS/watchdog timer to output the DAS messages and manage the HW/SW watchdog status. It starts a second timer to monitor for a $701 timeout. It then waits for input from other processes and processes received DAS and log entry data.

VIS calls the $701 callback function when it receives the $701 message. Upon receiving the data, the callback function processes the heartbeat and error indications in the message.

When the 10 Hz timer expires, the DAS/Watchdog Timer thread executes and outputs the contents of the DAS message buffer to the DAS through the WSU VIS. It resets the hardware and software watchdog timers if messages have been received from all other processes and logs an error if the configurable software watchdog interval has expired.

When the CAN Msg $701 timeout timer expires, CLOGP sets an error flag for output to the DAS in the $606 message and generates a log file entry if enabled.

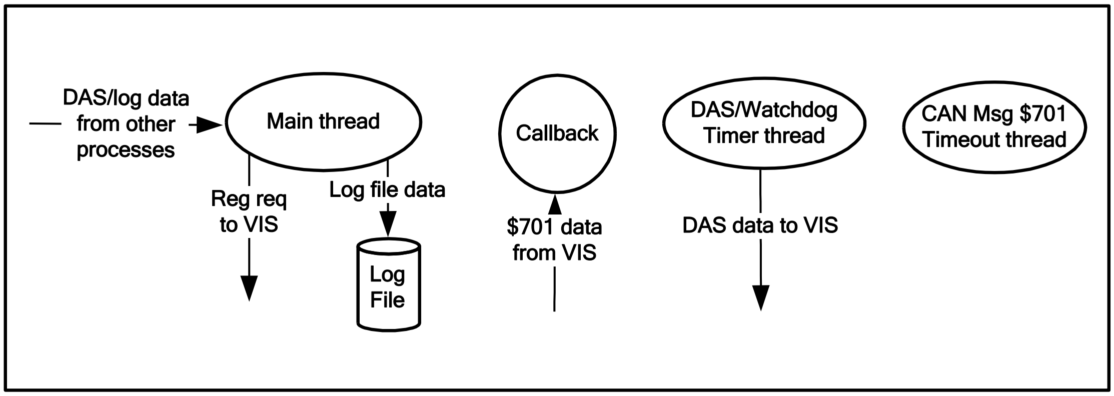

Figure 72: CLOGP Thread Decomposition

Functional Description

The following subparagraphs describe the processing for each of the threads.

A.10.1.1 CLOGP Main Thread

The CLOGP Main thread performs the following processing.

1. Create sockets to receive data from other processes.
2. Check the user configurable log mask to determine if a log file is required. If so, open a log file with name CICAS_LOG_yyyy_mm_dd_hh_mm_sec to ensure uniquely named log files for each execution of the application.
3. Set the DAS/Watchdog timer to execute at 10 Hz.
4. Set the $701 timeout timer to execute at 10 Hz.
5. Wait for input from other processes.
6. Upon receiving an input:
    - Check the log mask to determine if the type of data received should be output to the log file. If so, write the data to the log file.
    - If the data is included in the DAS output, check if the data is Metric Object data. If so, output the data immediately in at $751 message. If not, update the DAS message buffer with the data.
    - Update the software watchdog status to indicate a message has been received from the source process.
    - If the data contains an error set/clear indication, update the set/clear persistence time for the error. If the persistence is greater than a configurable threshold, update the OBE health status. The severity of each error type is configurable (maintenance or malfunction). Set the OBE health status to the highest severity active error. If the OBE health status has changed, notify the DVIN process.
7. Return to Step 5.

Figure 73 and Figure 74 illustrate the processing.

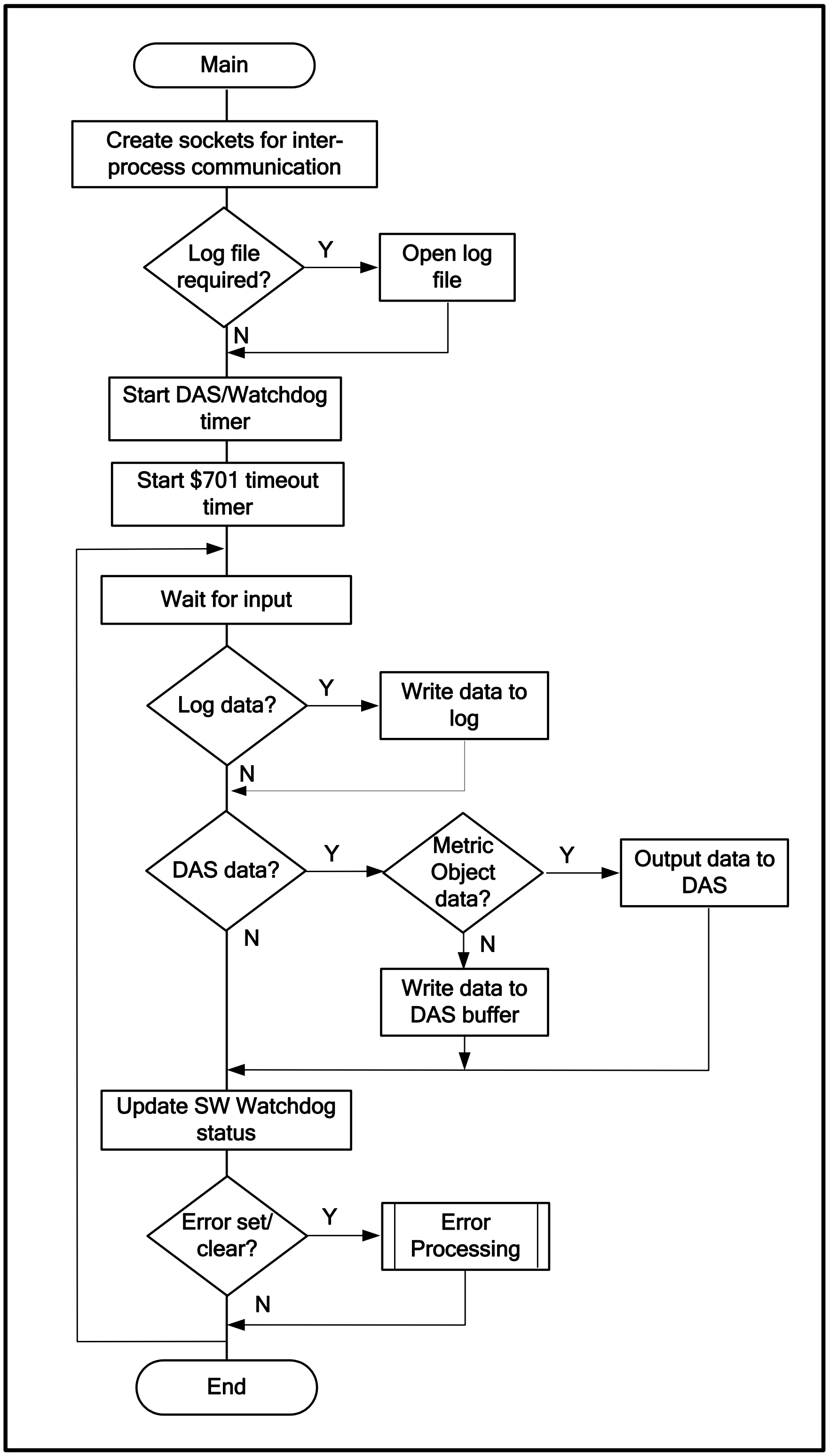

Figure 73: CLOGP Main Thread Processing

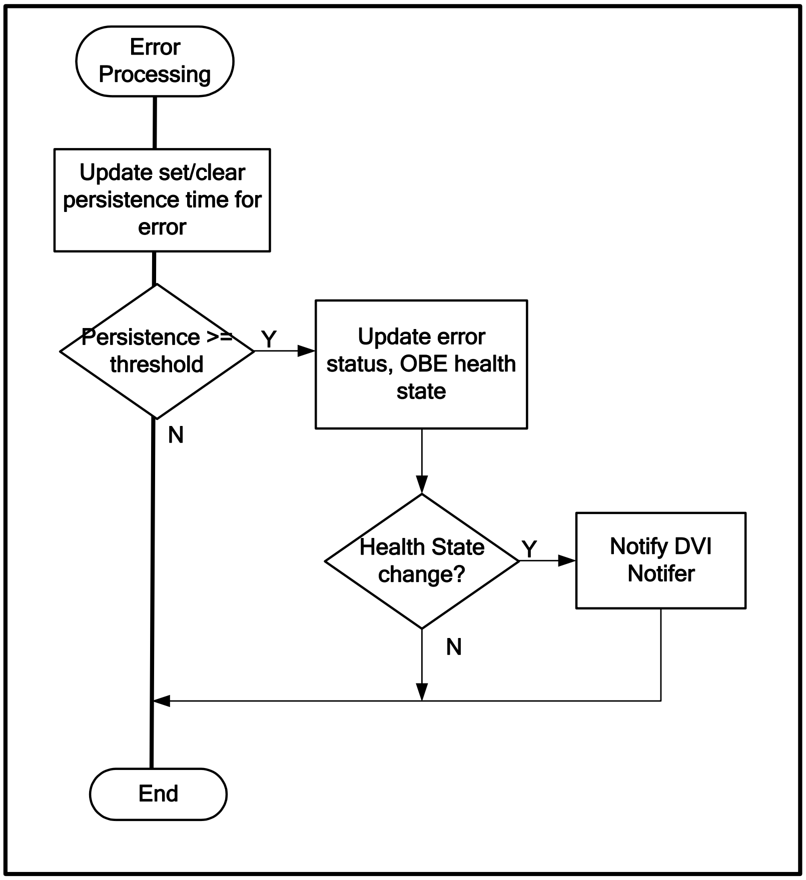

Figure 74: CLOGP Main Thread Error Processing

A.10.1.2 CAN $701 Callback

The WSU VIS calls the CAN $701 callback function when it receives the $701 message. The callback function performs the following processing:

1. Check the DAS heartbeat sequence number to determine if it is correct. If not, generate a log file entry (if enabled) and set the corresponding error bit in the $606 message.
2. Process the DAS error indications and update the OBE health status.
3. Reset the CAN $701 timeout timer.

A.10.1.3 DAS/Watchdog Timer Thread

When the DAS/Watchdog Timer thread executes, it outputs the contents of the DAS message buffer to the DAS via the WSU VIS. It also checks if messages have been received from all of the other CICAS-V processes. If so, it resets the hardware and software watchdog timers. If messages have not been received from all of the processes, and the configurable software watchdog duration has expired, it writes an error to the log file.

A.10.1.4 CAN $701 Timeout Thread

Upon expiration of the $701 timer, the timeout thread logs an error and sets an error bit in the CAN $606 message.

Log File Format and Contents

CLOGP writes the log file as an ASCII file. Each log entry is a separate row, with the parameters written as comma-separated variables. The log file is event driven (i.e., upon receipt of data, CLOGP timestamps the data and writes it to the log). The DAS messages provide a periodic snapshot of the system.

Appendix 0 specifies the log entries the software supports. The log mask determines the actual log file contents for any execution of the application. Many of the log entries contain data that duplicates information in the DAS output. This duplication enables the log file to be analyzed on a stand-alone basis.

OBE ATP Details

The ATP was run in the laboratory under controlled conditions with a simulated test environment:

- Vehicle motion was simulated by creating GPS Sequence data using a GPS Generator tool
- A GPS Simulator running on a Linux PC transferred GPS NMEA data to the WSU via a serial interface
- SPaT information to be transmitted by a RSE was simulated by creating SPaT Message Sequence data using a SPaT Generator tool
- The SPaT Message Sequence data, as well as GID and GPSC data, was supplied to a RSE Simulator
- The RSE Simulator ran on a Linux PC and transmitted WSMs containing the GPSC, GID, and SPaT messages

The laboratory test setup used for the ATP is illustrated in Figure 75. An alternate setup used for some tests consisted of playing back a recording using a Scenario Replicator tool which was developed as a part of another program.

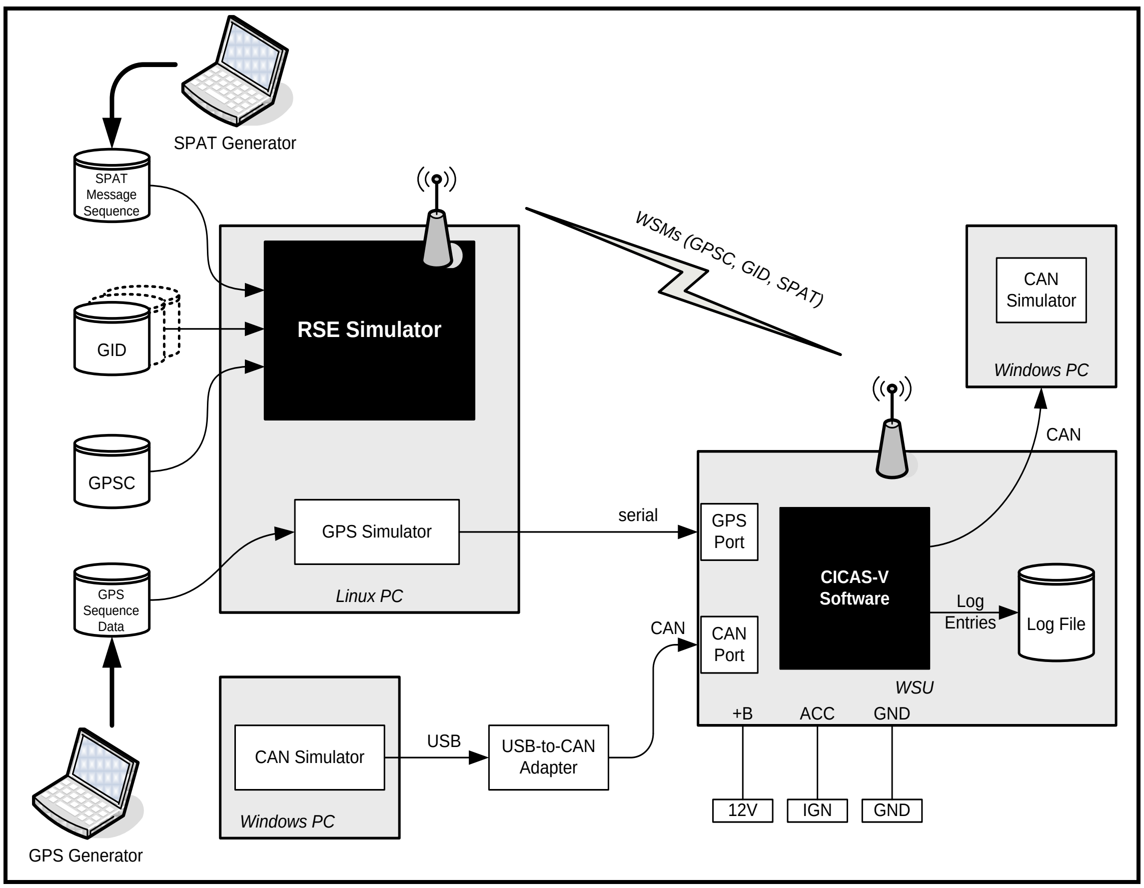

Figure 75: CICAS-V OBE ATP Test Setup

As part of the pre-test setup, use the cicas-v.conf file that was included in the final WSU Release 1.15 as the default Configuration Parameter File. Changes to this configuration file may be necessary to set up certain tests. When a test procedure specifies making changes to cicas-v.conf, the differences from the default file are expected to be only those specified in the test procedure.

A default configuration for A CAN device simulating the Netway box will be created and used for most tests. When a test procedure calls for configuration changes to the CAN application simulating the Netway box, the differences from the default configuration are expected to be only those specified in the test procedure.

Note: In the test results tables to follow, tests where the actual value of the test did not meet the expected value are highlighted in red.

A.11 Interface Tests

The tests in this section verify the requirements specified in the CICAS-V Software Specification Appendix A.1.

DVI Interface

The DVI Interface requirements are verified in A.19.

Netway Box Interface

The Netway Box Interface requirements are verified in 0.

GPS Interface

The GPS Interface requirements are verified in A.14.

WSU Interface

The WSU Interface requirements are verified in A.13.

DAS Interface

The DAS Interface requirements are verified in A.17.

Speaker Interface

The Speaker Interface requirements are verified in A.19.

A.12 Vehicle Message Handler

The tests in this section verify the requirements specified in Appendix 0 of the CICAS-V Software Specification.

CAN Heartbeat Tests

The tests in this section verify CAN handshaking and heartbeat requirements are satisfied.

Assumption:

Per Appendix A.22 the Netway box transmits CAN message $606 when it receives the $704 message. As such, the WSU expects to receive one CAN message $606 within a few ms following each $704 message it transmits.

A.12.1.1 Test Procedure

Req. #	Procedure
Pretest	• Reset the CAN application simulating the Netway box to the default configuration and start CAN message transmission. • Ensure cicas-v.dflt provides appropriate MIN and MAX values for configuration parameters being modified in these tests, as indicated in Table 36. • Ensure the cicas-v.conf parameters are reset to the default values, and then enable the log flags indicated in Table 37. • Boot the WSU.
10	• In cicas-v.conf, set the parameters listed in Table 38 to the values indicated. • Start the CICAS application. • Observe the period displayed in the CAN simulator for CAN message $704 to verify it is being transmitted at the configured interval. Record the results in Table 41.
	• In cicas-v.conf, set the parameters listed in Table 39 to the values indicated. • Restart the CICAS application. • Observe the period displayed in the CAN simulator for CAN message $704 to verify it is being transmitted at the configured interval. Record the results in Table 42.
11	• Stop the CICAS application. • Configure the CAN application simulating the Netway box to record CAN $704 messages that are received. • Start the CICAS application. • Stop recording CAN $704 messages on the CAN simulator after approximately 5 seconds. • View the recorded CAN $704 messages to verify the Heartbeat Sequence Counter in the first CAN message $704 received was set to 1. Record the results in Table 43.
12	• View the recorded CAN $704 messages to verify the Heartbeat Sequence Counter is incremented by 1 for each CAN $704 message received. Record the results in Table 43. • Close the recorded trace log in the CAN simulator.
14	• Ensure the CAN application simulating the Netway box is configured to send CAN message $606 with Netway to OBE Heartbeat Sequence Counter that does not match the last number sent in CAN message $704. • Stop the CICAS application and check the Log File to verify a CAN 606 Sequence Error is logged. Record the results in Table 44. • Delete all CICAS log files on the WSU.
Reset Config	• Reset the CAN application simulating the Netway box to the default configuration. • Start the CICAS application and allow it to run for at least 5 seconds.
15	• On the CAN simulator, pause transmission of CAN message $606 to force a situation where the WSU will send CAN message $704 but not receive CAN message $606 before sending the next CAN message $704. • Stop the CICAS application and check the Log File to verify a CAN 606 Expired error was logged. Record the results in Table 44. • Delete all CICAS log files on the WSU.

Req. #	Procedure
13	• In the cicas-v.conf, set the parameters listed in Table 40 to the values indicated and enable logging flags for the parameters listed in Table 44. • In the CAN simulator, set the OBE to Netway Heartbeat Error flag in CAN message $606. • Start the CICAS application and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify the CAN 606 Heartbeat Error was logged. Record the results in Table 44. • Delete all CICAS log files on the WSU.

Table 36: MIN/MAX Configuration Values

Parameter	MIN	MAX
CAN704TxInterval	100ms	500ms

Table 37: CAN Heartbeat Test Logging Parameters

Parameter	Value
GlobalLogFlag	1
CAN606ExpdLogFlag	1
CAN606HBErrLogFlag	1
CAN606SeqErrLogFlag	1

Table 38: 10Hz CAN message $704 Tx Interval

Parameter	Value
CAN704TxInterval	100ms
CANExpirationTime	400ms

Table 39: 5Hz CAN message $704 Tx Interval

Parameter	Value
CAN704TxInterval	200ms
CANExpirationTime	400ms

Table 40: OBE to Netway Heartbeat Error

Parameter	Value
CAN704TxInterval	500ms
CANExpirationTime	400ms

Table 41: Requirement 10 – 10Hz Test Results

Parameter	Expected Time Between Messages	Actual Time Between Messages
$704 CAN messages	100ms	103ms

Table 42: Requirement 10 – 5Hz Test Results

Parameter	Expected Time Between Messages	Actual Time Between Messages
$704 CAN messages	200ms	203ms

Table 43: Requirements 11 & 12 Test Results

Parameter	Sent CAN message $704	Expected Results	Actual Results
Heartbeat Sequence Counter value	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	5

Table 44: Requirements 13-15 Test Results

Test #	DAS / Log File Parameter	Expected Results	Actual Results
14	CAN 606 SEQ Error	Logged	Logged
15	CAN 606 Expd	Logged	Logged
13	CAN 606 HB Error	Logged	Logged

CAN Message Tests

The tests in this section verify CAN message errors are correctly logged to the Log File and CAN bus messages $600-$605 and $650 received by the WSU are correctly reported to the DAS.

A.12.1.2 Test Procedure

Req. #	Procedure
Pretest	• Ensure the cicas-v.conf parameters are reset to the default values, and then enable the log flags indicated in Table 45. • Reset the CAN application simulating the Netway box to the default configuration and start CAN message transmission.
17	• In the CAN simulator, configure CAN messages $601-$605 to be transmitted at 500ms intervals. • Start the CICAS application and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify a CAN Data Expiration error was logged. Record the results in Table 46. • Delete all CICAS log files on the WSU.
16	• Configure the CAN application simulating the Netway box to send CAN message $606 with Vehicle CAN Data Timeout set for CAN1, ($606 Byte-0, Bit-7). • Start the CICAS application and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify a Vehicle CAN Data Timeout error was logged. Record the results in Table 46. • Delete all CICAS log files on the WSU.
Reset Config	• Reset the CAN application simulating the Netway box to the default configuration.
19	• Configure the CAN application simulating the Netway box to send CAN messages $601-$605, but not send CAN message $600. • Start the CICAS application and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify the Incomplete CAN Data Received error is logged indicating CAN message $600 is missing. Record the result in Table 47. • Check the Log File to verify an CAN Data Received is not logged, indicating all expected CAN messages have not been received. Record the result in Table 47. • Repeat this test five additional times configuring CAN messages $601-605 in turn to each not be sent.
Reset Config	• Reset the CAN application simulating the Netway box to the default configuration.

Req. #	Procedure
20 22	• Configure the CAN application simulating the Netway box to send all 0’s for CAN messages $600-$605. • Start the CICAS application and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify CAN Data Received is logged, indicating all expected CAN messages have been received. Record the result in Table 48. • Check the Log File to verify an Incomplete CAN Data Received error is not logged. Record the result in Table 48. • Delete all CICAS log files on the WSU. • Restart the CICAS application. • View the CAN application simulating the DAS to verify the complete group of CAN messages $600-$605 was written to the DAS. Record the result in Table 49. • Configure the CAN application simulating the Netway box to send all A’s for CAN messages $600-$605. • View the CAN application simulating the DAS to verify the complete group of CAN messages $600-$605 was written to the DAS. Record the result in Table 49. • Configure the CAN application simulating the Netway box to send all F’s for CAN messages $600-$605. • View the CAN application simulating the DAS to verify the complete group of CAN messages $600-$605 was written to the DAS. Record the result in Table 49.
Reset Config	• Reset the CAN application simulating the Netway box to the default configuration.
21 22	• Configure the CAN application simulating the Netway box to send all 0’s for CAN messages $650. • Ensure the CICAS application is running on the WSU. • View the CAN application simulating the DAS to verify CAN message $650 was written to the DAS. Record the result in Table 50. • Configure the CAN application simulating the Netway box to send all A’s for CAN messages $650. • View the CAN application simulating the DAS to verify CAN message $650 was written to the DAS. Record the result in Table 50. • Configure the CAN application simulating the Netway box to send all F’s for CAN messages $650. • View the CAN application simulating the DAS to verify CAN message $650 was written to the DAS. Record the result in Table 50.

Table 45: CAN Message Test Logging Parameters

Parameter	Value
GlobalLogFlag	1
CANExpdLogFlag	1
VehCANDataTmoLogFlag	1
CANDataRxLogFlag	1
IncompCANDataRxLogFlag	1

Table 46: Requirements 16 & 17 Test Results

Test #	Log File Parameter	Expected Results	Actual Results
17	CAN Expd	1	Logged
16	Vehicle CAN Data Timeout	1	Logged

Table 47: Requirement 19 Test Results

Missing CAN message	Log Entry	Expected Value	Actual Value
$600	Incomplete CAN Data Rx	$600	600
	CAN Data Rx	Not Logged	Not Logged
$601	Incomplete CAN Data Rx	$601	601
	CAN Data Rx	Not Logged	Not Logged
$602	Incomplete CAN Data Rx	$602	602
	CAN Data Rx	Not Logged	Not Logged
$603	Incomplete CAN Data Rx	$603	603
	CAN Data Rx	Not Logged	Not Logged
$604	Incomplete CAN Data Rx	$604	604
	CAN Data Rx	Not Logged	Not Logged
$605	Incomplete CAN Data Rx	$605	Not Logged
	CAN Data Rx	Not Logged	Not Logged

Table 48: Requirement 20-22 Log File Test Results

Log Entry	Expected Value	Actual Value
CAN Date Rx	Logged	Logged
Incomplete CAN Data Rx	Not Logged	Not Logged

Table 49: Requirement 20-22 DAS Test Results

CAN message ID	Expected Value	Actual Value	Expected Value	Actual Value	Expected Value	Actual Value
$600-$605	All 0's	All 0's	All A's	All A's	All F's	All F's

Table 50: Requirement 21-22 DAS Test Results

CAN message ID	Expected Value	Actual Value	Expected Value	Actual Value	Expected Value	Actual Value
$650	All 0's	All 0's	All A's	All A's	All F's	All F's

A.13 Radio Handler / Data Demux Tests

Test Procedure

Configure the DAS Log Entries to include GPSC, GID, and SPaT data received in a TOM.

Req. #	Procedure
Pretest	• Reset the CAN application simulating the Netway box to the default configuration and start transmitting CAN messages. • Ensure the cicas-v.conf parameters are reset to the default values, and then enable the log flags indicated in Table 51. • Boot the WSU and start the CICAS application.
25	• Configure the RSE Simulator to transmit a TOM WSM with an unsupported Framework Version. • Use the RSE Simulator to broadcast this TOM WSM. • Stop the CICAS application and check the Log File to verify a TOM Header Check Failure is reported indicating a reason of Framework Version error. Record the results in Table 52.

Table 51: Radio Handler / Data Demux Test Logging Parameters

Parameter	Value
GlobalLogFlag	1
CDITomHdrChkFailedLogFlag	1

Table 52: Requirement 25 Test Results

Log Entry	Expected Value	Actual Value
TOM Hdr Chk Failed	Wrong TOM Framework Version	Wrong TOM Framework Version

A.14 GPS Handler Tests

Test Procedure

Req. #	Procedure
Pretest	• Ensure the cicas-v.conf parameters are reset to the default values, and then set the parameters listed in Table 53 to the indicated values. • Boot the WSU. • Configure the CAN simulator connected to CAN1 to capture CAN messages between the WSU and DVI. Start the CAN simulator application. • Reset the CAN application simulating the Netway box to the default configuration. Start the CAN simulator application.
31	• Start the CICAS application on the WSU. • Configure GPS Simulator data that contains an invalid checksum. • Start the GPS Simulator. • Check the DAS to verify CAN message $606 indicates a GPS NMEA Checksum error was detected. Record the results in Table 54. • Stop the GPS Simulator. • Stop the CICAS application.
32	• Configure GPS Simulator data indicating no GPS solution is available. • Start the GPS Simulator and allow it to run for at least 5 seconds. • Start the CICAS application on the WSU. • Check the DAS to verify CAN message $606 indicates an Inaccurate or No GPS Solution warning. Record the results in Table 55. • Stop the CICAS application and check the Log File to verify a GPS Timeout with No Fix indication was logged. Record the results in Table 55. • Stop the GPS Simulator.
33	• Configure the GPS Simulator with valid data and as indicated in Table 56. • Start the CICAS application on the WSU. • Start the GPS Simulator and allow it to run for at least 5 seconds. • Check the DAS to verify GPS data reported in CAN messages $606, $614, and $615 is correct. Record the results in Table 57. • Stop the CICAS application and check the Log File to verify the GPS data is correctly logged. Record the results in Table 57.

Req. #	Procedure
34 126 128	• Ensure the GPS Simulator is still running from the previous test. If it is not still running, restart the GPS Simulator using the GPS data from the previous test. • Start the CICAS application on the WSU and allow it to run for at least 5 seconds. • Stop the GPS Simulator. • Check the DAS to verify CAN message $606 indicates the GPS Data is Invalid. Record the results in Table 58. • Allow CICAS to continue running for at least 60 additional seconds. • Check the DAS to verify the DVI State indicates Malfunction. Record the results in Table 59. • Stop the CICAS application and check the Log File to verify a GPS Expiration error indicating no GPS data has been logged. Record the results in Table 58. • Check the Log File to verify the OBE Health State changed from Normal to Malfunction. Record the results in Table 59.
35	• Start the GPS Simulator using the GPS data from the previous test. • Start the CICAS application on the WSU and allow it to run for at least 5 seconds. • Stop the GPS Simulator. • Configure the GPS Simulator data so that no solution is available. • Start the GPS Simulator. • Check the DAS to verify a GPS Solution Lost indication has been logged. Record the results in Table 60. • Allow the GPS Simulator to continue to run for at least 60 seconds. • Stop the CICAS application and check the Log File to verify a GPS Solution Lost error has been logged. Record the results in Table 60.
37	• Configure the RSE Simulator to transmit a GPSC TOM with the GPS Status flag indicating Unhealthy. • Start transmitting the TOM. • Ensure the CICAS application is running on the WSU and allow it to run for at least 5 seconds. • Check the DAS to verify CAN message $618 indicates the GPSC was not received. Record the results in Table 61. • Stop transmitting the TOM. • Stop the CICAS application.
38	• With the WSU powered off, connect a NovAtel GPS receiver to the WSU’s serial port as illustrated in Figure 76. • Configure the RSE Simulator to transmit a GPSC TOM with non-zero length RTCM 1005 and 1001 messages and valid checksums. • Transmit the TOM. • Boot the WSU and start the CICAS application. • After the CICAS application has run for a few seconds, verify the red light on the NovAtel GPS Receiver flashes, indicating GPS Correction data has been received. Record the results in Table 62. • Stop transmitting the TOM.

Req. #	Procedure
39 (a)	• Configure the RSE Simulator to transmit a TOM containing GPS Corrections with an invalid RTCM 1005 checksum. • Transmit the TOM. • With the CICAS application running on the WSU, verify the red light on the NovAtel GPS Receiver does not flash. Record the results in Table 63. • Stop the CICAS application and check the Log File to verify a RTCM 1005 Checksum Error has been logged. Record the results in Table 63. • Stop transmitting the TOM.
39 (b)	• Configure the RSE Simulator to transmit a TOM containing GPS Corrections with an invalid RTCM 1001 checksum. • Transmit the TOM. • Start the CICAS application on the WSU, and allow it to run for at least 5 seconds. • Verify the red light on the NovAtel GPS Receiver does not flash. Record the results in Table 64. • Stop the CICAS application and check the Log File to verify a RTCM 1001 Checksum Error has been logged. Record the results in Table 64. • Stop transmitting the TOM.

Table 53: GPS Handler Test Logging Parameters

Parameter	Value
GlobalLogFlag	1
GPSSolLostLTTime	1
CVLGpsDataTmoutLogFlag	1
CVLGpsDataTmoutNoDataLogFlag	1
CLOGLogFlag	1
CVLGpsDataRxLogFlag	1
CVLGpsSolLostLTLogFlag	1
CVLGpscRtckCksmFailLogFlag	1
InitErrType75	2

Table 54: Requirement 31 Test Results

CAN Message	Parameter	Expected Value	Actual Value
$606	GPSNMEAChkError (Byte-4, Bit-3)	1	1

Table 55: Requirement 32 Test Results

CAN Message	Parameter	Expected Value	Actual Value
$606	GPSSolError (Byte-1, Bit-7)	1	1
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
GPS TmOut (No Fix)	N/A	Logged	Logged

Table 56: GPS Data

Parameter	Value
Time	085959
Date	040708
Latitude	3308.007 N
Longitude	11913.6490 W
Heading	203.4
Altitude	156.6
Groundspeed	000.0
HDOP	0.8
PDOP	1.5
Num of Satellites	11
GST major axis error ellipse standard deviation	6.6
GST minor axis error ellipse standard deviation	4.7

Table 57: Requirement 33 Test Results

CAN Message	Parameter	Expected Value	Actual Value
$606	GPSSolError (Byte-1, Bit-7)	0	0
$614	VehLatitude (Bytes 0-3)	331334500	331334499
	VehLongitude (Bytes 4-7)	-1192274800	-1192274833
$615	VehHeading (Bytes 0-1)	20340	20340
	VehAltitude (Bytes 2-3)	156.6	156.6
$616	PDOP (Byte 2)	15	15
	HDOP (Byte 3)	8	8
	NumSat ((Byte 4)	11	11
	GST	115	0
$617	DiffAge (Bytes 0-1)	0	0
	GPSSpeed (Bytes 2-3)	0	0
	GPSSolAge (Bytes 4-5)	Increasing	Increasing
	GPSWeekNum Bytes (6-7)	462	462
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
GPS Data Rx	Time	085959	085959
	Date	040708	040708
	Latitude	33.13345 N	33.13345
	Longitude	119.22748 W	-119.227483
	Altitude	156.6	156.6
	Groundspeed	000.0	0.00
	Hdop	0.8	0.8
	Pdop	1.5	1.5
	Numsats	11	11

Table 58: Requirement 34 Test Results

CAN Message ID	Parameter	Expected Value	Actual Value
$606	NoGPS (Byte-0, Bit-0)	1	1
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
GPS Tmout (No Data)	N/A	Logged	Logged
GPS Solution Lost Long-Term	N/A	Logged	Logged

Table 59: Requirements 126, 128 Test Results

CAN Message ID	Parameter	Expected Value	Actual Value
$700	DVIState (Byte-0)	1 (Red)	1
	DVIFrequency (Byte-2)	255 (ON)	255
$619	DVINIconStates (Byte-6, Bits 0-3)	5 (Malfunction)	5
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
CLOGP OBE Health	Current state	MALFUNCTION	MALFUNCTION

Table 60: Requirement 35 Test Results

CAN Message	Parameter	Expected Value	Actual Value
$606	GPSSolLost (Byte-0, Bit-1)	1	1
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
GPS Solution Lost Long-Term	N/A	Logged	Logged

Table 61: Requirement 37 Test Results

CAN Message	Parameter	Expected Value	Actual Value
$618	GPSCReceived (Byte-1, Bit-3)	0	0

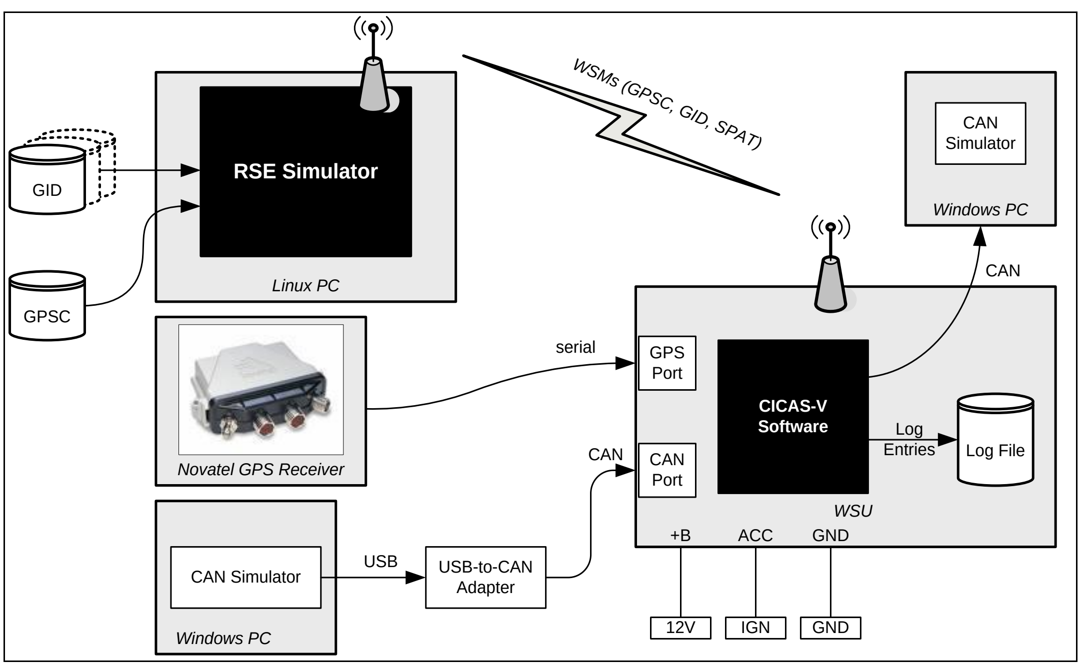

Figure 76: Test Setup

Table 62: Requirement 38 Test Results

Parameter	Expected Value	Actual Value
Red light on NovAtel GPS receiver	Flashes	Flashes

Table 63: Requirement 39 (a) Test Results

Parameter	Expected Value	Actual Value
Red light on NovAtel GPS receiver	Does Not Flash	Does not flash
Log Entry	**Expected Value**	**Actual Value**
GPSC RTCM Chksum Failure	MsgID 1005	MsgID,1005

Table 64: Requirement 39 (b) Test Results

Parameter	Expected Value	Actual Value
Red light on NovAtel GPS receiver	Does Not Flash	Does not flash
Log Entry	**Expected Value**	**Actual Value**
GPSC RTCM Chksum Failure	MsgId 1001	MsgID,1001

A.15 GID Database Handler Tests

Test Procedure

Req. #	Procedure
Pretest	• Setup the test environment as indicated in Figure 76. • Ensure the cicas-v.conf parameters are reset to the default values, and then enable logging flags for the parameters listed in Table 65 and set the parameters listed in Table 66 to the values indicated. • Ensure no GID data is stored in the WSU database. (Remove all files from /rwflash/giddb/) • Reset the CAN application simulating the Netway box to the default configuration. Start the CAN simulator. • Boot the WSU and start the CICAS application.
45	• Configure the RSE Simulator to broadcast a GID WSA on the CCH at least once per second, and the corresponding GID WSM on the SCH at least once per second. Configure the GID using the parameters listed in Table 67. • Ensure the RSE Simulator is set up to receive GPS PPS signals for channel switching synchronization. • Start the RSE Simulator and allow it to run for approximately 15 seconds. • Stop the RSE Simulator. • Stop the CICAS application and check the Log File to verify the GID data broadcast by the RSE Simulator was stored in the WSU database. Record the information in Table 68.
44	• Compare the recorded Load Times for GID Rx and New GID Rx in Table 68. • Verify the Load Time recorded for GID Rx is later than the Load Time recorded for New GID Rx by approximately 15 seconds. Record the information in Table 69.
41	• Disconnect the GPS receiver from the WSU. • Change the system time on the WSU to a date that is > 1 day later than the Load Time recorded in the GID Rx in the test above. Copy the Load Time recorded for GID Rx in Table 68 into the Expected Results column of Table 70. • Start the CICAS application on the WSU and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify the GID was removed from the WSU database. Record the results in Table 70.

Req. #	Procedure
42 43	• Setup the test environment as indicated in Figure 76. (Reconnect the GPS receiver to the WSU.) • Configure three unique GIDs to fill the space allocated to GID storage on the WSU. • Start the CICAS application on the WSU. • Use the RSE Simulator to broadcast the GIDs on the CCH. • Stop the CICAS application and check the Log File to verify the data for all GIDs has been stored in the WSU database. Record the information for each GID stored in the WSU database in Table 71. • Stop the RSE Simulator. • In cicas-v.conf, set the parameter listed in Table 72 to the value indicated. • Start the CICAS application on the WSU and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify which GID was deleted from the WSU database. Verify the GID with the oldest Load Time was deleted. Record the results in Table 73.

46	• In cicas-v.conf, set the parameters listed in Table 66 to the values indicated.
	• Ensure the CICAS application is not running and remove all GIDs from the WSU database. (Remove all files from /rwflash/giddb/)
	• Configure two RSE Simulators with GID, SPaT, and GPSC configurations as indicated in Table 74.
	    • The expected vehicle LAT/LON is shown in Table 74.
	    • Ensure the RSE Simulators are set up to receive GPS PPS signals for channel switching synchronization.
	• Use the RSE Simulator to broadcast the GID WSAs and WSMs.
	• Start the CICAS application on the WSU and allow it to run for at least 15 seconds.
	• Check the DAS parameters listed in Table 75 to verify the actual values match the expected values and record the results.
	• Stop the CICAS application and check the Log File to verify the parameters listed in Table 75 are logged in the sequence indicated.
	• Stop the RSE Simulators.

<table>
<tr><th>Req. #</th><th>Procedure</th></tr>
<tr><td>23
24
47</td><td>

Ensure the CICAS application is not running and remove all GIDs from the WSU database. (Remove all files from /rwflash/giddb/)
Configure one RSE Simulator to broadcast a GID WSA on the Control Channel and the corresponding GID WSM on Service Channel 180. Use the GID data listed in Table 76.
Configure this RSE Simulator so that it does not broadcast GPSC or SPaT WSMs.
Configure a second RSE Simulator to broadcast a GPSC WSA on the Control Channel and the corresponding GPSC WSM on Service Channel 176.
Configure this RSE Simulator so that it does not broadcast GID or SPaT WSMs.
Ensure both RSE Simulators are set up to receive GPS PPS signals for channel switching synchronization.
Start the RSE Simulators.
Start the CICAS application on the WSU and allow it to run for at least 15 seconds.
Stop the CICAS application and check the Log File to verify the GID Rx is logged before the GPSC Rx Hdr is logged. Record the results in Table 77.
Stop the RSE Simulators.
</td></tr>
<tr><td>49</td><td>

Ensure the CICAS application is not running and remove all GIDs from the WSU database. (Remove all files from /rwflash/giddb/)
Configure the RSE Simulator to broadcast a WSA on the CCH advertising an Area GID containing two (2) GIDs, and to broadcast the two corresponding GIDs on the SCH. Use the GID information indicated in Table 78.
Ensure the WSA does not advertise GPSC or SPaT on the SCH.
Ensure the RSE Simulator is set up to receive GPS PPS signals for channel switching synchronization.
Start the CICAS application on the WSU.
Start the RSE Simulator and allow it to run for approximately 15 seconds.
Stop the CICAS application and check the Log File to verify the GID data broadcast by the RSE Simulator was stored in the WSU database. Record the information in Table 79 under the “First Pass Results”.
Restart the CICAS application on the WSU (with the RSE Simulator still running) and allow if to run for at least 15 seconds.
Stop the CICAS application and check the Log File to verify the GID data broadcast by the RSE Simulator was not updated in the WSU database. Record the information in Table 79 under the “Second Pass Results”.
</td></tr>
</table>

Req. #	Procedure
50	• In cicas-v.conf, reset the parameters to the default values. Then enable the log flags indicated in Table 65 and set the parameter listed in Table 80 to the value listed. • Configure two unique GIDs to fill the allocated space. • Start the CICAS application on the WSU. • Use the RSE Simulator to broadcast the GID WSMs. • Stop the CICAS application and check the Log File to verify the data for both GIDs has been stored in the WSU database. Record the information for each GID stored in the WSU database in Table 81. • Stop the RSE Simulator. • Create a third unique GID and use the RSE Simulator to broadcast only this GID. • Start the CICAS application on the WSU and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify the GID with the oldest Load Time was deleted to make room for the new GID. Verify the new GID was recorded in the WSU database. Record the results in Table 82. • Stop the RSE Simulator. • Continue to the next test below and record the amount of time that passes between the tests in Table 83.
51	• Allow at least 10 seconds to pass after running the above test. • Use the RSE Simulator to broadcast a TOM on the CCH containing the most recent GID from the previous test. • Start the CICAS application on the WSU and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify the Load Time recorded for this GID is later than the Load Time recorded in Table 82 by the amount of time that passed between the previous test and this test. Record the information in Table 84.
52	• Change the Content Version in the GID TOM used in the above test. • Use the RSE Simulator to broadcast this updated GID TOM. • Start the CICAS application on the WSU and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify the updated GID TOM was processed by the WSU. Record the results in Table 85.

Table 65: GID Database Handler Test Logging Parameters

Parameter	Value
GlobalLogFlag	1
CDIGidRxLogFlag	1
CDINewGidRxLogFlag	1
CDIExpGIDRecDelLogFlag	1
CDIUnexpGIDDelLogFlag	1
CVLGpscRxLogFlag	1
CDIWbssJoinedLogFlag	1
CDIWsaRxLogFlag	1
CDINewModWSARxLogFlag	1

Table 66: GID Database Test Setup

Parameter	Value
GIDExpirationTime	1 day
GIDStorageAllocation	90 KB

Table 67: Requirement 45 Test Setup

Parameter	Value
Intersection ID	2
Format Version	1
Content Version	1

Table 68: Requirement 45 Test Results

Log Entry	Parameter	Expected Value	Actual Value
GID Rx	Intersection ID	2	2
	Format Version	1	1
	Content Version	1	1
	Load Time	N/A	1217975956
New GID Rx	Intersection ID	2	2
	Format Version	1	1
	Content Version	1	1
	Load Time	N/A	1217975941

Table 69: Requirement 44 Test Results

Log Entry	Parameter	Expected Value	Actual Value
GID Rx	Load Time	1217975956	1217975956
New GID Rx	Load Time	< GID Rx Load Time	1217975941

Table 70: Requirement 41 Test Results

Log Entry	Parameter	Expected Value	Actual Value
Exp GID Rec Del	Intersection ID	2	2
	Load Time	1217975941	1217975941

Table 71: Requirement 42 Test Setup

Log Entry	Parameter	Value
New GID Rx	Intersection ID	1
	Format Version	1
	Content Version	1
	Load Time	1217977596
New GID Rx	Intersection ID	2
	Format Version	1
	Content Version	1
	Load Time	1217977596
New GID Rx	Intersection ID	3
	Format Version	1
	Content Version	1
	Load Time	1217977596

Table 72: Requirement 42 Test Setup

Parameter	Value
GIDStorageAllocation	60 KB

Table 73: Requirement 42, 43 Test Results

Log Entry	Parameter	Value
Unexp GID Rec Del	Intersection ID	1
	Format Version	1
	Content Version	1
	Load Time	1217977596

Table 74: Requirement 46 Test Setup

RSE-1 Configuration	Value
GID & SPaT Intersection ID	2
GID Reference Point Latitude	33.1333
GID Reference Point Longitude	-117.2285
GID WSM Service Channel	176
SPaT WSM Channel	CCH
GPSC WSM Channel	CCH
RSE-2 Configuration	**Value**
GID & SPaT Intersection ID	4
GID Reference Point Latitude	33.1333
GID Reference Point Longitude	-117.2318
GID WSM Service Channel	180
SPaT WSM Channel	CCH
GPSC WSM Channel	CCH
Vehicle GPS Data	**Value**
Expected Vehicle Latitude	33.1333
Expected Vehicle Longitude	-117.2275

Table 75: Requirement 46 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$618	WSAReceived (Byte-1, Bit-6)	1	1
	SPATReceived (Byte-1, Bit-5)	1	1
	GIDReceived (Byte-1, Bit-4)	1	1
Log Entry	**Parameter**	**Expected Sequence**	**Actual Sequence**
WBSS Joined	SCH 176	1	2
GID Received	SCH 176	2	3
WBSS Joined	SCH 180	3	1
GID Received	SCH 180	4	Not Logged

Table 76: Requirement 23, 24, 47 Test Setup

RSE-1 Configuration	Value
GID Intersection ID	2
GID Reference Point Latitude	33.1333
GID Reference Point Longitude	-117.2285
GID WSM Service Channel	180
RSE-2 Configuration	**Value**
GPSC WSM Service Channel	176
Vehicle GPS Data	**Value**
Expected Vehicle Latitude	33.1333
Expected Vehicle Longitude	-117.2275

Table 77: Requirement 23, 24, 47 Test Results

Log Entry	Parameter	Expected Value	Actual Value
New GID Rx	Intersection ID	2	2
	Format Version	1	1
	Content Version	1	1
	Load Time	N/A	1218127803
GPSC Rx Hdr	Logged	Logged after New GID Rx	yes

Table 78: Requirement 49 Test Setup

GID-1 Parameter	Value
Intersection ID	1
Format Version	1
Content Version	1
GID-2 Parameter	**Value**
Intersection ID	2
Format Version	1
Content Version	1

Table 79: Requirement 49 Test Results

First Pass Results			
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
New GID Rx (IID 1)	Intersection ID	1	1
	Format Version	1	1
	Content Version	1	1
	Load Time	N/A	1218045253
New GID Rx (IID 2)	Intersection ID	2	2
	Format Version	1	1
	Content Version	1	1
	Load Time	N/A	1218045253
Second Pass Results			
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
New GID Rx (IID 1)	Intersection ID	Not logged	Not Logged
	Format Version	Not logged	Not Logged
	Content Version	Not logged	Not Logged
	Load Time	Not logged	Not Logged
New GID Rx (IID 2)	Intersection ID	Not logged	Not Logged

First Pass Results			
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
	Format Version	Not logged	Not Logged
	Content Version	Not logged	Not Logged
	Load Time	Not logged	Not Logged

Table 80: Requirement 50 Test Setup

Parameter	Value
GIDStorageAllocation	60 KB

Table 81: Requirement 50 Test Setup

GID	Parameter	Value
	Intersection ID	1
	Format Version	1
	Content Version	1
	Load Time	1219361401
	Intersection ID	2
	Format Version	1
	Content Version	1
	Load Time	1219361401

Table 82: Requirement 50 Test Results

Log Entry	Parameter	Value
Unexp GID Rec Del	Intersection ID	1
	Format Version	1
	Content Version	1
	Load Time	1219361401
New GID Rx	Intersection ID	3
	Format Version	1
	Content Version	1
	Load Time	1219361549

Table 83: Requirement 50-51 Time Between Tests

Parameter	Value (sec.)
Time Between Tests	~100s

Table 84: Requirement 51 Test Results

GID	Parameter	Expected Value	Actual Value
From data recorded for New GID Rx in Table 82.	Intersection ID	N/A	3
	Format Version	N/A	1
	Content Version	N/A	1
	Load Time	N/A	1219361549
Values recorded for Requirement 51 Test.	Intersection ID	Same as above	3
	Format Version	Same as above	1
	Content Version	Same as above	1
	Load Time	~= Value in Table 83	1219361688

Table 85: Requirement 52 Test Results

GID	Parameter	Expected Value	Actual Value
Values recorded in Table 84.	Intersection ID	Same as above	3
	Format Version	Same as above	1
	Content Version	New Content Version	1
	Load Time	New Load Time	1219361688
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
New GID Rx	Intersection ID	Same as above	3
	Format Version	Same as above	1
	Content Version	2	2
	Load Time	N/A	1219361912

A.16 SPaT Handler Tests

Test Procedure

Req. #	Procedure
Pretest	• Setup the test environment as indicated in Figure 75. • In cicas-v.conf, enable logging flags for the parameters listed in Table 86. • Reset the CAN application simulating the Netway box to the default configuration. Start the CAN simulator.

24
54
	• In cicas-v.conf, set the parameters in Table 87 to the values indicated.
	• Configure the RSE Simulator to broadcast a GID TOM, GPSC TOM, and SPaT TOM on the CCH, and all corresponding to the same intersection ID, (IID 1).
	  ◦ Configure the GID TOM to require lane-level accuracy.
	• Configure GPS Simulation data to simulate approaching the intersection described by the above GID.
	  ◦ Ensure the GGA sentence indicates Fix Quality = 5, (Float RTK).
	• Start broadcasting the GID, GPSC, and SPaT TOMs using the RSE Simulator.
	• Reboot the WSU and start the CICAS application.
	• Send the GPS data to the WSU using the GPS Simulator.
	• Check the DAS to verify the parameters in listed in Table 88 are logged. Record the results in the table.
	• Stop the CICAS application and check the Log File to verify the SPaT data is logged. Record the results in Table 88.

58	• Start the CICAS application on the WSU and allow it to run for at least 5 seconds. • Stop the RSE Simulator, but allow the CICAS application to continue to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify the SPaT data is flagged as expired. Record the results in Table 89. • Stop the GPS Simulator.

55	• Configure the SPaT TOM from the above test to include a SPaT CURRENTTIME Object (Object ID 9).
	  ◦ Ensure the SPaT TOM Format Version is set to 2.
	  ◦ Ensure the Date & Time information in the GPS Simulation data is later than the Date & Time information in the CURRENTTIME Object by more than the SPATTimeout threshold indicated in Table 90.
	• Reuse the GID and GPSC TOMs from the above tests.
	• Start broadcasting the GID, GPSC, and SPaT TOMs using the RSE Simulator.
	• Start the CICAS application on the WSU.
	• Use the GPS Simulator to send the GPS data used in the above test to the WSU.
	• Stop the CICAS application and check the Log File to verify the SPaT Age Greater than Threshold error has been logged. Record the results in Table 91.
	• Stop the RSE and GPS Simulators.

Req. #	Procedure
56	• Configure the RSE Simulator to broadcast a SPaT TOM that does not correspond to the Intersection ID matching the GID and GPS data in the above tests. (e.g., SPaT IID=2.) • Ensure the SPaT TOM does not include a SPaT CURRENTTIME Object. • Reuse the GID and GPSC TOMs from the above tests, (IID=1). • Start broadcasting the GID, GPSC, and SPaT TOMs using the RSE Simulator. • Start the CICAS application on the WSU. • Use the GPS Simulator to send the GPS data used in the above test to the WSU. • Stop the CICAS application and check the Log File to verify the SPaT is received and is recognized as not for the approaching intersection. Record the results in Table 92. • Stop the RSE and GPS Simulators.

Table 86: SPaT Handler Test Logging Parameters

Parameter	Value
GlobalLogFlag	1
CDISpatRxApprngLogFlag	1
CDISpatValExpLogFlag	1
SPATDataExpdLogFlag	1
SPATIntn-In-RngTmOutLogFlag	1
CDISpatRxLogFlag	1

Table 87: Requirement 54, 58 Test Setup

Parameter	Value
IntersectionIdentificationMethod	1
SPATExpirationTime	400ms

Table 88: Requirement 54 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$618	RoadLane (Byte-1, Bit-7)	1	1
$619	SPaTDataValid (Byte-6, Bit-7)	1	1
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
SPaT Rx Apprng	Intn	255	255

Table 89: Requirement 58 Test Results

Log Entry	Parameter	Expected Value	Actual Value
SPaT Val Exp	Intn	1	1

Table 90: Requirement 55 Test Setup

Parameter	Value
SPATTimeout	400ms

Table 91: Requirement 55 Test Results

Log Entry	Parameter	Expected Value	Actual Value
SPaT Val Exp	Intn	Logged	255
SPaT Data Expd	N/A	Logged	Logged

Table 92: Requirement 56 Test Results

Log Entry	Parameter	Expected Value	Actual Value
SPaT Rx	Intn	2	2
SPaT Rx Apprng	Intn	Not Logged	Not Logged

A.17 DAS Handler / Logger Tests

Test Procedure

Req. #	Procedure
Pretest	• Setup the test environment as indicated in Figure 75. • In cicas-v.conf, enable logging flags for the parameters listed in Table 93. • Reset the CAN application simulating the Netway box to the default configuration. Start the CAN simulator. • Configure GPS Sequence data for the GPS Simulator that will restart once the end of the sequence data is reached. • Configure the RSE Simulator to broadcast GID, GPSC, and SPaT TOMs periodically. • Start broadcasting the GID, GPSC, and SPaT TOMs using the RSE Simulator. • Start sending GPS data using the GPS Simulator.
60	• Reboot the WSU and start the CICAS application. • Check the DAS to verify the Heartbeat Sequence Counter in the first CAN message $606 received was set to 1. Record the results in Table 94.
61	• Check the DAS to verify the Heartbeat Sequence Counter in the 2nd through 5th messages received were each incremented by 1. Record the results in Table 94.
62 69	• Check the DAS to verify DAS messages $600-$606, $610-$619, and $650 are being sent by the WSU every 100ms. Record the results in Table 95.
63	• Configure the CAN application simulating the DAS to send CAN message $701 with a DAS System Error. • Stop the CICAS application and check the Log File to verify a DAS System Error is logged. Record the results in Table 96.
64	• Configure the CAN application simulating the DAS to send CAN message $701 with a DAS Boot Up Error. Start the CAN simulator. • Start the CICAS application on the WSU and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify a DAS Boot Up Error is logged. Record the results in Table 97.

Req. #	Procedure
65	• Configure the CAN application simulating the DAS to send CAN message $701 with a DAS Shutdown Error. • Start the CICAS application on the WSU and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify a DAS Shutdown Error is logged. Record the results in Table 98.
66	• Configure the CAN application simulating the DAS to send CAN message $701 with an OBE to DAS Heartbeat Error. • Start the CICAS application on the WSU and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify an OBE to DAS Heartbeat Error is logged. Record the results in Table 99.
67	• Configure the CAN application simulating the DAS to send CAN message $701 with a DAS Heartbeat Sequence value that is different than the OBE to DAS Heartbeat Sequence sent in the previous CAN message $606. • Start the CICAS application on the WSU and allow it to run for at least 5 seconds. • Stop the CICAS application and check the Log File to verify a DAS Heartbeat Sequence Error is logged. Record the results in Table 100.

68	• Configure the RSE Simulator to broadcast a GID TOM on the CCH with a Metric Object containing the data in Table 101.
	    • Ensure the SPaT and GPSC TOMs are configured to not include a Metric Object.
	• Broadcast the GID TOM using the RSE Simulator.
	• Set the date and time on the WSU to match the Date & Time information in the GPS Simulation data matching the data in Table 101.
	    • Start the CICAS application on the WSU immediately after setting the date and time.
	    • Allow CICAS to run for 5 seconds then stop the CICAS application on the WSU.
	• Verify the elapsed time is correctly calculated and output to the DAS. Record the results in Table 102.

	• Stop the RSE Simulator and configure it to broadcast a SPaT TOM on the CCH with a Metric Object containing the data in Table 101.
	    • Ensure the GID and GPSC TOMs are configured to not include a Metric Object.
	• Broadcast the SPaT TOM using the RSE Simulator.
	• Set the date and time on the WSU to match the Date & Time information in the GPS Simulation data matching the data in Table 101.
	    • Start the CICAS application on the WSU immediately after setting the date and time.
	    • Allow CICAS to run for 5 seconds then stop the CICAS application on the WSU.
	• Verify the elapsed time is correctly calculated and output to the DAS. Record the results in Table 102.

Req. #	Procedure

	• Stop the RSE Simulator and configure it to broadcast a GPSC TOM on the CCH with a Metric Object containing the data in Table 101.
	  ◦ Ensure the GID and SPaT TOMs are configured to not include a Metric Object.
	• Broadcast the GPSC TOM using the RSE Simulator.
	• Set the date and time on the WSU to match the Date & Time information in the GPS Simulation data matching the data in Table 101.
	  ◦ Start the CICAS application on the WSU immediately after setting the date and time.
	  ◦ Allow CICAS to run for 5 seconds then stop the CICAS application on the WSU.
	• Verify the elapsed time is correctly calculated and output to the DAS. Record the results in Table 102.

Table 93: DAS Handler / Logger Test Logging Parameters

Parameter	Value
GlobalLogFlag	1
DASSysErrLogFlag	1
DASBootupErrLogFlag	1
DASShutdownErrLogFlag	1
DASOBEDASHbErrLogFlag	1
DASHbMismatchLogFlag	1
InitErrType56	1
InitErrType57	1
InitErrType58	1
InitErrType59	1
InitErrType66	1

Table 94: Requirements 60 & 61 Test Results

Parameter	Sent CAN message $606	Expected Results	Actual Results
Heartbeat Sequence Counter value	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	5

Table 95: Requirement 62 & 69 – 10Hz Test Results

CAN Messages	Time Between Messages	
	Expected	Actual
$600	100ms	99-101ms
$601	100ms	99-101ms
$602	100ms	99-101ms
$603	100ms	99-101ms
$604	100ms	99-101ms
$605	100ms	99-101ms
$606	100ms	99-101ms
$610	100ms	99-101ms
$611	100ms	99-101ms
$612	100ms	99-101ms
$613	100ms	99-101ms
$614	100ms	99-101ms
$615	100ms	99-101ms
$616	100ms	99-101ms
$617	100ms	99-101ms
$618	100ms	99-101ms
$619	100ms	99-101ms
$650	100ms	99-101ms

Table 96: Requirement 63 Test Results

Log Entry	Parameter	Expected Value	Actual Value
DAS Error Indication	DASSystemError	1	1

Table 97: Requirement 64 Test Results

Log Entry	Parameter	Expected Value	Actual Value
DAS Error Indication	DASBootupError	1	1

Table 98: Requirement 65 Test Results

Log Entry	Parameter	Expected Value	Actual Value
DAS Error Indication	DASShutdownError	1	1

Table 99: Requirement 66 Test Results

Log Entry	Parameter	Expected Value	Actual Value
DAS Error Indication	OBEDASHeartbeatError	1	1

Table 100: Requirement 67 Test Results

Log Entry	Parameter	Expected Value	Actual Value
DAS Error Indication	DASHbSeqMismatch	1	1

Table 101: Metric Object Data

Parameter	Value
Object ID	255
Object Size	11
Year	2008
Month	3
Day	15
Hour	8
Minute	59
Milliseconds	59
Message Counter	11

Table 102: Requirement 68 Test Results

GID Metric Object Test Results			
CAN Message	**Parameter**	**Expected Value**	**Actual Value**
$751	Elapsed Years	0	0
	Elapsed Months	0	0
	Elapsed Days	0	0
	Elapsed Hours	0	0
	Elapsed Minutes	0	0
	Elapsed Seconds	5	0
	Elapsed Milliseconds	N/A	N/A
	Layer Type	1	1
	Message Counter	11	11
SPaT Metric Object Test Results			
CAN Message	**Parameter**	**Expected Value**	**Actual Value**
$751	Elapsed Years	0	0
	Elapsed Months	0	0
	Elapsed Days	0	0
	Elapsed Hours	0	0
	Elapsed Minutes	0	0
	Elapsed Seconds	5	0
	Elapsed Milliseconds	N/A	N/A
	Layer Type	2	2
	Message Counter	11	11
GPSC Metric Object Test Results			
CAN Message	**Parameter**	**Expected Value**	**Actual Value**
$751	Elapsed Years	0	0
	Elapsed Months	0	0
	Elapsed Days	0	0
	Elapsed Hours	0	0
	Elapsed Minutes	0	0
	Elapsed Seconds	5	0
	Elapsed Milliseconds	N/A	N/A
	Layer Type	3	3
	Message Counter	11	11

A.18 Violation Detection Module Tests

Intersection Identification Tests

A.18.1.1 Test Procedure

Req. #	Procedure
Pretest	• Setup the test environment as indicated in Figure 75. • In cicas-v.conf, reset the parameters to the default values. Then enable logging flags for the parameters listed in Table 103.
72	• Configure GPS Sequence data for the GPS Simulator with valid GPS data and HDOP = 1.0. • Boot the WSU and start the CICAS application. • Use the GPS Simulator to broadcast the GPS data for at least 5 seconds. • Check the DAS to verify HDOP Over-Limit indication is not indicated. Record the results in Table 104. • Stop the GPS Simulator and modify the above GPS Sequence data so that HDOP = 4.0 and remains at that level for a period of time longer than three seconds. • Use the GPS Simulator to broadcast the GPS data. Allow the GPS Simulator to run long enough so that the entire GPS data file has been transmitted. • Check the DAS parameters listed in Table 105 and record the results in the table. • Stop the CICAS application and check the Log File to verify the HDOP Status Change is logged. Record the results in Table 105.
73	• Configure the CAN simulator connected to CAN1 to simulate the Netway box using the default configuration. Start the CAN simulator. • Configure GPS Sequence data for the GPS Simulator with invalid Latitude information. • Start the CICAS application on the WSU. • Start sending GPS data using the GPS Simulator and allow it to continue for at least 5 seconds. • Check the DAS to verify an invalid GPS Position is indicated. Record the results in Table 106. • Stop the CICAS application and check the Log File to verify Map Match Unsucc is logged and record the results in Table 106. • Stop the GPS Simulator.
	• Configure GPS Sequence data for the GPS Simulator with invalid Longitude information. • Start sending GPS data using the GPS Simulator and allow it to continue for at least 5 seconds. • Check the DAS to verify an invalid GPS Position is indicated. Record the results in Table 107. • Stop the CICAS application and check the Log File to verify Map Match Unsucc is logged and record the results in Table 107.

Req. #	Procedure
74	Setup for all Requirement 74 test variations: • In cicas-v.conf, set the parameter listed in Table 108 to the value indicated. • Configure two RSE Simulators, (RSE-1 & RSE-2), each transmitting a different GID TOM, (GID-1 & GID-2 respectively). Configure the PSC Reference Points Latitude and Longitude values for the GIDs to provide approximately 200m spacing between the intersection reference points and such that the intersections lie along the same road, as indicated in Figure 77. Record the GID Info in Table 109. • Configure the RSE-1 Simulator to broadcast a GPSC TOM. • Configure GPS Simulation data to start with the vehicle positioned on both GID-1 and GID-2 and approaching both intersections, but closer to GID-2. Additionally, configure the GPS Simulation data so that the vehicle passes through the intersection corresponding to GID-2, and approaches the intersection corresponding to GID-1 but remains closer to GID-2, as indicated in Figure 77. • Reset the CAN application simulating the Netway box to the default configuration.
	74a) If low speed filtering is disabled, identify the approaching intersection based on distance to the intersection. The parameters for this test are set so the algorithm should select only GID-2 based on distance to the intersection. The GPS Simulation data should run until the vehicle has passed the GID-2 stop bar, but has not yet passed the GID-2 intersection reference point. • In cicas-v.conf, set the EnableLowVehicleSpeedFiltering parameter to 0 (low vehicle speed filtering disabled.) • In the CAN application simulating the Netway box, set the CAN vehicle speed to 2.0, (less than the Low Vehicle Speed Threshold indicated in Table 108.) Start the CAN simulator. • Configure the GPS Simulation data so the vehicle speed is 2.0, (less than the Low Vehicle Speed Threshold indicated in Table 108.) • Ensure the GPS Simulation data is sufficient to allow the GPS Simulator to run for least 15 seconds before reaching the end of the data. • Start broadcasting the GID and GPSC TOMs using the RSE Simulators. • Use the GPS Simulator to broadcast the GPS data. • Start the CICAS application on the WSU and allow it to run for 10 seconds. • Check the DAS parameters listed in Table 110 and record the results in the table. • Stop the CICAS application and check the Log File to verify the WSU identifies the intersection corresponding to GID-2 as the approaching intersection. Record the results in Table 110. • Stop the GPS Simulator. • Stop the CAN application simulating the Netway box.
	74b) If low speed filtering is enabled & vehicle speed is equal to or greater than the Low Vehicle Speed Threshold, identify the approaching intersection based on direction of travel. The parameters for these tests are set so the algorithm should select GID-1 at the conclusion of each test. The GPS Simulation data should run until the vehicle has passed the GID-2 intersection reference point, and should be stopped before the data reaches the end of the file. • In cicas-v.conf, set the EnableLowVehicleSpeedFiltering parameter to 1 (low vehicle speed filtering enabled.) • Run tests i – iii below. Record the results in Table 111.

<table>
<tr><th>Req. #</th><th>Procedure</th></tr>
<tr><td></td><td>i) Test using CAN vehicle speed and GPS heading:
In cicas-v.conf, set the IntersectionIdentificationMethod parameter to 1 (CAN vehicle speed and GPS heading.)
In the CAN application simulating the Netway box, configure the CAN vehicle speed to be 5.0, (greater than the Low Vehicle Speed Threshold indicated in Table 108.) Start the CAN simulator.
Reuse the GPS Simulation data from the previous test, (i.e., 74a.)
Start broadcasting the GID and GPSC TOMs using the RSE Simulators.
Use the GPS Simulator to broadcast the GPS data.
Start the CICAS application on the WSU.
Before the GPS Simulator reaches the end of its data, check the DAS to verify the WSU identifies CAN vehicle speed and the intersection corresponding to GID-1 as the most likely approaching intersection. Record the results in Table 111.
Before the GPS Simulator reaches the end of its data, stop the CICAS application and check the Log File to verify the WSU identifies the intersection corresponding to GID-1 as the most likely approaching intersection. Record the results in Table 111.
Stop the GPS Simulator.
Stop the CAN application simulating the Netway box.</td></tr>
<tr><td></td><td>ii) Test using GPS location data:
In cicas-v.conf, set the IntersectionIdentificationMethod parameter to 2 (GPS location data.)
Configure the GPS Simulation data so the vehicle speed is 5.0, (greater than the Low Vehicle Speed Threshold indicated in Table 108.)
Ensure the GPS Simulation data is sufficient to allow the GPS Simulator to run for least 15 seconds before reaching the end of the data.
Use the same CAN simulator settings from the previous test. Start the CAN simulator.
Start broadcasting the GID and GPSC TOMs using the RSE Simulators.
Use the GPS Simulator to broadcast the GPS data.
Start the CICAS application on the WSU.
Before the GPS Simulator reaches the end of its data, check the DAS to verify the WSU identifies CAN vehicle speed and the intersection corresponding to GID-1 as the most likely approaching intersection. Record the results in Table 112.
Before the GPS Simulator reaches the end of its data, stop the CICAS application and check the Log File to verify the WSU identifies the intersection corresponding to GID-1 as the most likely approaching intersection. Record the results in Table 112.
Stop the GPS Simulator.
Stop the CAN application simulating the Netway box.</td></tr>
</table>

Req. #	Procedure
	iii) Test using GPS location with Filter: • In the Configuration Parameter File, set the IntersectionIdentificationMethod parameter to 3 (GPS location with Filter.) • Reuse the GPS Simulation data from the previous test, (i.e., part ii.) • Use the same CAN simulator settings from the previous test. Start the CAN simulator. • Start broadcasting the GID and GPSC TOMs using the RSE Simulators. • Use the GPS Simulator to broadcast the GPS data. • Start the CICAS application on the WSU. • Before the GPS Simulator reaches the end of its data, check the DAS to verify the WSU identifies CAN vehicle speed and the intersection corresponding to GID-1 as the most likely approaching intersection. Record the results in Table 113. • Before the GPS Simulator reaches the end of its data, stop the CICAS application and check the Log File to verify the WSU identifies the intersection corresponding to GID-1 as the most likely approaching intersection. Record the results in Table 113. • Stop the GPS Simulator. • Stop the CAN application simulating the Netway box.
	74c) If low speed filtering is enabled, identify the approaching intersection based on the intersection closing most quickly. The parameters for this test are set so the algorithm should select both GID-2 and GID-1 since the vehicle is closing on both intersections at exactly the same rate. The GPS Simulation data should be stopped after the vehicle has passed the GID-2 stop bar, but has not yet passed the GID-2 intersection reference point. • In cicas-v.conf, set the EnableLowVehicleSpeedFiltering parameter to 1, IntersectionIdentificationMethod parameter to 2, and IntersectionSelectionCriteria parameter to 0. • In the CAN application simulating the Netway box, set the CAN vehicle speed to 5.0. Start the CAN simulator. • Configure the GPS Simulation data so the vehicle speed is 5.0. • Ensure the GPS Simulation data is sufficient to allow the GPS Simulator to run for least 15 seconds before reaching the end of the data. • Boot the WSU. • Start broadcasting the GID and GPSC TOMs using the RSE Simulators. • Use the GPS Simulator to broadcast the GPS data. • Start the CICAS application on the WSU and allow to run for 10 seconds, and then stop the CICAS application on the WSU. • Check the DAS parameters listed in Table 114 and record the results in the table. • Stop the CICAS application and check the Log File to verify the WSU identifies both intersections corresponding to GID-1 and GID-2 as the approaching intersections. Record the results in Table 114. • Stop the GPS Simulator. • Stop the CAN application simulating the Netway box.

Req. #	Procedure
75	• In cicas-v.conf, set the EnableLowVehicleSpeedFiltering parameter to 1 (low vehicle speed filtering enabled), IntersectionIdentificationMethod parameter to 1 (CAN vehicle speed and GPS heading), and set the IntersectionSelectionCriteria parameter to 1. • Reuse the GPS Simulation data from the previous test, (i.e., part ii.) • Use the same CAN simulator settings from the previous test. Start the CAN simulator. • Start the CICAS application on the WSU. • Use the GPS Simulator to broadcast the GPS data. • Allow the GPS Simulator to run for 5 seconds then stop the CICAS application on the WSU. • Check the Log File to verify the WSU identifies the intersection corresponding to GID-2 as the most likely approaching intersection. Record the results in Table 115. • Stop the GPS Simulator. • Stop the RSE Simulator.
77	• In cicas-v.conf, set the IntersectionIdentificationMethod parameter to 2 (GPS location data.) • Ensure no GID data is stored in the WSU database. (Remove all files from /rwflash/giddb/) • Reuse the GPSC data from the above tests. • Configure GID-1 and GID-2 from the above tests to require lane-level accuracy. • Configure the accuracy mode for the GPS simulation data from the previous test to 9 (WAAS). • Use the same CAN simulator settings from the previous test. Start the CAN simulator. • Start the CICAS application on the WSU. • Start broadcasting the GID and GPSC TOMs using the RSE Simulator. • Use the GPS Simulator to broadcast the GPS data. • Use the CAN simulator to record a Trace of the CAN data output to the DAS. • Allow the GPS Simulator to run for 5 seconds then stop the CAN simulator Trace and verify the WSU does not identify any intersection when the GPS Accuracy is insufficient. Record the results in Table 116. • Stop the GPS Simulator. • Stop the CICAS application on the WSU. • Keep the RSE Simulator running for the next test.

Req. #	Procedure
76	• Reuse cicas-v.conf from the above test. • Reuse the GPS Simulation data from the previous test, (i.e., 77.) • Configure the CAN vehicle speed to be 2.0, (less than the Low Vehicle Speed Threshold indicated in Table 108.) Start the CAN simulator. • Start the CICAS application on the WSU. • Send the GPS data to the WSU using the GPS Simulator. • Allow the GPS Simulator to run for 5 seconds then stop the CICAS application on the WSU. • Check the Log File to verify the WSU does not identify any intersection as the most likely approaching intersection. Record the results in Table 117. • Stop the GPS Simulator. • Stop the RSE Simulator.

Table 103: Intersection Identification Test Logging Parameters

Parameter	Value
GlobalLogFlag	1
HDOPStatusChngLogFlag	1
NewApprngIntnLogFlag	1
CVLGpsDataRxLogFlag	1

Table 104: Requirement 72 Test Results

CAN Message	Parameter	Expected Value	Actual Value
$619	HDOPUnderLimitDuringDuration (Byte-7, Bit-6)	1	1
	IntersectionInRange (Byte-7, Bit-4)	0	0
	IntersectionClosing Byte-7, Bit-3)	0	0

Table 105: Requirement 72 Test Results

CAN Message	Parameter	Expected Value	Actual Value
$619	HDOPUnderLimitDuringDuration (Byte-7, Bit-6)	0	0
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
HDOP Status Chng	NewVal	4.0	4.0

Table 106: Requirement 73 Test Results

CAN Message	Parameter	Expected Value	Actual Value
$619	GPSPosValid (Byte-7, Bit-5)	0	0

Table 107: Requirement 73 Test Results

CAN Message	Parameter	Expected Value	Actual Value
$619	GPSPosValid (Byte-7, Bit-5)	0	0

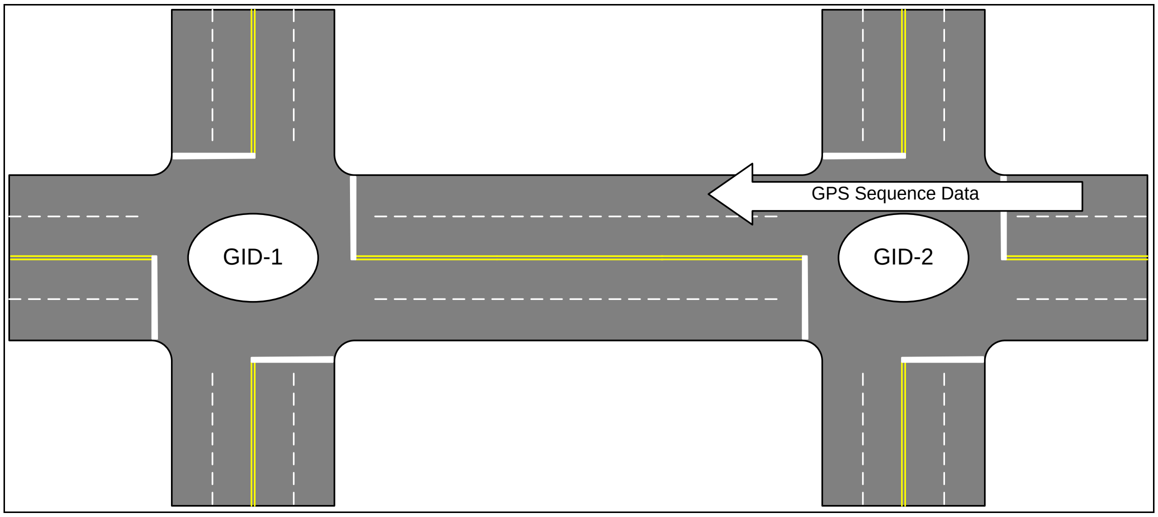

Figure 77: Test 74 GID & GPS Sequence Setup

Table 108: Requirement 74 Configuration Setup

Parameter	Value
LowVehicleSpeedThreshold	3.0

Table 109: Requirement 74 GID Setup

GID Info	Parameter	Value
GID-1	Intersection ID	1
	Intersection Type	0
GID-2	Intersection ID	2
	Intersection Type	0

Table 110: Requirement 74a Test Results

CAN Message	Parameter	Expected Value	Actual Value
$601	VehSpd (Bytes 3-4)	2.0	2.0
$619	IntersectionInrange (Byte-7, Bit-4)	1	1
	InteresectionClosing (Byte-7, Bit-3)	1	1
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
GPS Data Rx	Course	Logged	270
New Apprng Intn	Intersection ID	GID-2 Int. ID	2
	Intersection Type	GID-2 Int. Type	0
	RtofChngMovingAvg	0.556m/s	0.556
ROC method results	GPS	Logged	Logged
	CANSpeedHeading	Logged	Logged
	GPSFilter	Logged	Logged

Table 111: Requirement 74b (i) Test Results

Test (i) – CAN Vehicle Speed w/ GPS Heading			
DAS Entry	**Parameter**	**Expected Value**	**Actual Value**
$601	VehSpd (Bytes 3-4)	5.0	5.0
$610	GIDVersion	GID-1 Version	1
	IntersectionId	GID-1 Int. ID	1
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
GPS Data Rx	Course	Logged	270
New Apprng Intn	Intersection ID	GID-1 Int. ID	1
	Intersection Type	GID-1 Int. Type	0
	RtofChngMovingAvg	-1.389m/s	-5.000 [kmph] = -1.389 m/s
ROC method results	GPS	Logged	-1.389m/s
	CANSpeedHeading	Logged	-5.000 [kmph] = -1.389 m/s
	GPSFilter	Logged	-2.856m/s

Table 112: Requirement 74b (ii) Test Results

Test (ii) – GPS Location Data			
DAS Entry	**Parameter**	**Expected Value**	**Actual Value**
$601	VehSpd (Bytes 3-4)	5.0	5.0
$610	GIDVersion	GID-1 Version	1
	IntersectionId	GID-1 Int. ID	1
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
GPS Data Rx	Course	Logged	270
New Apprng Intn	Intersection ID	GID-1 Int. ID	1
	Intersection Type	GID-1 Int. Type	0
	RtofChngMovingAvg	-1.389m/s	-1.389
ROC method results	GPS	Logged	-1.389m/s
	CANSpeedHeading	Logged	-5.000 [kmph] = -1.389 m/s
	GPSFilter	Logged	-2.856m/s

Table 113: Requirement 74b (iii) Test Results

Test (iii) – GPS Location Data with Filter			
DAS Entry	**Parameter**	**Expected Value**	**Actual Value**
$601	VehSpd (Bytes 3-4)	5.0	5.0
$610	GIDVersion	GID-1 Version	1
	IntersectionId	GID-1 Int. ID	1
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
GPS Data Rx	Course	Logged	270
New Apprng Intn	Intersection ID	GID-1 Int. ID	1
	Intersection Type	GID-1 Int. Type	0
	RtofChngMovingAvg	-1.389m/s	-2.856
ROC method results	GPS	Logged	-1.389m/s
	CANSpeedHeading	Logged	-5.000 [kmph] = -1.389 m/s
	GPSFilter	Logged	-2.856m/s

Table 114: Requirement 74c Test Results

CAN Message	Parameter	Expected Value	Actual Value
$601	VehSpd (Bytes 3-4)	5.0	5.0
$619	IntersectionInrange (Byte-7, Bit-4)	1	1
	InteresectionClosing (Byte-7, Bit-3)	1	1
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
GPS Data Rx	Course	Logged	270
New Apprng Intn	Intersection ID	GID-1 Int. ID	1
	Intersection Type	GID-1 Int. Type	0
	RtofChngMovingAvg	-1.389m/s	-1.389
New Apprng Intn	Intersection ID	GID-2 Int. ID	2
	Intersection Type	GID-2 Int. Type	0
	RtofChngMovingAvg	-1.389m/s	-1.389
ROC method results	GPS	Logged	-1.389m/s
	CANSpeedHeading	Logged	-5.000 [kmph] = -1.389 m/s
	GPSFilter	Logged	-2.856m/s

Table 115: Requirement 75 Test Results

DAS Entry	Parameter	Expected Value	Actual Value
$601	VehSpd (Bytes 3-4)	5.0	5.0
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
GPS Data Rx	Course	Logged	270
New Apprng Intn	Intersection ID	GID-2 Int. ID	2
	Intersection Type	GID-2 Int. Type	0
	RtofChngMovingAvg	1.389m/s	-5.000 [kmph] = -1.389 m/s

Table 116: Requirement 77 Test Results

DAS Entry	Parameter	Expected Value	Actual Value
$610	IntersectionID (Bytes 1-4)	0	0
$618	GPSMode (Byte-0, Bits 7-4)	9	9
$619	GPSAccSatisfied (Byte-7, Bit-2)	0	0

Table 117: Requirement 76 Test Results

Log Entry	Expected Value	Actual Value
New Apprng Intn	Not Logged	Not Logged

Map Matching / Lane Identification Tests

A.18.1.2 Test Procedure

Req. #	Procedure
Pretest	• Setup the test environment as indicated in Figure 75. • In cicas-v.conf, reset the parameters to the default values. Then enable logging flags for the parameters listed in Table 118.

Req. #	Procedure

80	• In cicas-v.conf, set the parameters listed in Table 119 to the values indicated.
	• Select a SR file (or a portion of a SR file) that positions the vehicle within communication range of a RSE broadcasting a GID TOM, and also positions the vehicle inside a lane corresponding to the GID TOM being broadcast by the RSE.
	    • Ensure the lane in which the vehicle is positioned has likelihood greater than the default Map Match Single Match Threshold.
	    ⇨ An appropriate SR file is “12nF_07_10_22_36_22.rec”. The SR data starts at about time 18:36:45. Between 18:37:13 and 18:37:27 the vehicle is positioned inside a lane and lane confidence is > 40 when the default cicas-v.conf parameters are used.
	• Start the CICAS application in SR playback mode and replay the selected SR file (or portion).
	• Stop the CICAS application and check the Log File to verify Map Match Unsucc is logged. Record the results in Table 120.

81	• In cicas-v.conf, set the parameters listed in Table 121 to the values indicated. • Select a Scenario Replicator (SR) file (or a portion of a SR file) that positions the vehicle within communication range of a RSE broadcasting a GID TOM with multiple approaches in each direction of travel and a GPSC TOM, and also positions the vehicle off the GID but approaching the intersection, (e.g., in a parking lot.) ⇨ An appropriate SR file is “12nF_07_10_22_36_22.rec”. The SR data starts at about time 18:36:45. Between 18:39:35 and 18:39:55 the vehicle is positioned off the GID and moving toward the intersection. (The CICAS-V Data Visualization Tool may be useful for monitoring the time associated with SR playback.) • Boot the WSU in SR playback mode and play back the selected SR file (or portion) with the CICAS application. • Check the DAS to verify an Off GID indication is indicated. Record the results in Table 122.
82	• In cicas-v.conf, set the parameters listed in Table 123 to the values indicated. • Use the same SR file (or a portion of a SR file) that was used in the previous test. • Start the CICAS application in SR playback mode and replay the selected SR file (or portion). • Check the DAS to verify an Off GID indication is logged. Record the results in Table 124.
83	• In cicas-v.conf, set the parameters listed in Table 125 to the values indicated. • Use the same SR file (or a portion of a SR file) that was used in the previous test. • Start the CICAS application in SR playback mode and replay the selected SR file (or portion). • Check the DAS to verify the closest lane approach, Approach ID, Distance to Stop Bar, and Likelihood are logged. Record the results in Table 126.

<table>
<tr><th>Req. #</th><th>Procedure</th></tr>
<tr><td>84</td><td>In cicas-v.conf, set the parameters listed in Table 127 to the values indicated.
Select a SR file (or a portion of a SR file) that positions the vehicle within communication range of a RSE broadcasting a GID TOM with multiple approaches in each direction of travel and a GPSC TOM, and also positions the vehicle inside a lane corresponding to the GID TOM being broadcast by the RSE.
⇨ An appropriate SR file is “sr_orchard_loop.rec”. The SR data starts at about time 11:34:20. Between 11:36:10 and 11:36:22 the vehicle is positioned on the GID and approaching the intersection.
Start the CICAS application in SR playback mode and replay the selected SR file (or portion).
Check the Log File to verify only one approach and only one lane is logged. Record the results in Table 128.</td></tr>
<tr><td>85</td><td>In cicas-v.conf, set the parameters listed in Table 129 to the values indicated.
Select a SR file (or a portion of a SR file) that positions the vehicle within communication range of a RSE broadcasting a GID TOM with multiple lanes per approach and a GPSC TOM, and also positions the vehicle inside a lane corresponding to the GID TOM being broadcast by the RSE.
Ensure the lane in which the vehicle is positioned has likelihood greater than the Map Match Single Match Threshold indicated in Table 129.
⇨ An appropriate SR file is “sr_orchard_loop.rec”. The SR data starts at about time 11:34:20. Between 11:36:10 and 11:36:22 the vehicle is positioned inside a lane and lane confidence is > 40 when MapMatchAlwaysReturnMatch is set to 1.
Start the CICAS application in SR playback mode and replay the selected SR file (or portion).
Stop the CICAS application and check the Log File to verify only one approach and only one lane is logged. Record the results in Table 130.</td></tr>
<tr><td>86</td><td>In cicas-v.conf, set the parameters listed in Table 131 to the values indicated.
Use the same SR file (or a portion of a SR file) that was used in the previous test.
Start the CICAS application in SR playback mode and replay the selected SR file (or portion).
Stop the CICAS application and check the Log File to verify the Current Lane, Confidence Level, Current Approach ID, Distance to Stop Bar, and Distance to Nearest Lane Edge are logged. Record the results in Table 132.</td></tr>
</table>

<table>
<tr><th>Req. #</th><th>Procedure</th></tr>
<tr><td>87</td><td>In cicas-v.conf, set the parameters listed in Table 133 to the values indicated.
Select a SR file (or a portion of a SR file) that positions the vehicle within communication range of a RSE broadcasting a GID TOM with at least two approaches in the vehicle’s direction of travel and a GPSC TOM, and also positions the vehicle inside a lane corresponding to the GID TOM being broadcast by the RSE. Ensure the following:
No single lane has likelihood greater than the Map Match Single Match Threshold indicated in Table 133.
The “likelihood” difference between the highest likelihood approach and one other approach is less than the Approach Likelihood Difference indicated in Table 133.
⇨ An appropriate SR file is “12nF_07_10_22_36_22.rec”. The SR data starts at about time 18:36:45. Between 18:37:13 and 18:37:27 the vehicle is positioned inside a lane in one approach, with the adjacent lane belonging to a different approach.
Start the CICAS application in SR playback mode and replay the selected SR file (or portion).
Stop the CICAS application and check the Log File to verify the Lane, Approach, and Lane Confidence are logged for the most likely approaches. Record the results in Table 134.</td></tr>
<tr><td rowspan="2">88</td><td>In cicas-v.conf, set the parameters listed in Table 135 to the values indicated.
Use the same SR file from the above test.
Start the CICAS application in SR playback mode and replay the selected SR file (or portion).
Stop the CICAS application and check the Log File to verify the Lane, Approach, and Lane Confidence are logged for the most likely approaches. Record the results in Table 136.
Continue to the next test below.</td></tr>
<tr><td>In cicas-v.conf, modify the parameters listed in Table 137 to the values indicated.
Use the same SR file from the above test.
Start the CICAS application in SR playback mode and replay the selected SR file (or portion).
Stop the CICAS application and check the Log File to verify the Lane, Approach, and Lane Confidence are logged for the most likely approaches. Record the results in Table 138.</td></tr>
</table>

Req. #	Procedure

89	• In cicas-v.conf, reset the parameters to the default values. Then set the IntersectionIdentificationMethod parameter to 1 (CAN vehicle speed and GPS heading) and enable logging flags for the parameters listed in Table 118.
	• Reset the CAN application simulating the Netway box to the default configuration. Start the CAN simulator.
	• Configure the RSE Simulator to broadcast a GID TOM, GPSC TOM, and SPaT TOM on the CCH, and all corresponding to the same intersection ID, (IID 255).
	  ◦ Configure the GID TOM to include an approach with two lanes that have an overlapping segment.
	• Configure GPS Simulation data to simulate approaching the intersection described by the above GID on one of the lanes with the overlapping segment.
	  ◦ Ensure the GGA sentence indicates Fix Quality = 5, (Float RTK).
	• Start the CICAS application.
	• Start broadcasting the GID, GPSC, and SPaT TOMs using the RSE Simulator.
	• Send the GPS data to the WSU using the GPS Simulator.
	• Stop the CICAS application and check the log file to verify the lanes being reported change from a single lane number to multiple lanes as the vehicle enters the overlapping segment. Record the results in Table 139.
	  ◦ Also verify the lane being reported changes from no lane back to the current lane as the vehicle leaves the overlapping segment. Record the results in Table 140.

Table 118: Map Matching / Lane Identification Test Logging Parameters

Parameter	Value
GlobalLogFlag	1
MapMatchRsltsSuccLogFlag	1
MapMatchUnsuccLogFlag	1

Table 119: Requirement 80 Test Setup

Parameter	Value
MinAproachLikelihoodLocalCorr	100
MinAproachLikelihood	100

Table 120: Requirement 80 Test Results

Log Entry	Parameter	Expected Results	Actual Results
Map Match Unsucc	Intn	Logged	4
	LaneMatchState	Logged	VehLocationMapped
	DistFromClstLane	Logged	0.000
	LaneConf	Logged	0.000
Top Approach Matches	Approach	Logged	1
	Conf	Logged	85.474
	Approach	Logged	6
	Conf	Logged	7.761
Top Lane Matches	Lane	Logged	1
	Approach	Logged	1
	LaneConf	Logged	85.474
	Lane	Logged	1
	Approach	Logged	6
	LaneConf	Logged	5.116
	Lane	Logged	2
	Approach	Logged	6
	LaneConf	Logged	2.645

Table 121: Requirement 81 Test Setup

Parameter	Value
MapMatchAlwaysReturnMatch	0
OffGIDMethod	0 (Distance From Centerline method)

Table 122: Requirement 81 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$618	OffGID (Byte-1, Bit-2)	1	1

Table 123: Requirement 82 Test Setup

Parameter	Value
MapMatchAlwaysReturnMatch	0
OffGIDMethod	1 (Distance From Lane Edge method)

Table 124: Requirement 82 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$618	OffGID (Byte-1, Bit-2)	1	1

Table 125: Requirement 83 Test Setup

Parameter	Value
OffGIDMethod	0
MapMatchAlwaysReturnMatch	1
BaselineBehaviorFlag	1

Table 126: Requirement 83 Test Results

DAS ID	Parameter	Expected Results	Actual Value
$610	Approach (Byte-6)	Logged	6
	ApproachConf (Byte-7)	Logged	0
$611	Lane (Byte-1)	Logged	2 → 3
	LaneConf (Byte-7)	Logged	00
$618	DistStopBar (Bytes 4-5)	Logged	3A94 (= 149.96m)
	DistToCenterLine (Bytes 2-3)	Logged	016E (= 3.66m)

Table 127: Requirement 84 Test Setup

Parameter	Value
MapMatchAlwaysReturnMatch	0
BaselineBehaviorFlag	1

Table 128: Requirement 84 Test Results

DAS ID	Parameter	Expected Results	Actual Value
$610	Approach (Byte-6)	Logged	02
	ApproachConf (Byte-7)	Logged	DF
$611	Lane (Byte-1)	Logged	02
	LaneConf (Byte-7)	Logged	00
$618	DistStopBar (Bytes 4-5)	Logged	5A11→ 0000
	DistToCenterLine (Bytes 2-3)	Logged	0006 → 0000

Table 129: Requirement 85 Test Setup

Parameter	Value
MapMatchSingleMatchThreshold	40
MapMatchMaxNumberOfLanes	3
MapMatchAlwaysReturnMatch	1
BaselineBehaviorFlag	0

Table 130: Requirement 85 Test Results

Log Entry	Parameter	Expected Results	Actual Value
Top Lane Matches	Lane	Logged	2
	Approach	Logged	2
	LaneConf	Logged	49.079
	Lane	Not Logged	Not Logged as long as lane 2 confidence is > 40
	Approach	Not Logged	
	LaneConf	Not Logged	
	Lane	Not Logged	
	Approach	Not Logged	
	LaneConf	Not Logged	
Map Match Rslts Succ	Intn	Logged	2
	LaneMatchState	Logged	LaneCertain
	ProbLane	Logged	2
	ProbLaneConf	Logged	49.079
	LikelyAppr	Logged	2
	LikelyApprConf	Logged	49.079
	DistToStopBar	Logged	80.933
	DistToNearestLaneEdge	Logged	1.866

Table 131: Requirement 86 Test Setup

Parameter	Value
MapMatchSingleMatchThreshold	90
MapMatchMaxNumberOfLanes	3
MapMatchAlwaysReturnMatch	0
BaselineBehaviorFlag	0

Table 132: Requirement 86 Test Results

Log Entry	Parameter	Expected Results	Actual Value
Map Match Rslts Succ	Intn	Logged	2
	LaneMatchState	Logged	LaneCertain
	ProbLane	Logged	2
	ProbLaneConf	Logged	98.355
	LikelyAppr	Logged	2
	LikelyApprConf	Logged	98.355
	DistToStopBar	Logged	5.172.1.899
	DistToNearestLaneEdge	Logged	

Table 133: Requirement 87 Test Setup

Parameter	Value
MapMatchSingleMatchThreshold	100
MapMatchMaxNumberOfLanes	3
MapMatchAlwaysReturnMatch	1
BaselineBehaviorFlag	0
ApprLikelihoodDiffLocalCorr	100
ApprLikelihoodDiff	0

Table 134: Requirement 87 Test Results

Log Entry	Parameter	Expected Results	Actual Value
Top Approach Matches	Approach	Logged	1
	Conf	Logged	34.934
	Approach	Logged	6
	Conf	Logged	9.066
Top Lane Matches	Lane	Logged	1
	Approach	Logged	1
	LaneConf	Logged	34.934
	Lane	Logged	1
	Approach	Logged	6
	LaneConf	Logged	5.978
	Lane	Logged	2
	Approach	Logged	6
	LaneConf	Logged	3.088

Table 135: Requirement 88 Distance From Centerline Test Setup

Parameter	Value
MapMatchSingleMatchThreshold	100
MapMatchMaxNumberOfLanes	3
MapMatchAlwaysReturnMatch	0
BaselineBehaviorFlag	0
ApprLikelihoodDiffLocalCorr	100
ApprLikelihoodDiff	0
OffGIDMethod	0 (Distance From Centerline method)

Table 136: Requirement 88 Distance From Centerline Test Results

Log Entry	Parameter	Expected Results	Actual Value
Top Approach Matches	Approach	Logged	1
	Conf	Logged	78.319
	Approach	Logged	6
	Conf	Logged	13.046
Top Lane Matches	Lane	Logged	1
	Approach	Logged	1
	LaneConf	Logged	78.319
	Lane	Logged	1
	Approach	Logged	6
	LaneConf	Logged	8.602
	Lane	Logged	2
	Approach	Logged	6
	LaneConf	Logged	4.444
Map Match Rslts Succ	DistToStopBar	Logged	24.894
	DistToNearestLaneEdge	Logged	0.842

Table 137: Requirement 88 Distance From Lane Edge Test Setup

Parameter	Value
MapMatchAlwaysReturnMatch	0
BaselineBehaviorFlag	0
OffGIDMethod	1 (Distance From Lane Edge method)

Table 138: Requirement 88 Distance From Lane Edge Test Results

Log Entry	Parameter	Expected Results	Actual Value
Top Approach Matches	Approach	Logged	1
	Conf	Logged	78.319
	Approach	Logged	6
	Conf	Logged	13.046
Top Lane Matches	Lane	Logged	1
	Approach	Logged	1
	LaneConf	Logged	78.319
	Lane	Logged	1
	Approach	Logged	6
	LaneConf	Logged	8.602
	Lane	Logged	2
	Approach	Logged	6
	LaneConf	Logged	4.444
Map Match Rslts Succ	DistToStopBar	Logged	24.894
	DistToNearestLaneEdge	Logged	0.842

Table 139: Requirement 89 Test Results

Before Vehicle Enters Overlapping Segment			
Log Entry	**Parameter**	**Expected Results**	**Actual Value**
Top Approach Matches	Approach	Logged	1
	Conf	Logged	100.00
	Approach	Not Logged	Not Logged
	Conf	Not Logged	Not Logged
Top Lane Matches	Lane	Logged	1
	Approach	Logged	1
	LaneConf	Logged	100.00
	Lane	Not Logged	Not Logged
	Approach	Not Logged	Not Logged
	LaneConf	Not Logged	Not Logged
	Lane	Not Logged	Not Logged
	Approach	Not Logged	Not Logged
	LaneConf	Not Logged	Not Logged
While Vehicle is in Overlapping Segment			
Log Entry	**Parameter**	**Expected Results**	**Actual Value**
Top Approach Matches	Approach	Logged	1
	Conf	Logged	49.058
	Approach	Logged	2
	Conf	Logged	0.347
Top Lane Matches	Lane	Logged	1
	Approach	Logged	1
	LaneConf	Logged	24.529
	Lane	Logged	2
	Approach	Logged	1
	LaneConf	Logged	24.529
	Lane	Logged	1
	Approach	Logged	2
	LaneConf	Logged	0.347

Table 140: Requirement 89 Test Results

After Vehicle Leaves Overlapping Segment			
Log Entry	**Parameter**	**Expected Results**	**Actual Value**
Top Approach Matches	Approach	Logged	1
	Conf	Logged	100.00
	Approach	Not Logged	Not Logged
	Conf	Not Logged	Not Logged
Top Lane Matches	Lane	Logged	1
	Approach	Logged	1
	LaneConf	Logged	100.00
	Lane	Not Logged	Not Logged
	Approach	Not Logged	Not Logged
	LaneConf	Not Logged	Not Logged
	Lane	Not Logged	Not Logged
	Approach	Not Logged	Not Logged
	LaneConf	Not Logged	Not Logged

Warning Algorithm / State Machine Tests

A.18.1.3 Test Procedure

Req. #	Procedure
Pretest	• Setup the test environment as indicated in Figure 75. • Configure the CAN simulator connected to CAN1 to simulate the Netway box using the default configuration, and to capture CAN messages between the WSU and DVI. Start the CAN simulator. • In cicas-v.conf, reset the parameters to the default values. Then enable logging flags for the parameters listed in Table 141.

Req. #	Procedure
91 92	• Ensure the RSE and GPS Simulators are not running. • Boot the WSU and start the CICAS application, (not in SR playback mode.) • Allow CICAS to run for at least 5 seconds. • Check the DAS to verify the parameter values listed in Table 142 are as expected. Check the CAN simulator capturing the WSU-DVI CAN messages to verify the DVI State. Record the results in Table 142. • Stop the CICAS application and check the Log File to verify the Intersection in Range is FALSE. Record the results in Table 142.
	• Ensure the CAN simulator is not sending CAN data to the WSU. • Create/select a GID TOM for a signalized intersection, a SPaT TOM to correspond to the GID TOM, and a GPSC TOM to be broadcast by the RSE Simulator. • Create/select GPS data to place the vehicle within the lane of one approach defined by the GID TOM. • Broadcast the GID, GPSC, and SPaT data using the RSE Simulator. • Broadcast the GPS data using the GPS Simulator. • Start the CICAS application on the WSU. • Check the DAS to verify the parameter values listed in Table 143 are as expected. Check the CAN simulator capturing the WSU-DVI CAN messages to verify the DVI State. Record the results in Table 143. • Stop the CICAS application and check the Log File to verify the Intersection in Range is TRUE. Record the results in Table 143.

93
28
	• In cicas-v.conf, set the parameters listed in Table 144 to the values indicated.
	• Select a SR file (or a portion of a SR file) that positions the vehicle within communication range of a RSE broadcasting a GID TOM and a GPSC TOM, and also positions the vehicle inside a lane corresponding to the GID TOM being broadcast by the RSE. Ensure the following:
	  • The vehicle is approaching the intersection corresponding to the GID TOM being broadcast by the RSE.
	  • Vehicle speed remains greater than the default MinSignalSpeedThreshold.
	  • The vehicle position within a lane allows identifying a “most likely” approach.
	  ⇨ An appropriate SR file is “sr_orchard_loop.rec”. The SR data starts at about time 11:34:20. Between 11:36:10 and 11:36:22 the vehicle is positioned inside a lane approaching the intersection, and the signal remains green until the vehicle passes the stop bar.
	• Run athstats on the Linux console to verify WSU transmission of the RCMD TOM.
	• Start the CICAS application in SR playback mode and allow it to run until the vehicle crosses the intersection stop bar.
	• Monitor athstats to verify a RCMD is transmitted by the WSU. Record the results in Table 145.
	• Reset the parameters in Table 144 to their default values.

Req. #	Procedure

94	• Select a SR file (or a portion of a SR file) for a stop sign intersection. Ensure the following:
	  • The GID for the intersection is loaded in the WSU.
	  • The vehicle is approaching the intersection corresponding to the GID TOM being broadcast by the RSE.
	  • Vehicle speed is greater than the default MinStopSpeedThreshold.
	  • The vehicle position within a lane allows identifying a “most likely” approach.
	  ⇨ An appropriate SR file is “11nD_04_30_21_51_48.rec”. If SR file “10nO_04_18_20_01_36.rec” is run before the 11nD file, then the WSU will have the GID data for the 11nD intersection. The 11nD SR data starts at about time 17:52:15. Between 17:53:19 and 17:53:33 the vehicle approaches the intersection at a speed > 10 kph.
	• Start the CICAS application in SR playback mode and allow it to run until the vehicle stops at or near the intersection stop bar.
	• Check the DAS to verify the parameter values listed in Table 146 are as expected. Check the CAN simulator capturing the WSU-DVI CAN messages to verify the DVI State. Record the results in Table 146.
	• Stop the CICAS application and check the Log File to verify the latest Warning Algorithm State is “Intersection Equipped, No Warning”. Record the results in Table 146.

95	• Select a SR file (or a portion of a SR file) for a signalized intersection. Ensure the following:
	  • The vehicle is approaching the intersection corresponding to the GID TOM being broadcast by the RSE.
	  • Vehicle speed is greater than the default Low Vehicle Speed Threshold.
	  • The vehicle position within a lane allows identifying a “most likely” approach.
	  • The signal phase remains green for the approach being taken by the vehicle.
	  ⇨ An appropriate SR file is “sr_orchard_loop.rec”. The SR data starts at about time 11:34:20. Between 11:36:10 and 11:36:22 the vehicle is positioned inside a lane approaching the intersection, and the signal remains green until the vehicle passes the stop bar.
	• Start the CICAS application in SR playback mode and allow it to run until the vehicle reaches the intersection stop bar.
	• Check the DAS to verify the parameter values listed in Table 147 are as expected and record the results.
	• Stop the CICAS application.

Req. #	Procedure
96	• Reset the CAN application simulating the Netway box to the default configuration. Start the CAN simulator. • Create/select a GID TOM for a signalized intersection. • Create/select a GPSC TOM to correspond to the GID TOM. • Create/select GPS data to place the vehicle within the lane of one approach defined by the GID TOM. • Start broadcasting the GID and GPSC TOMs using the RSE Simulator. (Do not broadcast a SPaT TOM.) • Start the CICAS application on the WSU, (not in SR playback mode.) • Use the GPS Simulator to broadcast the GPS data. • Check the DAS to verify the parameter values listed in Table 148 are as expected and record the results. • Stop the CICAS application and check the Log File to verify the latest Warning Algorithm State is "Insufficient Information Available". Record the results in Table 148. • Stop the RSE Simulator and the GPS Simulator.

98	• In Dist_Warn_Sign.txt, set the parameters listed in Table 149 to the values indicated.
	• Select a SR file (or portion of a SR file) for a signalized intersection. Ensure the following:
	    • The vehicle is approaching the intersection corresponding to the GID TOM being broadcast by the RSE.
	    • The vehicle brake intent remains >= the Minimum Brake Intent value in Table 149.
	    ⇨ An appropriate SR file is "sr_orchard_loop.rec". The SR data starts at about time 11:34:20. Between 11:37:36 and 11:37:50 the vehicle approaches the intersection at a rate that results in a warning being issued if the default MinSignalBrakeIntent value is used.
	• Start the CICAS application in SR playback mode and allow it to run until the vehicle reaches the intersection stop bar.
	• Check the DAS to verify the parameter values listed in Table 150 are as expected and record the results.
	• Stop the CICAS application. (In Dist_Warn_Sign.txt, reset the parameter listed in Table 149 to the default value.)

<table>
<tr><th>Req. #</th><th>Procedure</th></tr>
<tr><td>99</td><td>

In Dist_Warn_Sign.txt, set the MinSignalSpeedThreshold parameter to the value indicated in Table 151.
Select a SR file (or portion of a SR file) for a signalized intersection. Ensure the following:

The vehicle is approaching the intersection corresponding to the GID TOM being broadcast by the RSE.
The signal phase remains red while the vehicle approaches the intersection.
The vehicle speed value is less than the Minimum Signal Speed Threshold value in Table 151.

⇨ An appropriate SR file is “sr_orchard_loop.rec”. The SR data starts at about time 11:34:20. Between 11:39:59 and 11:40:08 the vehicle approaches the intersection during a red light and at a rate that results in a warning being issued if the default MinSignalSpeedThreshold value is used.
Start the CICAS application in SR playback mode and allow it to run until the vehicle reaches the intersection stop bar.
Check the DAS to verify the parameter values listed in Table 152 are as expected and record the results.
Stop the CICAS application. (In Dist_Warn_Sign.txt, reset the MinSignalSpeedThreshold parameter to the default value.)

</td></tr>
<tr><td></td><td>

In Dist_Warn_Stop.txt, set the MinStopSpeedThreshold parameter to the value indicated in Table 151.
Select a SR file (or portion of a SR file) for a stop sign intersection. Ensure the following:

The vehicle is approaching the intersection corresponding to the GID TOM being broadcast by the RSE.
The vehicle speed value is less than the Minimum Stop Speed Threshold value in Table 151.
An appropriate SR file is “11nD_04_30_15_55_10.rec”. If SR file “10nO_04_18_20_01_36.rec” is run before the 11nD file, then the WSU will have the GID data for the 11nD intersection. The 11nD SR data starts at about time 11:55:30. Between 11:56:00 and 11:56:21 the vehicle approaches the intersection at a rate that results in a warning being issued if the default MinStopSpeedThreshold value is used.

Start the CICAS application in SR playback mode and allow it to run until the vehicle reaches the intersection stop bar.
Check the DAS to verify the parameter values listed in Table 153 are as expected and record the results.
Stop the CICAS application. (In Dist_Warn_Stop.txt, reset the MinStopSpeedThreshold parameter to the default value.)

</td></tr>
</table>

Req. #	Procedure

100	• Select a SR file (or portion of a SR file) for a signalized intersection. Ensure the following:
	  • The vehicle is approaching the intersection corresponding to the GID TOM being broadcast by the RSE.
	  • The signal phase remains green while the vehicle approaches the intersection.
	  • The vehicle speed remains greater than the default Minimum Signal Speed Threshold.
	  ⇨ An appropriate SR file is “sr_orchard_loop.rec”. The SR data starts at about time 11:34:20. Between 11:36:10 and 11:36:22 the vehicle is positioned inside a lane approaching the intersection, and the signal remains green until the vehicle passes the stop bar.
	• Start the CICAS application in SR playback mode and allow it to run until the vehicle reaches the intersection stop bar.
	• Check the DAS to verify the parameter values listed in Table 154 are as expected and record the results.
	• Stop the CICAS application.

	• Select a SR file (or portion of a SR file) for a signalized intersection. Ensure the following:
	  • The vehicle is stopped at the intersection corresponding to the GID TOM being broadcast by the RSE.
	  ⇨ An appropriate SR file is “sr_orchard_loop.rec”. The SR data starts at about time 11:34:20. Between 11:37:52 and 11:38:25 the vehicle remains stopped at the stop bar.
	• Start the CICAS application in SR playback mode and allow it to run while the vehicle is stopped at the intersection.
	• Check the DAS to verify the parameter values listed in Table 155 are as expected and record the results.
	• Stop the CICAS application.

Req. #	Procedure

101	• In cicas-v.conf, set the parameters listed in Table 156 to the values indicated.
	• Select a SR file (or portion of a SR file) for a signalized intersection. Ensure the following:
	    • The vehicle is approaching the intersection corresponding to the GID TOM being broadcast by the RSE.
	    • The signal phase remains red while the vehicle approaches the intersection.
	    • The vehicle speed remains greater than the default Minimum Signal Speed Threshold.
	    • The vehicle location never falls within the Minimum and Maximum Warning Distances to the intersection indicated in Table 156.
	    ⇨ An appropriate SR file is “sr_orchard_loop.rec”. The SR data starts at about time 11:34:20. Between 11:39:59 and 11:40:08 the vehicle approaches the intersection during a red light and at a rate that results in a warning being issued if the default Maximum Warning Distance value is used.
	• Start the CICAS application in SR playback mode and allow it to run while the vehicle approaches the intersection stop bar during a red light, but do not allow the vehicle position to fall within the Minimum and Maximum Warning Distances to the intersection indicated in Table 156.
	• Check the DAS to verify the parameter values listed in Table 157 are as expected and record the results.
	• Stop the CICAS application.

102	• In cicas-v.conf, set the parameters listed in Table 158 to the values indicated.
	• Select a SR file (or portion of a SR file) for a signalized intersection. Ensure the following:
	    • The vehicle is approaching the intersection corresponding to the GID TOM being broadcast by the RSE.
	    • The signal phase remains red while the vehicle approaches the intersection.
	    • The vehicle speed remains greater than the default Minimum Signal Speed Threshold.
	    • The vehicle location does fall within the Minimum and Maximum Warning Distances to the intersection indicated in Table 158.
	    ⇨ An appropriate SR file is “sr_orchard_loop.rec”. The SR data starts at about time 11:34:20. Between 11:39:59 and 11:40:08 the vehicle approaches the intersection during a red light and at a rate that results in a warning being issued if the default Distance To Warn values are used.
	• Start the CICAS application in SR playback mode and allow it to run until the vehicle reaches the intersection stop bar.
	• Check the DAS to verify the parameter values listed in Table 159 are as expected and record the results.
	• Stop the CICAS application.

Req. #	Procedure
103	• Reset the CAN application simulating the Netway box to the default configuration, then set the vehicle speed to 80kph. Start the CAN simulator. • In cicas-v.conf, set the parameter in Table 160 to the value indicated. • Configure the RSE Simulator to broadcast a GID TOM for a signalized intersection, a GPSC TOM, and a SPaT TOM all corresponding to the same intersection ID. • Configure the SPaT TOM with a signal phase that is flashing red for all possible approaches for the vehicle, (corresponding to the GPS Simulation data.) • Configure GPS Simulation data to simulate approaching and passing through the intersection described by the above GID at a rate of 80kph. • Start broadcasting the GID, GPSC, and SPaT TOMs using the RSE Simulator. • Start the CICAS application on the WSU, (not in SR playback mode.) • Use the GPS Simulator to broadcast the GPS data. • Watch the DAS to verify the Algorithm Status and DVI Icon State remains "Intersection Equipped, No Warning" as the vehicle approaches the intersection. Record the results in Table 161. • Stop the CICAS application.
104 106 107	• Reset the CAN application simulating the Netway box to the default configuration, then set the vehicle speed to 80kph. Start the CAN simulator. • In cicas-v.conf, restore the default parameter values. Then enable logging flags for the parameters listed in Table 141. • Ensure the Brakes Active setting in the Netway Simulator is set to FALSE. • Set the TSVWGEnable parameter in the Configuration Parameter File to 1. • Configure the RSE Simulator to broadcast a GID TOM for a signalized intersection, a GPSC TOM, as well as a SPaT TOM all corresponding to the same intersection ID. • Configure the SPaT TOM with a signal phase that remains red for all possible approaches for the vehicle, (corresponding to the GPS Simulation data.) • Configure GPS Simulation data to simulate approaching the intersection described by the above GID at a speed of 80kph. • Start broadcasting the GID, GPSC, and SPaT TOMs using the RSE Simulator. • Start the CICAS application on the WSU, (not in SR playback mode.) • Use the GPS Simulator to broadcast the GPS data. • Check the DAS and the CAN simulator capturing the WSU-DVI CAN messages to verify the parameters listed in Table 162 are as expected. Record the results in the table. • Run athstats on the Linux console to capture the TSVWG transmitted by the WSU. • Verify the TSVWG message is transmitted by the WSU. Record the results in Table 163. • Stop the CICAS application and check the Log File to verify the latest Warning Algorithm State is "Warning". Record the results in Table 162. • Use the Netway Simulator to change the Brakes Active setting to TRUE. • Start the CICAS application on the WSU, (not in SR playback mode.) • Check the DAS to verify the Brakes Active status was updated. Check the CAN simulator capturing the WSU-DVI CAN messages to verify the DVI State. Record the results in Table 164. • Stop the RSE Simulator and the GPS Simulator.

Req. #	Procedure
105	• Reset the CAN application simulating the Netway box to the default configuration, then set the vehicle speed to 80kph. Start the CAN simulator. • Reuse the GID TOM for the signalized intersection in the above test. • Reuse the GPSC TOM from the above test. • Reconfigure the SPaT TOM from the above test with a signal phase that remains green for one or more possible approaches for the vehicle, (corresponding to the GPS Simulation data.) ◦ Configure the time-to-next-phase to remain fixed at 30 seconds. • Record the SPaT Intersection ID in Table 165 as the expected value for Spat Rx, Intn. • Reuse the GPS Simulation data from the above test. • Start broadcasting the GID, GPSC, and SPaT TOMs using the RSE Simulator. • Start the CICAS application on the WSU, (not in SR playback mode.) • Use the GPS Simulator to broadcast the GPS data. • Check the DAS and record the Algorithm Status, Current Signal Phase, Time to Next Phase, Intersection Type, and DVI State in Table 165. • Stop the CICAS application and check the Log File to verify the latest Warning Algorithm State is "Intersection Equipped, No Warning" and the SPaT is being received. Record the results in Table 165.
	• Restart the CICAS application on the WSU and allow it to run for 3-5 seconds. • Stop the RSE Simulator. • Check the DAS to verify the Algorithm Status, Current Signal Phase, Time to Next Phase, Intersection Type, and DVI State continue to be recorded. Record the results in Table 166. • Stop the CICAS application and check the Log File to verify the Warning Algorithm State remained "Intersection Equipped, No Warning" and the SPaT is not being received. Record the results in Table 166. • Stop the GPS Simulator.

Table 141: Warning Algorithm / State Machine Test Logging Parameters

Parameter	Value
GlobalLogFlag	1
WarnAlgRsltsNoIntnInRngLogFlag	1
WarnStateChngLogFlag	1
CDISpatRxLogFlag	1

Table 142: Requirement 91 & 92 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	0	0
$619	DVIIconStates (Byte-6, Bits 0-3)	0	0
	OnGIDAppIdentifiedValid (Byte-7, Bit-1)	0	0
CAN Message ID	**Parameter**	**Expected Value**	**Actual Value**
$700	DVIState (Byte-0)	0 (OFF)	0
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
Warn Alg Rslts (No Intn In Rng)	IIMDataAvail	0 (GPS Invalid)	0
	MM/LIMDataAvail	0	0

Table 143: Requirement 91 & 92 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	0	0
$619	GPSDataValid (Byte-7, Bit-7)	1	1
	OnGIDAppIdentifiedValid (Byte-7, Bit-1)	0	0
	CANDataValid (Byte-7, Bit 0)	0	0
	SPaTDataValid (Byte-6, Bit-7)	1	0 (Re-run test since this has been confirmed in the past)
	DVIIconStates (Byte-6, Bits 0-3)	0	0
CAN Message ID	**Parameter**	**Expected Value**	**Actual Value**
$700	DVIState (Byte-0)	0 (OFF)	0
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
Warn Alg Rslts (No Intn In Rng)	IIMDataAvail	4 (CAN Data Invalid)	4
	MM/LIMDataAvail	0	0

Table 144: Requirement 93 Test Setup

Parameter	Value
TimeToPreempt	5.0
RCMDEnable	1

Table 145: Requirement 93 Test Results

athstats	Parameter	Expected Value	Actual Value
	RCMD	Transmitted	10/sec transmitted

Table 146: Requirement 94 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$600	BrkAct (Byte-0, Bit-6)	0	0
$611	AlgoStatus (Byte-6)	1	01
$618	IntersectionType (Byte-0, Bits 0-3)	2	2
$619	DVIIconStates (Byte-6, Bits 0-3)	1	1
	CANDataValid (Byte-7, Bit 0)	1	1
CAN Message ID	**Parameter**	**Expected Value**	**Actual Value**
$700	DVIState (Byte-0)	2 (Blue)	2
	DVIFrequency (Byte-2)	255 (ON)	FF (= 255)
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
Warn State Chng	WarnState	1	1

Table 147: Requirement 95 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$600	BrkAct (Byte-0, Bit-6)	0	0
$610	NumApproach (Byte-5)	4	04
$611	TTI (Bytes 2-5)	Decreasing	Decreasing
	AlgolStatus (Byte-6)	1	01
$612	RSEGPSMsTime (Bytes 4-7)	Increasing	Not changing – all 0's
$615	LocalGPSTimeInWk (Bytes 4-7)	Increasing	Increasing
$616	BSGPSStatus (Bytes 6-5)	Logged	00 20
$618	IntersectionType (Byte-0, Bits 0-3)	1	01
$619	IIDSpeedThreshMet (Byte-6, Bit-5)	1	1
	IntersectionInrange (Byte-7, Bit-4)	1	1
	IntersectionClosing (Byte-7, Bit-3)	1	1
	DVIIconStates (Byte-6, Bits 0-3)	1	01

Table 148: Requirement 96 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	0	0
$618	IntersectionType (Byte-0, Bits 0-3)	1	1
$619	DVIIconStates (Byte-6, Bits 0-3)	0	0
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
Warn Alg Rslts (No Intn In Rng)	IIMDataAvail	9 (Int. Identified)	9
	MM/LIMDataAvail	0	0
	IntnInRngSPAT	0	0
Warn State Chng	WarnState	0	0

Table 149: Requirement 98 Setup

Parameter	Value
MinSignalBrakeIntent	0

Table 150: Requirement 98 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	1	01
$618	IntersectionType (Byte-0, Bits 0-3)	1	1
$619	DVIIconStates (Byte-6, Bits 0-3)	1	1

Table 151: Requirement 99 Test Setup

Parameter	Value
MinSignalSpeedThreshold	80 kmph
MinStopSpeedThreshold	80 kmph

Table 152: Requirement 99 Signal Intersection Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	1	01
$618	IntersectionType (Byte-0, Bits 0-3)	1	1
$619	DVIIconStates (Byte-6, Bits 0-3)	1	1

Table 153: Requirement 99 Stop Sign Intersection Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	1	01
$618	IntersectionType (Byte-0, Bits 0-3)	2	2
$619	DVIIconStates (Byte-6, Bits 0-3)	1	1

Table 154: Requirement 100 Signal Intersection Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	1	01
$618	IntersectionType (Byte-0, Bits 0-3)	1	1
$619	DVIIconStates (Byte-6, Bits 0-3)	1	1

Table 155: Requirement 100 Signal Intersection Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	1	01
$618	IntersectionType (Byte-0, Bits 0-3)	1	1
$619	DVIIconStates (Byte-6, Bits 0-3)	1	1

Table 156: Requirement 101 Test Setup

Parameter	Value
MinWarnDistMeters	0
MaxWarnDistMeters	1

Table 157: Requirement 101 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	1	01
$618	IntersectionType (Byte-0, Bits 0-3)	1	1
$619	DVIIconStates (Byte-6, Bits 0-3)	1	1

Table 158: Requirement 102 Test Setup

Parameter	Value
MinWarnDistMeters	0
MaxWarnDistMeters	500
DistanceToWarnSign[1]	0.00

Table 159: Requirement 102 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	1	01
$618	IntersectionType (Byte-0, Bits 0-3)	1	1
$619	DVIIconStates (Byte-6, Bits 0-3)	1	1

Table 160: Requirement 103 Test Setup

Parameter	Value
HysteresisSpeed	180.0

[1] Set the value for all DistanceToWarnSign parameters (1 km/h to 200 km/h) to the value indicated.

Table 161: Requirement 103 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	1	1
$619	DVIIconStates (Byte-6, Bits 0-3)	1	1

Table 162: Requirement 104 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$600	BrkAct (Byte-0, Bit-6)	0	0
$611	AlgoStatus (Byte-6)	1	1
$618	IntersectionType (Byte-0, Bits 0-3)	1	1
$619	DVIIconStates (Byte-6, Bits 0-3)	3 (Warning)	3
CAN Message ID	**Parameter**	**Expected Value**	**Actual Value**
$700	DVIState (Byte-0)	1 (Red)	1
	DVIFrequency (Byte-2)	1-50 (Flashing)	10
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
Warn State Chng	WarnState	2 (Warning)	2

Table 163: Requirement 107 Test Results

athstats	Parameter	Expected Value	Actual Value
	TSVWG	transmitted	transmitted

Table 164: Requirement 106 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$600	BrkAct (Byte-0, Bit-6)	1	1
CAN Message ID	**Parameter**	**Expected Value**	**Actual Value**
$700	DVIState (Byte-0)	1 (Red)	1
	DVIFrequency (Byte-2)	1-50 (Flashing)	10

Table 165: Requirement 105 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	1	1
$612	CurSigPhase (Byte-0 – Byte-1, Bit 4)	1	00 1
	TimeNextPhase (Bytes 2-3)	3000	04 40 (Re-run test since this has been confirmed previously)
$618	IntersectionType (Byte-0, Bits 0-3)	1	1
$619	DVIIconStates (Byte-6, Bits 0-3)	1	1
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
Spat Rx	Intn	Logged	2
Warn State Chng	WarnState	1	1

Table 166: Requirement 105 Test Results

DAS ID	Parameter	Expected Value	Actual Value
$611	AlgoStatus (Byte-6)	1	1
$612	CurSigPhase (Byte-0 – Byte-1, Bit 4)	1	00 1
	TimeNextPhase (Bytes 2-3)	Less than recorded in Table 165.	04 40 (Need clarification on how this should be calculated)
$618	IntersectionType (Byte-0, Bits 0-3)	1	1
$619	DVIIconStates (Byte-6, Bits 0-3)	1	1
Log Entry	**Parameter**	**Expected Value**	**Actual Value**
Spat Rx	Intn	Not logged	Not Logged
Warn State Chng	WarnState	1	1

A.19 DVI Notifier Tests

Test Procedure

Req. #	Procedure
Pretest	• Setup the test environment as indicated in Figure 75. • In cicas-v.conf, enable logging flags for the parameters listed in Table 167. • Configure the CAN simulator connected to CAN1 to simulate the Netway box using the default configuration, and to capture CAN messages between the WSU and DVI. Start the CAN simulator.
109	• Reboot the WSU. • Check the CAN simulator connected to CAN1 to verify the DVI Heartbeat Sequence Counter in the first CAN message $700 received was set to 1. Record the results in Table 168.
111	• Check the CAN simulator connected to CAN1 to verify the DVI Heartbeat Sequence Counter in the 2nd through 5th CAN $700 messages received were each incremented by 1. Record the results in Table 168.
110	• Check the CAN simulator connected to CAN1 to verify CAN message$700 is being sent by the WSU every 100ms. Record the results in Table 169.
116	• Configure the CAN simulator to simulate the DVI sending CAN message $702 with an OBE to DVI Heartbeat Error. • Check the Log File to verify an OBE to DVI Heartbeat Error is logged. Record the results in Table 170.
117	• Configure the CAN simulator to simulate the DVI sending CAN message $702 with a DVI Heartbeat Sequence value that is different than the OBE to DVI Heartbeat Sequence sent in the previous CAN message $700. • Check the Log File to verify a DVI Heartbeat Sequence Error is logged. Record the results in Table 171.
118	• Configure the CAN simulator to simulate the DVI sending CAN message $702 with a DVI System Error.

Req. #	Procedure
	• Check the Log File to verify a DVI System Error is logged. Record the results in Table 172.
119	• Disconnect the connection to the CAN1 Port of the WSU to force a situation where the WSU will send CAN message $700 but not receive CAN message $702 before sending the next CAN message $700. • Check the Log File to verify a CAN message $702 Timeout error was logged. Record the results in Table 173.

Table 167: DVI Notifier Test Logging Parameters

Parameter	Value
GlobalLogFlag	1
DVIOBEHbErrLogFlag	1
OBEDVIHbErrLogFlag	1
DVISysErrorLogFlag	1
DVIOBETimeoutLogFlag	1

Table 168: Requirements 109 & 111 Test Results

CAN $700 Message Parameter	Sent CAN message $700	Expected Results	Actual Results
OBE to DVI Heartbeat Sequence	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	5

Table 169: Requirement 110 – 10Hz Test Results

Parameter	Expected Time Between Messages	Actual Time Between Messages
CAN $700 message	100ms	99-103ms
	100ms	99-103ms
	100ms	99-103ms
	100ms	99-103ms
	100ms	99-103ms
	100ms	99-103ms
	100ms	99-103ms
	100ms	99-103ms
	100ms	99-103ms
	100ms	99-103ms

Table 170: Requirement 116 Test Results

Log Entry	Expected Value	Actual Value
OBEDVIHbError	Logged	Logged

Table 171: Requirement 117 Test Results

Log Entry	Expected Value	Actual Value
DVIOBEHbError	Logged	Logged

Table 172: Requirement 118 Test Results

Log Entry	Expected Value	Actual Value
DVISysError	Logged	Logged

Table 173: Requirement 119 Test Results

Log Entry	Expected Value	Actual Value
DVIOBETimeout	Logged	Logged

A.20 Error Handler Tests

Test Procedure

Req. #	Procedure
121 122 123 124 125	• Reboot the WSU and start the CICAS application on the WSU. • Stop the CDI process by entering “killall -9 cdi”. • The CDI process is the Radio Handler, and includes the GID, SPaT, and GPSC Handlers. • Verify the WSU reboots after 10 seconds. Record the results in Table 174.
	• Reboot the WSU and start the CICAS application on the WSU. • Stop the CVL process by entering “killall -9 cvl”. • The CVL process is the GPS Handler. • Verify the WSU reboots after 10 seconds. Record the results in Table 174.
	• Reboot the WSU and start the CICAS application on the WSU. • Stop the CWA process by entering “killall -9 cwa”. • The CWA process includes Intersection Identification, Map Matching, and Warning Algorithm. • Verify the WSU reboots after 10 seconds. Record the results in Table 174.
	• Reboot the WSU and start the CICAS application on the WSU. • Stop the CLOGP process by entering “killall -9 clogp”. • The CLOGP process is the DAS and Error Handler. • Verify the WSU reboots after 10 seconds. Record the results in Table 174.
	• Reboot the WSU and start the CICAS application on the WSU. • Stop the DVIN process by entering “killall -9 dvin”. • The DVIN process is the DVI Notifier. • Verify the WSU reboots after 10 seconds. Record the results in Table 174.
	• Reboot the WSU and start the CICAS application on the WSU. • Stop the CVIP process by entering “killall -9 cvip”. • The CVIP process is the Vehicle Message Handler. • Verify the WSU reboots after 10 seconds. Record the results in Table 174.

Table 174: Requirement 121 to 125 Software Heartbeat Test Results

Log Entry	Parameter	Expected Value	Actual Value
CDI Process	killall -9 cdi	Reboots the WSU	Reboots the WSU
CVL Process	killall -9 cvl	Reboots the WSU	Reboots the WSU
CWA Process	killall -9 cwa	Reboots the WSU	Reboots the WSU
CLOGP Process	killall -9 clogp	Reboots the WSU	Reboots the WSU
DVIN Process	killall -9 dvin	Reboots the WSU	Reboots the WSU
CVIP Process	killall -9 cvip	Reboots the WSU	Reboots the WSU

A.21 Requirements Test Case Status

Table 175: ATP Requirements Test Case Status

Req. #	Category	Requirement Description	Test Status*	Notes
1.	Power Moding	The CICAS-V software shall manage the WSU power up and control the power down based on the IGN status	NA	Tested as part of SW component tests cases
2.	DVI Interface	The CICAS-V software shall interface to the DVI to control the DVI icon and input DVI status.	NA	
3.	Netway Box Interface	The CICAS-V software shall interface to a Netway Box to input vehicle and Netway box status, and output Netway heartbeat information.	NA	
4.	GPS Interface	The CICAS-V software shall interface to a NovAtel OEMV GPS Receiver.	NA	
5.		The software shall output GPS correction data received from the Roadside Equipment (RSE).	NA	
6.		The software shall use the NMEA and PPS inputs to obtain time and location.	NA	
7.	WSU Interface	The CICAS-V software shall interface to the WSU WAVE radio to transmit and receive messages over-the-air to/from the RSE.	NA	
8.	DAS Interface	The CICAS-V software shall interface to the DAS to output OBE status and input DAS status.	NA	
9.	Speaker Interface	The CICAS-V software shall interface to a speaker to generate audible alerts using the WSU Audio Out interface.	NA	
10.	CAN Heartbeat Test (Part of Vehicle Message Handler)	Periodically transmit CAN message $704 to the Netway box at a configurable interval.	P	
11.		Upon startup, initialize the OBE to Netway Heartbeat Sequence counter to 1.	P	
12.		Increment the Netway Heartbeat Sequence counter by 1 for each subsequent transmission of $704 to the Netway box	P	
13.		Upon receipt of CAN message $606 with an OBE to Netway heartbeat error indication, send an OBE to Netway Heartbeat error indication to the DAS Handler/Logger.	P	
14.		Upon receipt of CAN message $606 with Netway to OBE Heartbeat Sequence Counter that does not match the last number output in CAN message $704, send an Incorrect Heartbeat Sequence Counter Received to the DAS Handler/Logger.	P	
15.		If after sending a CAN message $704, a CAN message $606 is not received prior the next periodic submission of CAN message $704, send a CAN Message $606 Timeout indication to the DAS Handler/Logger.	P	
16.	CAN Message Test (Part of Vehicle Message Handler)	If CAN message $606 is received with Vehicle CAN Data Timeout set for any supported CAN, send a Vehicle CAN Timeout indication to the DAS Handler/Logger.	P	

Req. #	Category	Requirement Description	Test Status*	Notes
17.		If a complete set of CAN messages $600-$605 has not been received for a configurable amount of time, notify the other modules that the CAN data is invalid, and send a CAN Data Expiration indication to the DAS Handler/Logger.	P	
18.		If the WSU VIS indicates a CAN bus driver error, send an error indication to the DAS Handler/Logger.	NT	Error Handling logic deferred pending Phase II FOT
19.		Upon receipt of the group of CAN messages $600-605, discard the data if any of the messages are missing, and send an error indication to the DAS Handler/Logger.	F	See Table 176 for status
20.		Upon receipt of a complete group of CAN messages $600-605, output the data to other modules.	P	
21.		Upon receipt of CAN message $650 from the Netway box, output the contents of that CAN message $650 to other modules.	P	
22.		Periodically output a message to the DAS Handler/Logger containing data required for the DAS.	P	
23.	Radio Handler / Data Demux	When a WSA is received with supported TOM Framework and Layer Versions, differentiate between GID and GPSC services then initiate processing of the WSA.	P	
24.		When a WSM is received with supported TOM Framework and Layer Versions and with a TOM containing GPSC, GID, or SPaT data, differentiate between GPSC, GID, or SPaT services, then initiate processing of the WSM.	P	
25.		When a WSM is received without a supported TOM Framework Version, send an Unsupported TOM Framework Version indication to the DAS Handler/Logger.	P	
26.		When a WSM is received with a TOM that does not contain GPSC, GID, or SPaT data, send a Non-TOM WSM Received indication to the DAS Handler/Logger.	NT	Error Handling logic deferred pending Phase II FOT
27.		When requested, transmit a WSM with a TOM containing a TSVWG, and at the configurable data rate, transmit power, and priority.	D	
28.		When requested, transmit a WSM with a TOM containing a RCMD, and at the configurable data rate, transmit power, and priority.	P	

Req. #	Category	Requirement Description	Test Status*	Notes
29.		Periodically output a message to the DAS Handler/Logger containing data required for the DAS.	D	
30.		Obsolete	O	
31.	GPS Handler	If a GPS NMEA checksum error indication is received, send a GPS NMEA Checksum Error to the DAS Handler/Logger.	P	
32.		If NMEA data is received indicating no solution is available, send a No GPS Solution indication to the DAS Handler/Logger.	P	
33.		If NMEA data is received, send the NMEA data to other modules.	RR	See Table 177 for status
34.		If no GPS input has been received for a configurable period of time, send a GPS Data Invalid indication to other modules and send a GPS Data Expiration indication and a No GPS Input indication to the DAS Handler/Logger.	P	
35.		If no GPS solution has been received for a configurable period of time, send a GPS Data Invalid indication to other modules and send a GPS Data Expiration indication and a No GPS Solution (short term) indication to the DAS Handler/Logger. If no solution has been received for the configurable long term solution timeout, send a No GPS Solution indication (long term) to the DAS Handler/Logger.	P	
36.		If a TOM containing GPS Corrections is received with a Metric Object, send the elapsed time since the time indicated in the Metric Object to the DAS.	D	
37.		If a TOM containing GPS Corrections is received with the GPS Status flag indicating Unhealthy, discard the data in the TOM and send a GPSC RSE GPS Status not Healthy indication to the DAS Handler/Logger.	P	
38.		If a TOM containing GPS Corrections is received with non-zero length for the RTCM messages and valid RTCM 1005 and 1001 checksums, send the corrections to the GPS receiver.	P	
39.		If a TOM containing GPS Corrections is received with invalid RTCM 1005 or 1001 checksums, send a RTCM Checksum Error indication to the DAS Handler/Logger.	P	
40.		Send a periodic software heartbeat message to the DAS Handler/Logger at a configurable interval.	D	
41.	GID Database Handler	On startup, delete GID records that are older than the GID Expiration Period.	P	

Req. #	Category	Requirement Description	Test Status*	Notes
42.		On startup, if the size of the local GID database exceeds the configurable storage allocation, delete the records with the oldest Load Times to meet the storage allocation.	P	
43.		On startup, send the list of intersection reference points stored in the local GID database to the Intersection Identification module.	P	
44.		If a GID WSA is received with Intersection ID/Area ID, and Content Version that is already stored in the local GID database, update the Load Time for the corresponding intersection(s) and discontinue processing the WSA.	P	
45.		If a GID WSA is received from a single RSE, and the Intersection ID/Area ID and GID Content Version received in the WSA indicate the GID is not in the database, join the WBSS advertised in that WSA.	P	
46.		If GID WSAs are received from more than one RSE, and the Intersection ID/Area ID and GID Content Version received the GID WSAs indicate more than one GID is not in the database, join the WBSS corresponding to the RSE that is the closest based on the PSC Reference Point Latitude and Longitude received in the WSA.	F	See Table 176 for status
47.		If a GPSC WSA is received and (a) no WBSS has been joined based on GID WSAs, and (b) the GPSC is for the approaching intersection or no intersection has been selected, and (c) the status is healthy and monitored, join the WBSS corresponding to the GPSC WSA.	P	
48.		If a GID TOM is received with the Object ID value indicating a Metric Object, send the elapsed time since the time indicated in the Metric Object to the DAS.	D	
49.		If a GID contains an Area object, process individual intersection objects with the Area object. If a GID TOM is received, and the database does not contain GID data corresponding to the Intersection ID, store the contents of the GID in the database, including the load time, then send the Intersection Reference Point to the Intersection Identification module.	P	
50.		If the GID database does have sufficient space to store the latest GID received in a WSM, delete the records with the oldest Load Times to meet the storage allocation needs before adding the new GID record.	P	
51.		If a GID TOM is received with GID data that is already stored in the local GID database, update the Load Time for the corresponding intersection.	P	

Req. #	Category	Requirement Description	Test Status*	Notes
52.		If a GID TOM is received with GID data that is already stored in the local GID database, and the received data has a different Content Version than the stored data, update the Load Time and the GID data in the database, and send the Intersection Reference Point to the Intersection Identification module.	P	
53.		Periodically output a message to the DAS Handler/Logger containing data required for the DAS.	D	
54.	SPaT Handler	If a SPaT TOM is received with data corresponding to the approaching Intersection ID, send the SPaT data to the Warning Algorithm module.	P	
55.		If a SPaT TOM is received with data corresponding to the approaching Intersection ID, and with a Timestamp value in the SPaT Object ID exceeding a configurable threshold, send a SPaT Age Greater than Threshold indication to the DAS Handler/Logger.	P	
56.		If a SPaT TOM is received with data that does not correspond to the approaching Intersection ID, discard the SPaT data.	P	
57.		If a SPaT TOM is received with data corresponding to the approaching Intersection ID, and the Object ID value indicating a Metric Object, send the elapsed time since the time indicated in the Metric Object to the DAS.	D	
58.		If no valid SPaT data has been received for a configurable amount of time, send a Data Invalid indication to the Warning Algorithm module and send a SPaT Expiration indication to the DAS Handler/Logger.	P	
59.		Periodically output a message to the DAS Handler/Logger containing data for the DAS.	D	
60.	DAS Handler / Logger	Upon startup, initialize the OBE to DAS Heartbeat Sequence Counter to 1.	P	
61.		Increment the OBE to DAS Heartbeat Sequence Counter by 1 for each subsequent transmission of CAN message $606 to the DAS.	P	
62.		Send DAS messages $600-$606, $610-$619, and $650 at 10Hz intervals.	P	
63.		If CAN message $701 is received with a DAS System Error indication, send a DAS System Error indication to the DAS Handler/Logger.	P	
64.		If CAN message $701 is received with a DAS Boot Up Error indication, send a DAS Boot Up Error indication to the DAS Handler/Logger	P	
65.		If CAN message $701 is received with a DAS Shutdown Error indication, send a DAS Shutdown Error indication to the DAS Handler/Logger	P	

Req. #	Category	Requirement Description	Test Status*	Notes
66.		If CAN message $701 is received with an OBE to DAS Heartbeat Error indication, send an OBE to DAS Heartbeat Error indication to the DAS Handler/Logger	P	
67.		If CAN message $701 is received with a DAS Heartbeat Sequence value that is different than the OBE to DAS Heartbeat Sequence value sent in the previous CAN message $606, send a DAS Heartbeat Sequence Error indication to the DAS Handler/Logger	P	
68.		When a GID, GPSC, or SPaT message that contains a Metric Object is received by the OBE, immediately send DAS message $751 with the elapsed time since the time indicated in the Metric Object.	F	See Table 176 for status
69.		Data received by the OBE for DAS messages $600-$606, $610-$618, and $650, shall be buffered and sent to the DAS at the next 10 Hz interval.	P	
70.		If data received by the OBE must be logged (enabled by a configurable log mask), send the data record to the log file.	D	
71.		Obsolete	O	
72.	Intersection Identification	If GPS data is received that indicates HDOP exceeds a configurable threshold for a configurable time period, do not identify an intersection and send a GPS Insufficient Solution error to the DAS Handler/Logger.	P	
73.		If GPS data is received containing invalid Latitude or Longitude information, do not identify an intersection and send a GPS Insufficient Solution error to the DAS Handler/Logger.	P	
74.		If low speed filtering is disabled (by a configurable parameter) or if the vehicle speed exceeds a configurable threshold, identify the most likely approaching intersection as the one with (a) the rate of change in distance indicating the vehicle's direction of travel is toward the intersection and (b) meets the configurable intersection selection criteria which is either approaching most quickly or closest distance, based on a configurable parameter for selection using (1) CAN vehicle speed and GPS heading, (2) GPS location data, or (3) GPS location data with filter.	F	See Table 176 for status

Req. #	Category	Requirement Description	Test Status*	Notes
75.		If low speed filtering is enabled and the vehicle speed is equal to or lower than the configurable threshold, and an intersection was previously identified, identify the most likely intersection as the previously identified intersection. Log the rate of change in distance to the intersection, CAN vehicle speed, GPS reported heading, and velocity made good.	F	See Table 176 for status
76.		If low speed filtering is enabled and the vehicle speed is equal to or lower than the configurable threshold, and no intersection was previously identified, no intersection shall be identified.	P	
77.		If the identified intersection indicates lane level accuracy is required and the vehicle GPS quality indicates it is not in high accuracy mode, no intersection shall be identified.	P	
78.		If an intersection is identified following all of the above tests, log the rate of change in distance to the intersection, CAN vehicle speed, GPS reported heading, and velocity made good.	D	
79.		Periodically output a message to the DAS Handler/Logger containing data for the DAS.	D	
80.	Map Matching/ Lane Identification	If Approaching Intersection Information is not available, send a MapMatchUnsuccLogFlag to the DAS Handler/Logger.	P	
81.		If Approaching Intersection Information is available, and the configurable Always Return Match flag is FALSE, and the vehicle is not inside a lane, and the vehicle is determined to be off GID based on the configurable Distance from Centerline Method, send an Off GID indication to the DAS Handler/Logger.	P	
82.		If Approaching Intersection Information is available, and the configurable Always Return Match flag is FALSE, and the vehicle is not inside a lane, and the vehicle is determined to be off GID based on the configurable Distance from Lane Edge Method, send an Off GID indication to the DAS Handler/Logger.	P	
83.		If Approaching Intersection Information is available, and the configurable Always Return Match flag is TRUE and the configurable Baseline Behavior flag is TRUE, report the closest lane approach, Approach ID, Distance to Stop Bar, and "Likelihood" to the DAS.	F	See Table 176 for status
84.		If Approaching Intersection Information is available, and the vehicle is inside a lane and the configurable Baseline Behavior flag is TRUE, report the closest lane approach, Approach ID, Distance to Stop Bar, and "Likelihood" to the DAS.	F	See Table 176 for status

Req. #	Category	Requirement Description	Test Status*	Notes
85.		If Approaching Intersection Information is available, and the configurable Always Return Match flag is TRUE, and the configurable Baseline Behavior flag is FALSE, and a single lane has “likelihood” greater than a configurable threshold, report the Current Lane, Confidence Level, Current Approach ID, Distance to Stop Bar, and Distance to Nearest Lane Edge to the Log.	P	
86.		If Approaching Intersection Information is available, and the vehicle is inside a lane, and the configurable Baseline Behavior flag is FALSE, and a single lane has “likelihood” greater than a configurable threshold, report the Current Lane, Confidence Level, Current Approach ID, Distance to Stop Bar, and Distance to Nearest Lane Edge to the Log.	P	
87.		If Approaching Intersection Information is available, and the configurable Always Return Match flag is TRUE, and the configurable Baseline Behavior flag is FALSE, and no single lane has “likelihood” greater than a configurable threshold, report the Current Lane, Confidence Level, Current Approach ID, Distance to Stop Bar, and Distance to Nearest Lane for the most likely approaches with “likelihood” greater than a configurable threshold, up to a configurable maximum number of approaches, to the Log. (Include only approaches where the difference between the highest likelihood approach and their likelihood is less than or equal to a configurable likelihood difference threshold. The minimum and maximum likelihood threshold values used differ based on the availability of local GPS corrections)	P	
88.		If Approaching Intersection Information is available, and the vehicle is inside a lane, and the configurable Baseline Behavior flag is FALSE, and no single lane has “likelihood” greater than a configurable threshold, report the Current Lane, Confidence Level, Current Approach ID, Distance to Stop Bar, and Distance to Nearest Lane for the most likely approaches, up to a configurable maximum number of approaches, to the Log. (Verify Distance from Centerline Method and Distance from Lane Edge Method.) (Include only approaches where the difference between the highest likelihood approach and their likelihood is less than or equal to a configurable likelihood difference threshold.)	P	

Req. #	Category	Requirement Description	Test Status*	Notes
89.		If a GID is constructed such that 2 lanes have 1 or more node points that are identical (e.g., a right or left turn lane branches off an existing lane), then it is possible for the vehicle to exist in two lanes simultaneously. If this occurs, calculate the likelihood of all lanes in the intersection and select the highest likelihood lane.	P	
90.		Periodically output a message to the DAS Handler/Logger containing data required for the DAS.	D	
91.	Warning Algorithm	Upon startup, initialize status to Insufficient Information Available and set Intersection In Range to FALSE.	P	
92.		When an intersection is not within range, send Insufficient Information Available to the DVI and send the following to the DAS: (Algorithm Status = 0), (Current Signal Phase = 0), (Intersection Type = 0) If the CAN data is invalid, send Insufficient Information Available to the DVI and send the following to the DAS: (Algorithm Status = 0), (Current Signal Phase = 0), (Intersection Type = 0)	RR	See Table 177 for status
93.		If an intersection is within range and the time to stop bar for the most likely approach is less than the configurable time at which to preempt, and RCMD is enabled, transmit an RCMD.	P	
94.		When a stop sign intersection is within range and the approaching intersection ID has been received, send the warning status Intersection Equipped, No Warning, the Intersection type, and the Brakes Active status to the DVI.	P	
95.		When a signalized intersection is within range and the approaching intersection ID has been received, and valid SPaT data for the approaching intersection has been received within the configurable SPaT timeout applicable to the intersection in range indication, send the warning status Intersection Equipped, No Warning, the Intersection type, and the Brakes Active status to the DVI.	F	See Table 176 for status
96.		If valid SPaT data is not available for a signalized intersection, the warning status shall remain Insufficient Information Available.	P	
97.		Obsolete	O	
98.		If the vehicle brake intent is >= the configured Minimum Driver Intended Brake threshold, the warning status shall remain Intersection Equipped, No Warning.	P	

Req. #	Category	Requirement Description	Test Status*	Notes
99.		If the vehicle speed is below a configurable minimum threshold, the warning status shall remain Intersection Equipped, No Warning.	P	
100.		If the vehicle is not slowing down and for any possible approach the calculated timeToRed is >= the calculated timeToStopBar, or the timeToStopBar is the default maximum value (vehicle is stationary or moving backwards) the warning status shall remain Intersection Equipped, No Warning.	P	
101.		If the vehicle is not slowing down and for any possible approach the calculated timeToRed is < the calculated timeToStopBar, and the distToStopBar does not lie between the configured values for the Minimum and Maximum Warning Distances, the warning status shall remain Intersection Equipped, No Warning.	P	
102.		If the vehicle is not slowing down and for any possible approach the calculated timeToRed is < the calculated timeToStopBar, and the calculated distToStopBar lies between the configured values for the Minimum and Maximum Warning Distances, and the calculated distToStopBar is >= distForWarn, the warning status shall remain Intersection Equipped, No Warning.	P	
103.		When approaching a signalized intersection with a flashing red signal phase, check if there is a braking exception (i.e., the lowest speed over the last configurable hysteresis time is less than a configurable hysteresis speed threshold). If so, clear the warning and exit loop.	P	
104.		If the vehicle is not slowing down and for any possible approach the calculated timeToRed is < the calculated timeToStopBar, and the calculated distToStopBar lies between the configured values for the Minimum and Maximum Warning Distances, and the calculated distToStopBar is < distForWarn, and there is no braking exception, send the warning status Warning, the Intersection type, and the Brakes Active status to the DVI.	P	
105.		If the vehicle is not slowing down and the warning status is Intersection Equipped, No Warning, and the latest SPaT data is not current, estimate the time of the next phase based on information from the latest SPaT.	RR	See Table 177 for status
106.		If the Brakes Active status changes, update the status to the DVI on the next periodic update.	P	
107.		If the status is Warning and the TSVWG message is enabled, transmit a TSVWG message.	P	
108.		Output a message to the DAS Handler/Logger containing data required for the DAS.	D	

Req. #	Category	Requirement Description	Test Status*	Notes
109.	DVI Notifier	Upon startup, initialize the OBE to DVI Heartbeat Sequence counter to 1.	P	
110.		Periodically transmit CAN message $700 at a configurable interval.	P	
111.		Increment the OBE to DVI Heartbeat Sequence counter by 1 for each subsequent transmission of CAN message $700.	P	
112.		When a message is received from the DAS Handler/Logger, update the Malfunction and Maintenance flags based on the message.	D	
113.		When input is received from the Warning Algorithm, fill the contents of CAN messages $700 and $703, and update the DVI state machine based on the status received from the Warning Algorithm and the Maintenance and Malfunction flags updated by the DAS Handler/Logger. Send CAN message $703. Play the configurable audio file after the configurable delay in the configurable audible alert flag is enabled.	D	
114.		Complete processing a series of messages received from the Warning Algorithm and send CAN message $703 before processing any other series of messages received from the Warning Algorithm.	NT	Lower priority timing related test
115.		For the Warning and Equipped states, use the configurable Keep High and Keep Low durations to maintain the state for the desired duration and prevent an immediate change of state.	NT	Implicitly Verified
116.		If CAN message $702 is received with an OBE to DVI Heartbeat Error indication, send an OBE to DVI Heartbeat Error indication to the DAS Handler/Logger.	P	
117.		If CAN message $702 is received with a DVI to OBE Heartbeat Sequence Counter that does not match the last number output in CAN message $700, send an Incorrect DVI to OBE Heartbeat Sequence Counter error indication to the DAS Handler/Logger, and set the DVI to OBE Heartbeat Error indication in the next CAN message $700.	P	
118.		If CAN message $702 is received with a DVI System Error indication, send a DVI System Error indication to the DAS Handler/Logger.	P	
119.		If after sending a CAN Message $700, no CAN message $702 is received prior to the next periodic transmission of CAN message $700, send a CAN message $702 Timeout indication to the DAS Handler/Logger.	P	
120.		Output a message to the DAS Handler/Logger containing data required for the DAS.	D	

Req. #	Category	Requirement Description	Test Status*	Notes
121.	Error Handler (Part of DAS Handler / Logger)	Upon startup, activate the Hardware Watchdog timer.	P	
122.		Upon startup initialize the Software Watchdog timer.	P	
123.		If a message has not been received from all CICAS-V modules within a configurable time interval, allow the Software Watchdog timer to expire and log the error.	P	
124.		If a message has not been received from all CICAS-V modules within a configurable time interval, allow Hardware Watchdog timer to expire.	P	
125.		If the hardware watchdog timer has not been reset within a configurable time interval, reset the WSU.	P	
126.		When an error indication (set/clear) is received from a CICAS-V module, update the set of currently active errors if the indication persists for a configurable period of time, and set the OBE Health State to Normal, Maintenance, or Malfunction based on the most severe currently active error indication.	P	
127.		Obsolete	O	
128.		If the OBE Health State has changed, notify the DVI Notifier of the OBE Health.	P	
129.		Obsolete	O	

Table 176: Open ATP Test Issues Requiring a Fix

	Req. #	Issue	Analysis / Status
1.	19	In the Log File, missing CAN Message $605 not logged.	Known issue. The WSU VIS software sends Messages $600-605 as a group to the CICAS-V application when $605 is received to improve efficiency. If $605 is not received, the group will not be sent. A CAN timeout will occur and be logged but not a $605 missing event. The error is detected. Pending Phase II FOT to implement and incorporate fix.
2.	33	In DAS message $616, GST error ellipse is not reported correctly.	DAS Spec calls for GST to be calculated using error ellipse minor axis standard deviation and error ellipse major axis standard deviation. However the Map Matching code calculates the error ellipse using the standard deviation of latitude & longitude errors based on the interpretation of the Map Matching Specification. Need to confirm which is correct.

	Req. #	Issue	Analysis / Status
3.	46	In the Log File, WBSS corresponding to the RSE that is closest was not joined before the WBSS for a distant RSE was joined. (Both WBSS were eventually joined, though.)	Verified this is a bug in the v1.15 psc.c code. Fix tested and verified to correct the failure. Pending Phase II FOT to incorporate fix.
4.	68	In DAS message $751, Elapsed Seconds does not match the expected value.	Verified this is a bug in the v1.15 clogp.c code. Fix tested and verified to correct the failure. Pending Phase II FOT to incorporate fix.
5.	74b (iii)	In the Log File, RtofChngMovingAvg does not match expected value.	Spec was not clear w.r.t. numerator for ROC of distance calculation in GPS Location with Filter equation. The intended numerator was misinterpreted. Pending Phase II FOT to implement and incorporate fix.
6.	74b (i)(ii)(iii), 74c, 75	In the Log File, CAN ROC is reported as km/h instead of m/s.	Verified this is a bug in the v1.15 iimmli.c code. Pending Phase II FOT to implement and incorporate fix.
7.	83 & 84	In the Log File, LaneConf always = 0 when BaselineBehaviorFlag = 1.	Verified this is a bug in the v1.15 iimmli.c code. Pending Phase II FOT to implement and incorporate fix.
8.	95	In DAS message $612, RSEGPSMsTime is not changing.	RESGPSMsTime is not supported in the CICAS-V software. Pending Phase II FOT to determine need for fix.
9.	105	In DAS message $612, TimeNextPhase does not decrement when SPaT is lost.	Request clarification on whether this parameter is expected to maintain the last-received SPaT Time To Next Phase, or report the adjusted Time To Next Phase.

Table 177: Open ATP Test Issues for Re-Confirmation

	Req. #	Issue	Analysis / Status
1.	33	In DAS message $616, GST error ellipse is not reported correctly.	DAS Spec calls for GST to be calculated using error ellipse minor axis standard deviation and error ellipse major axis standard deviation. However the Map Matching code calculates the error ellipse using the standard deviation of latitude & longitude errors based on the interpretation of the Map Matching Specification. Need to confirm which method is correct.
2.	92	In DAS message $619 the SPaTDataValid value was not set properly.	Previous testing confirmed this was being set properly.
3.	105	In DAS message $612, TimeNextPhase does not decrement when SPaT is lost.	Clarification needed on whether this parameter is expected to maintain the last-received SPaT Time To Next Phase, or report the adjusted Time To Next Phase.

OBE Interface Definitions

A.22 OBE – Vehicle CAN Gateway Interface Definition

The purpose of this table is to define the CAN interface between the vehicle OBE and the vehicle CAN gateway in order to provide a consistent data logging interface between these two devices. All CAN message data follow the ‘Big Endian’ data format. Following the table are some general and referenced notes.

Table 178: OBE – Vehicle CAN Gateway Interface Definition

CAN Message	CAN ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
Vehicle-OBE 1	$600	ABS Active	ABSAct	0	7	1	BLN	N/A	$1=True; $0=False	100
		Brakes Active	BrkAct	0	6	1	BLN	N/A	$1=True; $0=False	
		Extended Brake Switch	ExtBrkSw	0	5	1	BLN	N/A	$1=True; $0=False	
		Stability Control Active	TrcCtrlAct	0	4	1	BLN	N/A	$1=True; $0=False	
		PRNDL	PRNDL	0	3	4	ENM	N/A	$1=P $2=R $3=N $4=OD $5=D $6=D1 $7=D2 $8=D3 $9=D4	
		Driver Intended Braking Level[7]	DrvIntendedBrk	1	7	8	ENM	NA	$00=No Braking $01=Braking	
		Reserved for future use	*Reserved*	*2*	*7*	*8*	*NA*	*NA*	*NA*	
		Brake Pedal Position	BrkPedalPos	3	7	8	UNM	0 - 255	E = N * 1	
		Battery Voltage	BattVolt	4	7	16	UNM	0 - 655.35 volts	E = N * .01	
		Yaw Rate	YawRate	6	7	16	SNM	-327.68 - 327.67 deg/s	E = N * .01	
Vehicle-OBE 2	$601	Steering Wheel Angle	StrWhlAng	0	7	16	SNM	-1024 - 1024 deg	E = N * 0.03125	100
		Outside Air Temperature	OutAirTemp	2	7	8	UNM	-40 - 87.5 deg C	E = N * .5 - 40	
		Vehicle Speed	VehSpd	3	7	16	UNM	0 – 655.35 kph	E = N * .01	
		Lateral Acceleration	LatAccel	5	7	10	UNM	-9.9 - 9.9462 m/s/s	E = N * 0.0194 - 9.9	
		Panic brake active	PanicBrake	6	5	1	BLN	NA	$1=True $0=False	
		Pre charge status	BrakePreChg	6	4	1	BLN	NA	$1=True $0=False	
		Headlight status	HdLights	6	3	1	BLN	NA	$1=True $0=False	
		Wiper Status	WiperSw	6	2	3	ENM	N/A	$0=Off $1=Delay 5 $2=Delay 4 $3=Delay 3 $4=Delay 2 $5=Delay 1 $6=Low $7=High	
		Interior Dim Status	IntDimState	7	7	1	BLN	NA	$1=Night time Dim $0=Day time	
Vehicle-OBE 3	$602	Longitudinal Acceleration	LonAccel	0	7	10	UNM	-9.9 - 9.9462 m/s/s	E = N * 0.0194 - 9.9	100
		Vertical Acceleration	VertAccel	1	5	10	UNM	-19.7 - 0.1462 m/s/s	E = N * 0.0194 - 19.7	

CAN Message	CAN ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
		Left Turn Signal	LftTurnSig	2	3	1	BLN	N/A	$1=True $0=False	
		Right Turn Signal	RtTurnSig	2	2	1	BLN	N/A	$1=True $0=False	
		Cruise Engaged	CrusEngd	2	1	1	BLN	N/A	$1=True $0=False	
		Cruise Override	CrusOvrd	2	0	1	BLN	N/A	$1=True $0=False	
		Accelerator Pedal Position %	AccelPos	3	7	8	UNM	0 - 100 %	E = N * 100/255	
		Vehicle Width	VehicleWidth	4	7	8	UNM	0 - 255 cm	E = N * 1	
		Vehicle Length	VehicleLength	5	7	10	UNM	100 – 611.5 cm	E = 100 + N * 0.5	
		Cruise set speed	CruiseSetSpeed	7	7	8	UNM	0-255 kph	E = N * 1	
Vehicle-OBE 4	$603	Wheel Velocity LF	WheelVelocityLF	0	7	16	SNM	-8192 - 8191.75 rpm	E = N * 0.25	100
		Wheel Velocity RF	WheelVelocityRF	2	7	16	SNM	-8192 - 8191.75 rpm	E = N * 0.25	
		Wheel Velocity LR	Wheel VelocityLR	4	7	16	SNM	-8192 - 8191.75 rpm	E = N * 0.25	
		Wheel Velocity RR	WheelVelocityRR	6	7	16	SNM	-8192 - 8191.75 rpm	E = N * 0.25	
Vehicle-OBE 5	$604	Tire Pressure LF	TirePressure LF	0	7	8	UNM	0 - 255 psi	E = N * 1	100
		Tire Pressure RF	TirePressure RF	1	7	8	UNM	0 - 255 psi	E = N * 1	
		Tire Pressure LR	TirePressure LR	2	7	8	UNM	0 - 255 psi	E = N * 1	
		Tire Pressure RR	TirePressure RR	3	7	8	UNM	0 - 255 psi	E = N * 1	
		Seatbelt status	Seatbelt	4	7	6	ENM	NA	$0=Driver unbelted, No front passengers $1=Driver belted, No front passengers $2=Driver unbelted, Front passenger un belted $3=Driver belted, Front passenger unbelted $4=Driver unbelted, Front latched empty $5=Driver belted, Front latched empty $6=Driver unbelted, Front passenger belted $7=Driver belted, Front passenger belted	
		Horn status	Horn	4	1	1	BLN	NA	$1=ON $0=OFF	
		ACC status	ACC	4	0	1	BLN	NA	$1=ON $0=OFF	
		ACC set speed	ACCSetSpd	5	7	8	UNM	0 – 255 kph	E = N * 1	
Vehicle-OBE 6	$605	Odometer reading	Odo	0	7	32	UNM	0 – 4.29497e+09 m	E = N * 1	100
Vehicle-OBE 7	$606	Vehicle CAN Data Timeout[1]	VehCANTimeout	0	7	4	ENM	N/A	$1=CAN1 timeout $2=CAN2 timeout $4=CAN3 timeout $8=CAN4 timeout	100

CAN Message	CAN ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
		Reserved bits up to the end of 4^{th} byte to be filled by OBE when transmitting to DAS				28	NA	NA		
		OBE to Netway6 Heartbeat Error[4]	NoOBENwHb	4	7	1	BLN	N/A	$1=Error $0=Healthy	
		Netway6 to OBE Heartbeat Sequence[2]	OBEHbSeq	5	7	24	UNM	1-1.67772e+07	E = N * 1	
Vehicle-OBE 8	$650[6]	ABS Active Mask	m_ABSAct	0	7	1	BLN	NA	$0=Not Available $1=Available	100
		Brakes Active Mask	m_BrkAct	0	6	1	BLN	NA	$0=Not Available $1=Available	
		Extended Brake Switch Mask	m_ExtBrkSw	0	5	1	BLN	NA	$0=Not Available $1=Available	
		Stability Control Active Mask	m_TrcCtrlAct	0	4	1	BLN	NA	$0=Not Available $1=Available	
		PRNDL Mask	m_PRNDL	0	3	1	BLN	NA	$0=Not Available $1=Available	
		Driver Intended Braking Level Mask	m_DrvIntendedBrk	0	2	1	BLN	NA	$0=Not Available $1=Available	
		Brake Pedal Position Mask	m_BrkPedalPos	0	1	1	BLN	NA	$0=Not Available $1=Available	
		Battery Voltage Mask	m_BattVolt	0	0	1	BLN	NA	$0=Not Available $1=Available	
		Yaw Rate Mask	m_YawRate	1	7	1	BLN	NA	$0=Not Available $1=Available	
		Steering Wheel Angle Mask	m_StrWhlAng	1	6	1	BLN	NA	$0=Not Available $1=Available	
		Outside Air Temperature Mask	m_OutAirTemp	1	5	1	BLN	NA	$0=Not Available $1=Available	
		Vehicle Speed Mask	m_VehSpd	1	4	1	BLN	NA	$0=Not Available $1=Available	
		Lateral Acceleration Mask	m_LatAccel	1	3	1	BLN	NA	$0=Not Available $1=Available	
		Panic brake active Mask	m_PanicBrake	1	2	1	BLN	NA	$0=Not Available $1=Available	
		Pre charge status Mask	m_BrakePreChg	1	1	1	BLN	NA	$0=Not Available $1=Available	
		Headlight status Mask	m_HdLights	1	0	1	BLN	NA	$0=Not Available $1=Available	
		Wiper Status Mask	m_WiperStatus	2	7	1	BLN	NA	$0=Not Available $1=Available	
		Interior Dim Status	m_IntDimStatus	2	6	1	BLN	NA	$0=Not Available $1=Available	
		Longitudinal Acceleration Mask	m_LonAccel	2	5	1	BLN	NA	$0=Not Available $1=Available	
		Vertical Acceleration Mask	m_VertAccel	2	4	1	BLN	NA	$0=Not Available $1=Available	
		Left & Right Turn Signal Mask	m_LftTurnSig	2	3	1	BLN	NA	$0=Not Available $1=Available	
		Crusise Engaged Mask	m_CrusEngd	2	2	1	BLN	NA	$0=Not Available $1=Available	
		Crusise Override Mask	m_CrusOvrd	2	1	1	BLN	NA	$0=Not Available $1=Available	
		Accelerator Pedal Position % Mask	m_AccelPos	2	0	1	BLN	NA	$0=Not Available $1=Available	
		Vehicle SizeMask (Width & Length)	m_VehicleSize	3	7	1	BLN	NA	$0=Not Available $1=Available	
		Cruise Set Speed	m_CruiseSetSpd	3	6	1	BLN	NA	$0=Not Available $1=Available	

CAN Message	CAN ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
		Wheel Velocity Mask	m_WheelVelocity	3	5	1	BLN	NA	$0=Not Available $1=Available	
		Tire Pressure Mask	m_TirePressure	3	4	1	BLN	NA	$0=Not Available $1=Available	
		Seatbelt status Mask	m_Seatbelt	3	3	1	BLN	NA	$0=Not Available $1=Available	
		Horn status Mask	m_Horn	3	2	1	BLN	NA	$0=Not Available $1=Available	
		ACC status Mask	m_ACC	3	1	1	BLN	NA	$0=Not Available $1=Available	
		ACC set speed Mask	m_ACCSetSpd	3	0	1	BLN	NA	$0=Not Available $1=Available	
		Odometer reading Mask	m_Odo	4	7	1	BLN	NA	$0=Not Available $1=Available	
		Vehicle Bus Timeout Mask	m_VehBusTimeout	4	6	4	ENM	NA	$1=CAN1 status available $2=CAN2 status available $4=CAN3 status available $8=CAN4 status available	
Vehicle-OBE 11	$703	Time Elapsed Since Last Warning	WarnTimeElapsed	0	7	8	UNM	0-255 seconds	E = N * 1	100
		Raw Warning Output	RawWarnTrigger	1	7	1	BLN	N/A	$0=Not Warning $1=Warning	
		Flexible Warning Output[3]	FlexWarnTrigger	1	6	7	ENM	N/A	$00=TBD by user $01=TBD by user $02=TBD by user $04=TBD by user $08=TBD by user $10=TBD by user $20=TBD by user $40=TBD by user	
Vehicle-OBE 12	$704	OBE to Netway6 Heartbeat Sequence[4]	OBENWHbSeq	0	7	24	UNM	1-1.67772e+07	E = N * 1	100
		Netway6 to OBE Heartbeat Error[5]	NWOBEHbError	3	7	1	BLN	NA	$0=Healthy $1=Error	

General Notes:

1. The 'Signal' text color indicates one of the following:

   BLACK: Data elements that are not supported

   BLUE: Data elements that are supported

   PURPLE: Data elements that may or may not be supported depending on the vehicle, will need to check the $650 message to determine availability

2. For Steering Wheel and Yaw Rate parameters, clockwise is considered positive while anti-clockwise is negative. Zero reference is the center position of the steering wheel with wheel position straight.

3. For lateral acceleration, the East is considered positive and West is considered negative, with respect to the vehicle coordinate system. The vehicle heading is considered North.

4. CAN message Ids $650, $600 - $606 are transmitted from Netway6 to OBE. Only the first nibble and the last 3 bytes of the $606 is populated in the Netway6.
5. CAN messages $600 - $605 are transmitted sequencially from the Netway6 as a group. The other CAN messages may be transmitted asynchronously.

Referenced Notes:

[1] TimeOut Period for monitoring the selected vehicle data messages is 100ms. Selecting the vehicle data elements to determine the 'Vehicle CAN Data Timeout' is OEM specific. These vehicle data elements are expected to be received (updated) by the Netway6 at least once within the 100ms time interval in order to set the 'healthy' status of the Vehicle CAN Data Timeout parameter. OEMs may monitor periodic reception of selected high frequency data messages such as one that contains the vehicle speed and brake information of their vehicle to determine the correct value (state) of the Vehicle CAN Data Timeout parameter.

[2] When the CICAS-V application in the OBE is started, OBE initializes $704 message with OBE to Netway6 Heartbeat Sequence counter set to 1. The subsequent $704 transmissions will increase this sequence counter by 1. Netway6 sets the Netway6 to OBE Heartbeat Sequence counter value in the $606 message to the OBE to Netway6 Heartbeat Sequence counter value received from the latest OBE to Netway6 $704 message.

[3] The Flexible Warning Output triggers were defined such that the DVI presented to the driver could be tailored to the individual vehicle and could expand in the future if necessary. The current use is discussed in Appendix 0 – 'DVI Notifier.'

[4] When Netway6 receives the $704 message, it sets the Netway6 to OBE Heartbeat Sequence counter value in $606 message to the OBE to Netway6 Heartbeat Sequence counter value and transmits the $606 message to the Netway6<->OBE CAN bus. Netway6 monitors periodic reception of the CAN message $704 from the OBE. If there is no $704 from the OBE for 400ms, the OBE to Netway6 Heartbeat Error flag in the CAN message $606 is set.

[5] The Netway6 to OBE Heartbeat error condition is set when there is no $606 message received by the OBE for 400ms.

[6] The message $650 shows the availability of data from different OEM vehicles. The available data elements from different OEM vehicles may be different, but is time invariant (static) for a given OEM vehicle.

[7] Determination of Driver Intended Braking Level value is OEM specific. For example, each OEM may decide the value of this parameter by taking some combination of their vehicle brake switch, brake pressure, brake pedal position, brake torque, and / or vehicle deceleration parameters according some OEM specific logic.

A.23 OBE – Visual DVI Interface Definition

The purpose of this table is to define the CAN interface between the vehicle OBE and the visual DVI in order to provide a consistent data logging interface between these two devices. All CAN message data follow the 'Big Endian' data format. Following the table are some general and referenced notes.

Table 179: OBE – Visual DVI Interface Definition

CAN Message	CAN ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
Vehicle-OBE 9	$700	DVI State Select	DVIState	0	7	8	ENM	N/A	$0=OFF $1=Red $2=Blue $3=AlernateRB	100
		DVI Icon Brightness	DVIBrightness	1	7	8	ENM	N/A	0=Min bright 10=Max bright	
		DVI Flash Frequency	DVIFrequency	2	7	8	ENM	0-50, 255	E = N * 50	
		DVI to OBE Heartbeat Error[3]	DVIOBEHbError	3	7	1	BLN	NA	$0=Healthy $1=Error	
		OBE to DVI Heartbeat Sequence[1]	OBEDVIHb	5	7	24	UNM	1-1.67772e+07	E = N * 1	
Vehicle-OBE 10	$702	DVI to OBE Heartbeat Sequence[1]	DVIHbSeq	0	7	24	UNM	1-1.67772e+07	E = N * 1	100
		DVI System Error	DVIError	3	7	1	BLN	NA	$0=Healthy $1=Error	
		OBE to DVI Heartbeat Error[2]	OBEDVIHbError	3	6	1	BLN	NA	$0=Healthy $1=Error	

General Notes:

- All of the 'Signal' items are supported.
- CAN message Id $700 is transmitted from OBE to DVI. CAN message Id $702 is transmitted from DVI to OBE as the periodic heartbeat message between the DVI and OBE.

Referenced Notes:

1 When the CICAS-V application in the OBE is started, OBE initializes the OBE to DVI Heartbeat Sequence counter to 1 and transmits the first $700 message on the Netway6-OBE-DVI CAN bus. The subsequent $700 transmissions will increase this sequence counter by 1. When DVI receives the first $700 message it replies with the $702 by broadcasting $702 message on the Netway6-OBE-DVI CAN bus. The DVI also sets the DVI to OBE Heartbeat sequence counter value to the OBE to DVI Heartbeat sequence counter value from the last received $700 message.

2 The OBE to DVI heartbeat error condition is set when there is no $700 message is received by the DVI for 400ms.

3 The DVI to OBE Heartbeat error condition is set when there is no $702 message is received by the OBE for 400ms.

A.24 OBE – DAS Interface Definition

The purpose of this table is to define the CAN interface between the vehicle OBE and the vehicle DAS in order to provide a consistent data logging interface between these two devices. All CAN message data follow the 'Big Endian' data format. Following the table are some general and referenced notes.

Table 180: OBE – DAS Interface Definition

CAN Message	ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
Vehicle-OBE 1	$600	ABS Active	ABSAct	0	7	1	BLN	N/A	$1=True; $0=False	100
		Brakes Active	BrkAct	0	6	1	BLN	N/A	$1=True; $0=False	
		Extended Brake Switch	ExtBrkSw	0	5	1	BLN	N/A	$1=True; $0=False	
		Stability Control Active	TrcCtrlAct	0	4	1	BLN	N/A	$1=True; $0=False	
		PRNDL	PRNDL	0	3	4	ENM	N/A	$1=P $2=R $3=N $4=OD $5=D $6=D1 $7=D2 $8=D3 $9=D4	
		Driver Intended Braking Level[14]	DrvIntendedBrk	1	7	8	ENM	NA	$00=No Braking $01=Braking	
		Reserved for future use	*Reserved*	*2*	*7*	*8*	*NA*	*NA*	*NA*	
		Brake Pedal Position	BrkPedalPos	3	7	8	UNM	0 - 255	E = N * 1	
		Battery Voltage	BattVolt	4	7	16	UNM	0 - 655.35 volts	E = N * .01	
		Yaw Rate	YawRate	6	7	16	SNM	-327.68 - 327.67 deg/s	E = N * .01	
Vehicle-OBE 2	$601	Steering Wheel Angle	StrWhlAng	0	7	16	SNM	-1024 - 1024 deg	E = N * 0.03125	100
		Outside Air Temperature	OutAirTemp	2	7	8	UNM	-40 - 87.5 deg C	E = N * .5 - 40	
		Vehicle Speed	VehSpd	3	7	16	UNM	0 – 655.35 kph	E = N * .01	
		Lateral Acceleration	LatAccel	5	7	10	UNM	-9.9 - 9.9462 m/s/s	E = N * 0.0194 - 9.9	
		Panic brake active	PanicBrake	6	5	1	BLN	NA	$1=True $0=False	
		Pre charge status	BrakePreChg	6	4	1	BLN	NA	$1=True $0=False	
		Headlight status	HdLights	6	3	1	BLN	NA	$1=True $0=False	
		Wiper Status	WiperSw	6	2	3	ENM	N/A	$0=Off $1=Delay 5 $2=Delay 4 $3=Delay 3 $4=Delay 2 $5=Delay 1 $6=Low $7=High	
		Interior Dim Status	IntDimState	7	7	1	BLN	NA	$1=Night time Dim $0=Day time	
Vehicle-OBE 3	$602	Longitudinal Acceleration	LonAccel	0	7	10	UNM	-9.9 - 9.9462 m/s/s	E = N * 0.0194 - 9.9	100
		Vertical Acceleration	VertAccel	1	5	10	UNM	-19.7 - 0.1462 m/s/s	E = N * 0.0194 - 19.7	
		Left Turn Signal	LftTurnSig	2	3	1	BLN	N/A	$1=True $0=False	
		Right Turn Signal	RtTurnSig	2	2	1	BLN	N/A	$1=True $0=False	
		Cruise Engaged	CrusEngd	2	1	1	BLN	N/A	$1=True $0=False	

CAN Message	ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
		Cruise Override	CrusOvrd	2	0	1	BLN	N/A	$1=True $0=False	
		Accelerator Pedal Position %	AccelPos	3	7	8	UNM	0 - 100 %	E = N * 100/255	
		Vehicle Width	VehicleWidth	4	7	8	UNM	0 - 255 cm	E = N * 1	
		Vehicle Length	VehicleLength	5	7	10	UNM	100 - 610 cm	E = N * 1	
		Cruise set speed	CruiseSetSpeed	7	7	8	UNM	0-255 kph	E = N * 1	
Vehicle-OBE 4	$603	Wheel Velocity LF	WheelVelocityLF	0	7	16	SNM	-8192 - 8191.75 rpm	E = N * 0.25	100
		Wheel Velocity RF	WheelVelocityRF	2	7	16	SNM	-8192 - 8191.75 rpm	E = N * 0.25	
		Wheel Velocity LR	Wheel VelocityLR	4	7	16	SNM	-8192 - 8191.75 rpm	E = N * 0.25	
		Wheel Velocity RR	WheelVelocityRR	6	7	16	SNM	-8192 - 8191.75 rpm	E = N * 0.25	
Vehicle-OBE 5	$604	Tire Pressure LF	TirePressure LF	0	7	8	UNM	0 - 255 psi	E = N * 1	100
		Tire Pressure RF	TirePressure RF	1	7	8	UNM	0 - 255 psi	E = N * 1	
		Tire Pressure LR	TirePressure LR	2	7	8	UNM	0 - 255 psi	E = N * 1	
		Tire Pressure RR	TirePressure RR	3	7	8	UNM	0 - 255 psi	E = N * 1	
		Seatbelt status	Seatbelt	4	7	6	ENM	NA	$0=Driver unbelted, No front pasengers $1=Driver belted, No front passengers $2=Driver unbelted, Front passenger un belted $3=Driver belted, Front passenger unbelted $4=Driver unbelted, Front latched empty $5=Driver belted, Front latched empty $6=Driver unbelted, Front passenger belted $7=Driver belted, Front passenger belted	
		Horn status	Horn	4	1	1	BLN	NA	$1=ON $0=OFF	
		ACC status	ACC	4	0	1	BLN	NA	$1=ON $0=OFF	
		ACC set speed	ACCSetSpd	5	7	8	UNM	0 – 255 kph	E = N * 1	
Vehicle-OBE 6	$605	Odometer reading	Odo	0	7	32	UNM	0 – 4.29497e+09 m	E = N * 1	100
Vehicle-OBE 7	$606	Vehicle CAN Data Timeout[1]	VehCANTimeout	0	7	4	ENM	N/A	$1=CAN1 timeout $2=CAN2 timeout $4=CAN3 timeout $8=CAN4 timeout	100
		Netway to OBE Heartbeat Error[2]	NetHbError	0	3	1	BLN	NA	$1=Error $0=Healthy	
		DAS to OBE Heartbeat Error[3]	DASHbError	0	2	1	BLN	NA	$1=Error $0=Healthy	
		GPS Solution Lost (long term)	GPSSolLost	0	1	1	BLN	NA	$1=Error $0=Healthy	

CAN Message	ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
		GPS Communication Timeout	NoGPS	0	0	1	BLN	NA	$1=Warning $0=Healthy	
		Inaccurate or No GPS Solution (short term)	GPSSolError	1	7	1	BLN	NA	$1=Error $0=Healthy	
		GPSC TOM Checksum Failure	GPSCChecksum Error	1	6	1	BLN	NA	$1=Error $0=Healthy	
		SPaT TOM Checksum Failure	SPATChecksum Error	1	5	1	BLN	NA	$1=Error $0=Healthy	
		SPaT Timeout	NoSPAT	1	4	1	BLN	NA	$1=Error $0=Healthy	
		SPaT GPS Age Warning	SPATGPSAgeW arn	1	3	1	BLN	NA	$1=Warning $0=Healthy	
		SPaT Blackout (intersection error)	SPATBlackout	1	2	1	BLN	NA	$1=Error $0=Healthy	
		GID TOM Checksum Failure	GIDCheckSumEr ror	1	1	1	BLN	NA	$1=Error $0=Healthy	
		DVI to OBE Heartbeat Error[4]	NoDVIHb	1	0	1	BLN	NA	$1=Error $0=Healthy	
		DVI System Error	DVISysError	2	7	1	BLN	NA	$1=Error $0=Healthy	
		Netway-OBE Timeout	NetwayOBETime out	2	6	1	BLN	NA	$1=Error $0=Healthy	
		DVI-OBE Timeout	DVIOBETimeout	2	5	1	BLN	NA	$1=Error $0=Healthy	
		Hardware Watchdog Reset Counter	HWResetCounte r	3	7	8	UNM	0-255	E = N * 1	
		OBE to Netway6 Heartbeat Error[5]	NoOBENwHb	4	7	1	BLN	NA	$1=Error $0=Healthy	
		OBE to DAS Heartbeat Error	NoOBEDASHb	4	6	1	BLN	NA	$1=Error $0=Healthy	
		OBE to DVI Heartbeat Error	NoOBEDVIHb	4	5	1	BLN	NA	$1=Error $0=Healthy	
		DSRC Radio Error	DSRCError	4	4	1	BLN	NA	$1=Error $0=Healthy	
		GPS NMEA Checksum Error	GPSNMEAChkEr ror	4	3	1	BLN	NA	$1=Error $0=Healthy	
		WSU Low Voltage Error	WSULowVoltErr	4	2	1	BLN	NA	$1=Low Voltage $0=Healthy	
		WSU Over Temp Error	WSUOverTempE rr	4	1	1	BLN	NA	$1=OverTemp $0=Healthy	
		OBE to DAS Heartbeat Sequence[6]	OBEDASHbSeq	5	7	24	UNM	1- 1.67772e+07	E = N * 1	
Vehicle-OBE 8	$650	ABS Active Mask	m_ABSAct	0	7	1	BLN	NA	$0=Not Available $1=Available	100
		Brakes Active Mask	m_BrkAct	0	6	1	BLN	NA	$0=Not Available $1=Available	
		Extended Brake Switch Mask	m_ExtBrkSw	0	5	1	BLN	NA	$0=Not Available $1=Available	
		Stability Control Active Mask	m_TrcCtrlAct	0	4	1	BLN	NA	$0=Not Available $1=Available	
		PRNDL Mask	m_PRNDL	0	3	1	BLN	NA	$0=Not Available $1=Available	
		Driver Intended Braking Level Mask	m_DrvIntendedB rk	0	2	1	BLN	NA	$0=Not Available $1=Available	
		Brake Pedal Position Mask	m_BrkPedalPos	0	1	1	BLN	NA	$0=Not Available $1=Available	
		Battery Voltage Mask	m_BattVolt	0	0	1	BLN	NA	$0=Not Available $1=Available	
		Yaw Rate Mask	m_YawRate	1	7	1	BLN	NA	$0=Not Available $1=Available	

CAN Message	ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
		Steering Wheel Angle Mask	m_StrWhlAng	1	6	1	BLN	NA	$0=Not Available $1=Available	
		Outside Air Temperature Mask	m_OutAirTemp	1	5	1	BLN	NA	$0=Not Available $1=Available	
		Vehicle Speed Mask	m_VehSpd	1	4	1	BLN	NA	$0=Not Available $1=Available	
		Lateral Acceleration Mask	m_LatAccel	1	3	1	BLN	NA	$0=Not Available $1=Available	
		Panic brake active Mask	m_PanicBrake	1	2	1	BLN	NA	$0=Not Available $1=Available	
		Pre charge status Mask	m_BrakePreChg	1	1	1	BLN	NA	$0=Not Available $1=Available	
		Headlight status Mask	m_HdLights	1	0	1	BLN	NA	$0=Not Available $1=Available	
		Wiper Status Mask	m_WiperStatus	2	7	1	BLN	NA	$0=Not Available $1=Available	
		Interior Dim Status	m_IntDimStatus	2	6	1	BLN	NA	$0=Not Available $1=Available	
		Longitudinal Acceleration Mask	m_LonAccel	2	5	1	BLN	NA	$0=Not Available $1=Available	
		Vertical Acceleration Mask	m_VertAccel	2	4	1	BLN	NA	$0=Not Available $1=Available	
		Left & Right Turn Signal Mask	m_LftTurnSig	2	3	1	BLN	NA	$0=Not Available $1=Available	
		Crusise Engaged Mask	m_CrusEngd	2	2	1	BLN	NA	$0=Not Available $1=Available	
		Crusise Override Mask	m_CrusOvrd	2	1	1	BLN	NA	$0=Not Available $1=Available	
		Accelerator Pedal Position % Mask	m_AccelPos	2	0	1	BLN	NA	$0=Not Available $1=Available	
		Vehicle SizeMask (Width & Length)	m_VehicleSize	3	7	1	BLN	NA	$0=Not Available $1=Available	
		Cruise Set Speed	m_CruiseSetSpd	3	6	1	BLN	NA	$0=Not Available $1=Available	
		Wheel Velocity Mask	m_WheelVelocity	3	5	1	BLN	NA	$0=Not Available $1=Available	
		Tire Pressure Mask	m_TirePressure	3	4	1	BLN	NA	$0=Not Available $1=Available	
		Seatbelt status Mask	m_Seatbelt	3	3	1	BLN	NA	$0=Not Available $1=Available	
		Horn status Mask	m_Horn	3	2	1	BLN	NA	$0=Not Available $1=Available	
		ACC status Mask	m_ACC	3	1	1	BLN	NA	$0=Not Available $1=Available	
		ACC set speed Mask	m_ACCSetSpd	3	0	1	BLN	NA	$0=Not Available $1=Available	
		Odometer reading Mask	m_Odo	4	7	1	BLN	NA	$0=Not Available $1=Available	
		Vehicle Bus Timeout Mask	m_VehBusTimeout	4	6	4	ENM	NA	$1=CAN1 status available $2=CAN2 status available $4=CAN3 status available $8=CAN4 status available	
OBE-DAS 1	$610	GID Map Version	GIDVersion	0	7	8	UNM	1 – 255	E = N * 1	100
		Intersection Id	IntersectionId	1	7	32	UNM	0 – 4.29497e+09	E = N * 1	
		Number of Approaches	NumApproach	5	7	8	UNM	0 – 255	E = N * 1	

CAN Message	ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
		Present Approach	Approach	6	7	8	UNM	0 – 255	E = N * 1	
		Approach Likelihood	ApproachConf	7	7	8	UNM	0 – 100%	E = N * 100/255	
OBE-DAS 2	$611	Number of Lanes	NumLanes	0	7	8	UNM	0 – 255	E = N * 1	100
		Present Lane	Lane	1	7	8	UNM	0 - 255	E = N * 1	
		Time To Intersection	TTI	2	7	32	UNM	0 – 4.29497e+09	E = N * 1	
		Algorithm Status	AlgolStatus	6	7	8	ENM	NA	$0=Algol Disabled $1=No Warning $2=Warning $3=TBD	
		Lane Likelihood	LaneConf	7	7	8	UNM	0 - 100 %	E = N * 100/255	
OBE-DAS 3	$612	Current Signal Phase	CurSigPhase	0	7	12	UNM	NA	$0=Not known $1=Green $2=Yellow $3=Red Flashing $4=Red	100
		Left Turn Phase	LeftPhase	1	3	4	UNM	NA	$0=Not known $1=Green $2=Yellow $3=Red Flashing $4=Red	
		Time to Next Phase	TimeNextPhase	2	7	16	UNM	0 – 655.35 sec	E = N * 0.01	
		RSE GPS ms time of the week	RSEGpsMsTime	4	7	32	UNM	0 – 4.29497e+09 ms	E = N * 1	
OBE-DAS 4	$613	Right Turn Phase	RightPhase	0	7	4	UNM	NA	$0=Not known $1=Green $2=Yellow $3=Red Flashing $4=Red	100
		Pedestrian Phase	PedPhase	0	3	4	UNM	NA	$0=Not known $1=Green $2=Yellow $3=Red Flashing $4=Red	
		Time to Left Phase	TimeLeftPhase	1	7	16	UNM	0 – 655.35 sec	E = N * 0.01	
		Time to Right Phase	TimeRightPhase	3	7	16	UNM	0 – 655.35 sec	E = N * 0.01	
		Time to Pedestrian Phase	TimePedPhase	5	7	16	UNM	0 – 655.35 sec	E = N * 0.01	
		Seconds Since Pedestrian Phase	ElapsedSecPed	7	7	8	UNM	0 – 255 sec	E = N * 1	
OBE-DAS 5	$614	Vehicle Latitude	VehLatitude	0	7	32	SNM	-214.748 – 214.748 deg	E = N * .0000001	100
		Vehicle Longitude	VehLongitude	4	7	32	SNM	-214.748 – 214.748 deg	E = N * .0000001	
OBE-DAS 6	$615	Vehicle Heading	VehHeading	0	7	16	UNM	0 – 655.35 deg	E = N * .01	100
		Vehicle Altitude	VehAltitude	2	7	16	SNM	-526.8 – 6026.7 m	E = N *.1 + 2750	
		Local GPS Milliseconds in Week	LocalGPSTimeInWk	4	7	32	UNM	0 – 4.29497e+09 ms	E = N * 1	
OBE-DAS 7	$616	Vehicle Lane Position	LanePos	0	7	8	UNM	0 – 255	E = N * 1	100
		Reserved for future use	*NA*	*1*	*7*	*8*	*NA*	*NA*	*NA*	
		PDOP	PDOP	2	7	8	UNM	0 - 25.5%	E = N * .1	
		HDOP	HDOP	3	7	8	UNM	0 - 25.5%	E = N * .1	
		Number of Satellites Used	NumSat	4	7	8	UNM	0 - 255	E = N * 1	

CAN Message	ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
		Base Station GPS Status[10]	BSGPSStatus	5	7	16	ENM	NA	E = N * 1	
		GST error ellipse[11]	GST	7	7	8	UNM	0 – 12.7m	E = N * .05	
OBE-DAS 8	$617	Local GPS Differential Age	DiffAge	0	7	16	UNM	0 – 65.535 sec	E = N * 0.001	100
		Local GPS Speed	GPSSpeed	2	7	16	UNM	0 – 655.35 m/s	E = N * 0.01	
		Local GPS Solution Age	GPSSolAge	4	7	16	UNM	0 – 655.35 sec	E = N * 0.01	
		Local GPS Week Number[9]	GPSWeekNum	6	7	16	UNM	0 – 65535 wk	E = N * 1	
OBE-DAS 9	$618	Rover GPS Quality / Mode	GPSMode	0	7	4	ENM	NA	$0=Fix not available $1=GPS fix $2=DGPS, CDGPS,.. $4=RTK fixed ambiguity solution $5=RTK float ambiguity solution $6=DR mode $7=Manual input $8=Simulator mode $9=WAAS mode	100
		Intersection Type	IntersectionType	0	3	4	ENM	NA	$0=Undefined $1=Signalized $2=Stop sign …	
		Road / Lane level	RoadLane	1	7	1	BLN	NA	$0=Road Level $1=Lane Level	
		WSA Reception	WSAReceived	1	6	1	BLN	NA	$0=Not Received $1=Received	
		SPaT Reception	SPATReceived	1	5	1	BLN	NA	$0=Not Received $1=Received	
		GID Reception	GIDReceived	1	4	1	BLN	NA	$0=Not Received $1=Received	
		GPSC Reception	GPSCReceived	1	3	1	BLN	NA	$0=Not Received $1=Received	
		Vehicle is Off GID	OffGID	1	2	1	BLN	NA	$0=On GID $1=Off GID	
		Distance to Lane / Approach Center Line[12]	DistToCenterLine	2	7	16	SNM	-3276 cm – 3276 cm	E = N * 0.1	
		Distance to Stop Bar	DistStopBar	4	7	16	UNM	0 – 65535 cm	E = N * 1	
OBE-DAS 10	$619	GPS Data Valid	GPSDataValid	7	7	1	BLN	NA	$0=Invalid Data $1=Valid Data	100
		HDOP Under Limit during Duration	HDOPUnderLimit DuringDuration	7	6	1	BLN	NA	$0=HDOP Over Limit for duration $1=HDOP OK	
		GPS Position Valid	GPSPosValid	7	5	1	BLN	NA	$0=Invalid Position $1=Valid Position	
		Intersection In Range	IntersectionInRange	7	4	1	BLN	NA	$0=No Int In Range $1=Int In Range	
		Intersection Closing	IntersectionClosing	7	3	1	BLN	NA	$0=No Int Closing $1=Int Closing	
		GPS Accuracy Requirement Satisfied	GPSAccSatisfied	7	2	1	BLN	NA	$0=Not Satisfied $1=Satisfied	
		On GID and Approach Identified and Validated	OnGIDAppIdentifiedValided	7	1	1	BLN	NA	$0=Not On or Not Identified $1=On, Identified and Validated	
		CAN Data Valid	CANDataValid	7	0	1	BLN	NA	$0=Invalid Data $1=Valid Data	
		SPaT Data Valid	SPaTDataValid	6	7	1	BLN	NA	$0=Invalid Data $1=Valid Data	

CAN Message	ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
		DVI Equipped Keep Low Active	DVIEquipKeepLowActive	6	6	1	BLN	NA	$0=Inactive $1=Active	
		IID Speed Threshold Met	IIDSpeedThreshMeg	6	5	1	BLN	NA	$0=Threshold Not Met $1=Threshold Met	
		DVIN Icon States	DVINIconStates	6	3	4	ENM	NA	$0=Standby $1=Equipped $2=Pre-Warning $3=Warning $4=Maintenance $5=Malfunction $6-15=RESERVED	
OBE-DAS 11	$701	DAS System Error	DASError	0	7	1	BLN	NA	$0=Healthy $1=Error	100
		DAS Boot Up Error	DASBootError	0	6	1	BLN	NA	$0=Healthy $1=Error	
		DAS Shutdown Error	DASShutdownError	0	5	1	BLN	NA	$0=Healthy $1=Error	
		OBE to DAS Heartbeat Error[7]	OBEHbError	0	4	1	BLN	NA	$0=Healthy $1=Error	
		DAS Run Mode	DASRUNMode	1	7	1	BLN	NA	$0=Not in RUN $1=RUN	
		DAS Shutdown Mode	DASShutdownMode	1	6	1	BLN	NA	$0=Not in Shutdown $1=Shutdown	
		DAS Video Health Status	DASVideoStatus	2	7	4	ENM	NA	$00=Healthy $01=Q1 Error $02=Q2 Error $04=Q3 Error $08=Q4 Error	
		DAS Radar Health Status	DASRadarStatus	2	3	4	ENM	NA	$00=Healthy $01=Front Error $02=Rear Error	
		DAS Hard-drive Free Space	DASHDSpace	3	7	8	UNM	0 - 255	E=(N * 400) MB	
		DAS Battery Voltage	DASBatVolt	4	7	8	UNM	0 - 255	E=(N /10) Volt	
		DAS Heartbeat Sequence[8]	OBEHbSeq	5	7	24	UNM	2-1.67772e+07	E = N * 1	
OBE-DAS 12[13]	$751	Negative Elapsed Time	NegTime	0	7	1	UNM	0 – 1	E = N * 1	Event-based
		Overflow Elapsed Time	Overflow	0	6	1	UNM	0 – 1	E = N * 1	
		Reserved	-	0	5	2	-	-	-	
		Elapsed Months	e_MM	0	3	4	UNM	0 – 11	E = N * 1	
		Elapsed Days	e_DD	1	7	5	UNM	0 – 30	E = N * 1	
		Elapsed Hours	e_hh	1	2	5	UNM	0 – 23	E = N * 1	
		Elapsed Minutes	e_mm	2	5	6	UNM	0 – 59	E = N * 1	
		Elapsed Seconds	e_ss	3	7	6	UNM	0 – 59	E = N * 1	
		Elapsed Milliseconds	e_ms	3	1	10	UNM	0 – 999	E = N * 1	
		Layer Type (LSByte)	LType	5	7	8	UNM	0 – 255	E = N * 1	
		Message Counter	MsgCtr	6	7	16	UNM	0 – 65535	E = N * 1	

General Notes:

6. The ‘Signal’ text color indicates one of the following:

   BLACK: Data elements that are not supported

   BLUE: Data elements that are supported

PURPLE: Data elements that may or may not be supported depending on the vehicle, will need to check the $650 message to determine availability

7. For Steering Wheel and Yaw Rate parameters, clockwise is considered positive, anti-clockwise is negative. Zero reference is the center position of the steering wheel with wheel position straight.
8. For lateral acceleration, the East is considered positive, West is negative, with respect to the vehicle coordinate system. The vehicle heading is considered North.
9. CAN message Ids $650, $600 - $606 are transmitted from Netway6 to OBE and then forwarded on to the DAS. Only the first nibble and the last 3 bytes of the $606 is populated in the Netway6 the remaining bits are filled in by the OBE prior to forwarding the message to the DAS to indicate additional error status. In addition the last three bytes of the $606 message are overwritten with the OBE – DAS Sequence Number. CAN message Id $650 is used to indicate the availability (static) of individual OEM vehicle data elements for the CICAS-V.
10. CAN message Id $751 is transmitted from OBE to DAS to log DSRC message reception, one per event.

Referenced Notes:

[1] TimeOut Period for monitoring the selected vehicle data messages is 100ms. Selecting the vehicle data elements to determine the 'Vehicle CAN Data Timeout' is OEM specific. These vehicle data elements are expected to be received (updated) by the Netway6 at least once within the 100ms time interval in order to set the 'healthy' status of the Vehicle CAN Data Timeout parameter. OEMs may monitor periodic reception of selected high frequency data messages such as one that contains the vehicle speed and brake information of their vehicle to determine the correct value (state) of the Vehicle CAN Data Timeout parameter.

[2] CAN message $606 is used for status reporting and heartbeat functionality between the Netway6 and OBE. The Netway to OBE heartbeat error condition is set when the $606 heartbeat message is not received by OBE for the timeout period of 100 ms or the OBE to Netway6 heartbeat error flag is set in $606.

[3] CAN message $701 reports DAS system status and is used in the heartbeat functionality between the DAS and OBE. The DAS to OBE heartbeat error condition is set when there is no $701 message received by the OBE for 100ms or the OBE to DAS heartbeat error flag in $701 is set.

[4] CAN message $702 reports DVI status and acts as the heartbeat between the DVI and OBE. The error condition DVI to OBE Heartbeat Error is set when there is no $702 heartbeat message is received by the OBE or the OBE to DVI heartbeat error flag in $702 is set.

[5] Netway6 should monitor the CAN message $704 from OBE. If there is no $704 from OBE for 100ms, then this error flag (OBE to Netway6 Heartbeat Error) in the $606 message is set.

[6] When CICAS-V application in the OBE is started, OBE initializes the OBE to DAS Heartbeat Sequence counter to 1 and transmits the first $606 message on the OBE<->DAS CAN bus. OBE increases this sequence counter every time it transmits a new $606 message to the DAS. When DAS receives this $606, it replies with the CAN message $701 with the DAS to OBE Heartbeat Sequence number set to the same value as the OBE to DAS Heartbeat Sequence number obtained from the last received $606 message.

[7] CAN message $606 reports CICAS-V status to DAS and acts as the heartbeat between the OBE and DAS. The error condition is set when $606 heartbeat message is not received by the DAS for consecutive 100ms.

[8] When DAS receives the OBE heartbeat message ($606), it reads the OBE Heartbeat Sequence counter and set this value in the DAS Heartbeat Sequence counter in message $701.

[9] The local GPS Week number can be calculated from year, month, date, hour, minute, second information from the onboard receiver's GPZDA NEMA sentence or directly reading from the GPSC DSRC message when available near intersections. Calculating from on-board receiver is recommended, due to wider availability of the information (not limited at CICAS-V intersections). Current offset (e.g., 14 seconds in 2007) between the GPS time and UTC time should be considered in this calculation.

[10] Base station GPS status is received from GPSC message. When there is no GPSC message for 4 seconds (timeout) this parameter is set to 0x00.

[11] GST error ellipse is calculated as the RMS value of error ellipse minor axis standard deviation and the error ellipse major axis standard deviation. For GST errors greater than 12.7 m, the maximum value of 12.7 is used.

[12] Distance to Lane / Approach Center Lane: This is the perpendicular distance to the Lane center line. When Lane match is not available, approach center line is used instead.

[13] When a GID, SPaT or GPSC message is received by the OBE, CAN message $751 is sent to DAS to log the event. Elapsed time is determined by subtracting the Metric Object's timestamp from the OBE current system time at time of reception. LType is the least significant byte of the DSRC message's Layer Type. MsgCtr is the Metric Object's Message Counter. NegTime is 1 only if the elapsed time calculation yields a negative time. Overflow is 1 only if the elapsed time exceeds 1 year.

[14] Determination of Driver Intended Braking Level value is OEM specific. For example, each OEM may decide the value of this parameter by taking some combination of their vehicle brake switch, brake pressure, brake pedal position, brake torque, and / or vehicle deceleration parameters according some OEM specific logic.

Intersection SW Development Details

A.25 GPS Corrections Service Provider Application Details

To achieve "which-lane" positioning determination, the CICAS-V overall system architecture requires that vehicles receive GPS corrections information provided by one or more ground reference stations. With the use of locally-produced GPS corrections from a fixed reference station, vehicle GPS position estimates are improved typically to better than 1.0 meter accuracy, often better than 0.5 meters. While there are few different available standards for specifying and transmitting correction data, CICAS-V uses the Radio Technical Commission for Maritime Services (RTCM) Differential GNSS (Global Navigation Satellite Systems) Services - Version 3 as developed by the RTCM Special Committee 104 (SC-104). This standard provides compact corrections encoding and options for various levels of correction sophistication. GPS performance testing conducted early in the CICAS-V development project demonstrated that RTCM v. 3.0 corrections applied to single (L1-only) frequency position estimates consistently yielded sub-meter positioning accuracy, which is deemed acceptable for CICAS-V system performance at reasonable cost levels in terms of GPS receiver equipment cost and over-the-air DSRC bandwidth utilization.

System Context

The GPSC SPA shall execute on a suitable computer host which is part of the roadside equipment (RSE) associated with a CICAS-V equipped intersection. The host computer shall either include or interface with a GPS receiver operating in fixed base station mode producing GPS corrections data. The host computer shall also include or interface with a DSRC/WAVE wireless network transceiver device. The primary function of the GPSC SPA is to receive corrections data periodically from the base station receiver, verify that it has no data errors, and broadcast it using DSRC to vehicles equipped to receive the messages.

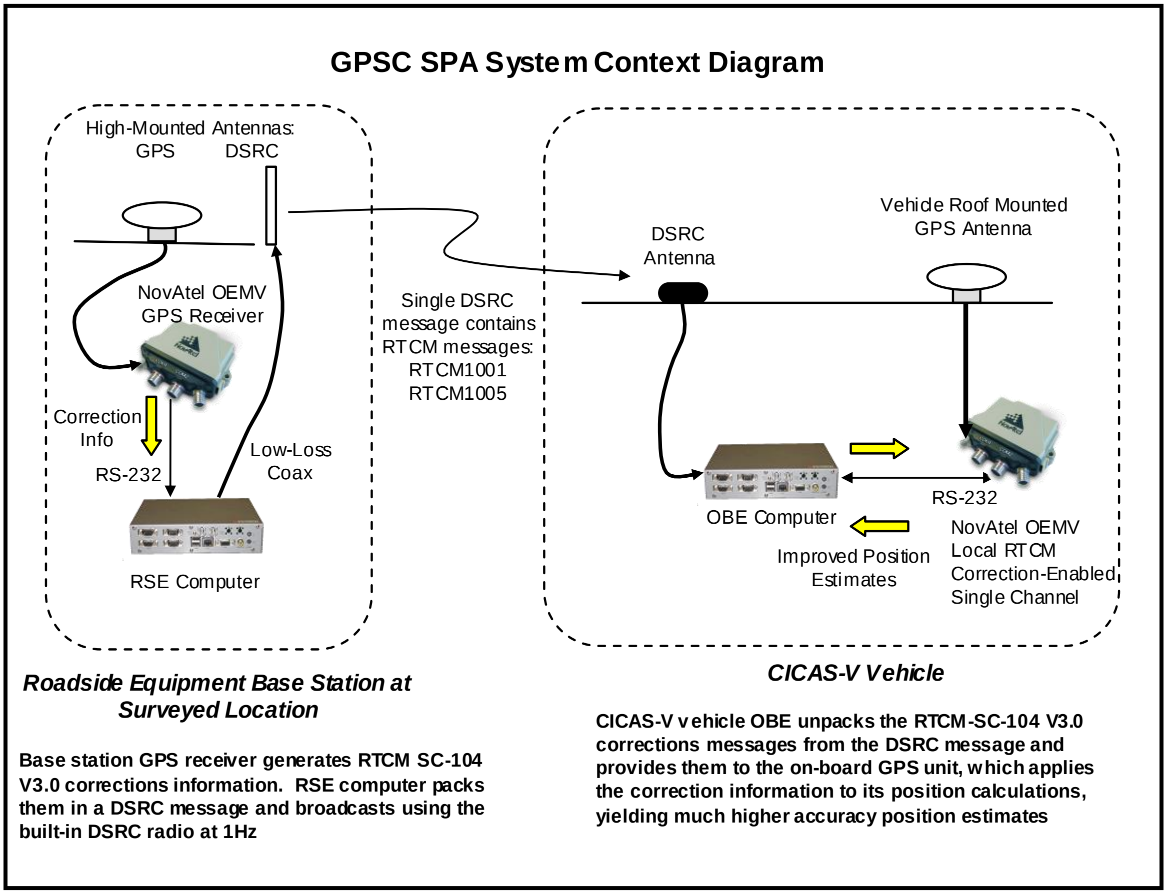

Figure 78: GPSC SPA System Context Diagram

Required and Optional Features

Required features include:

- Verification that the base station GPS receiver is operating correctly
- Reception of GPS corrections in RTCM v. 3.0 format at a rate of at least 1 Hz
- Detection of corrupt GPS corrections data
- Configurability of operation via command-line options and/or a configuration file
- Ability to broadcast RTCM v. 3.0 corrections data packets using the WAVE Short Message (WSM) protocol in TOM format
- Ability to broadcast into a specified DSRC channel, which is either the DSRC Control Channel or a specified Service Channel
- Transmission of WAVE Service Advertisement (WSA) messages on the DSRC Control Channel whenever the GPSC is broadcast into a DSRC Service Channel

Optional features may include:

- Synchronization of the host computer system clock time with GPS-based Universal Coordinated Time (UTC)

- Inclusion of timing data in the OTA DSRC message to assist in determining DSRC latency
- Display of current operational parameters on an operations console
- Logging of operational parameters to a local file or to external devices on a Controller-Area Network (CAN) bus
- Ability to broadcast corrections data packets via User Datagram Protocol (UDP) instead of WSM

Functional Requirements

A.25.1.1 Initialization Functions

Parse Configuration Settings and Options

At a minimum, the GPSC SPA shall accept the following configuration settings from either the command-line, configuration file, or equivalent mechanism that allows configuration without recompiling the GPS SPA executable software:

1. DSRC Channel – channel identification number for GPSC broadcast; may be a service channel or the control channel (which may be used as a default)
2. Intersection Identifier number – ID number used to uniquely identify CICAS-V intersections; only required for Wave Service Advertisement (WSA) broadcasts associated with Service Channel data broadcasts

The form and use of the configuration and option settings should follow the pattern established by other software on the same platform, especially other DSRC service provider applications which may be co-resident on the same roadside equipment host computer. If any required configuration setting is missing for which there is no default, the GPSC SPA should abort. If any configuration setting or option is not recognized or is provided an invalid value, the GPSC SPA may either abort or may attempt to continue with default values. In any of these cases, the GPSC SPA must provide a descriptive console and/or log message.

Establish Communications with the Base Station GPS Subsystem

The GPSC SPA must open a channel to receive RTCM and other data from the GPS base station receiver. If the channel is not available, the GPSC SPA must abort and provide a descriptive console and/or log message.

Initialize DSRC Communications Channel

The GPSC SPA must open the default or specified DSRC channel to broadcast GPS Correction TOMs. If the DSRC channel is not available, the GPSC SPA must abort and provide a descriptive console and/or log message.

Open Logging Interfaces *(If Applicable)*

If the implementation of the GPSC SPA provides special logging features, it should open the interface(s) needed to perform the logging functions (e.g., create a log file and open it, or open a CAN channel). If any logging interface is unavailable, the GPSC SPA may abort and the GPSC SPA shall attempt to provide a descriptive message through an interface that is.

A.25.1.2 Normal Operations

Receive and verify RTCM Corrections Information

The GPSC SPA may either actively request RTCM data for each broadcast period or passively receive it, assuming the GPS base station receiver is configured to periodically generate and emit the required RTCM 1005 and RTCM 1001 messages. The GPSC SPA shall check the initial three bytes of the RTCM v. 3.0 transport link layer header for a proper preamble/tag byte and message length as well as the first two bytes of the RTCM message for the message type (i.e., verify it is a 1001 or 1005). If any of these received data fields are not as expected, the GPSC SPA shall search for the next valid RTCM message.

If the initial bytes appear valid, the GPSC SPA shall receive the remaining bytes for the complete message, including the 24-bit cyclic redundancy check (CRC) three byte footer. The GPSC SPA shall compute the CRC value for the received message and compare it to the received CRC. If they do not match, the GPSC SPA shall reject the current message and search for the next valid message. If the CRC check passes, the complete message, including the three transport link layer header bytes and the CRC footer bytes, shall be saved in a buffer. If the RTCM message type is 1001, the GPS milliseconds "time of week" shall be extracted and stored.

Receive GPS Status Information

The GPSC SPA may either actively request GPS status data for each broadcast period, or passively receive it, assuming the GPS base station receiver is configured to periodically generate and emit the required NMEA messages. The following MNEA messages are sufficient to populate all NMEA-specific defined fields of the TOM 'RTCM_3.0_L1_CORRECTION' data object:

- $GPZDA: 'UTC Time and Date'
- $GPGSA: 'GPS DOP and Active Satellites'

Format Outgoing TOM Packet

The outgoing TOM for GPS corrections shall be formatted according to Appendix A.29 – 'WSM OTA Message Definitions.' The TOM format includes a general header and a layer structure that must be followed to ensure application interoperability. The GPSC SPA may transmit a TOM with only one layer containing a single 'RTCM_3.0_L1_CORRECTION' data object, which is object type #2 in the GPS Corrections layer type #3 and contains the following data fields:

- GPS Status Flags, indicating 'unhealthy' conditions and the source of corrections (see "fault handling: below for details)
- GPS Week Number: Unique week number as defined by the GPS network

- GPS Milliseconds in Week: Msec's since Midnight Sunday-to-Monday transition
- Length of both RTCM 1005 and RTCM 1001 data: Total number of bytes in the RTCM 1005 and RTCM 1001 complete messages
- RTCM 1005 Message Array (if available): Complete RTCM 1005 transport link layer message, including framing bytes; if available, the total message length is 25 bytes (3 header bytes, 19 RTCM data bytes, 3 CRC footer bytes)
- RTCM 1001 Message Array (if available): Complete RTCM 1001 transport link layer message, including framing bytes; if available, the total message length will vary between 36 and 101 bytes (3 header bytes, varying RTCM data bytes, 3 CRC footer bytes) depending on the number of satellites observed

The GPSC SPA shall implement the TOM header checksum as specified in Appendix A.29. The GPSC SPA shall transmit a TOM in each broadcast period even if the RTCM corrections are not currently available, however in this case, the 'Length of both RTCM 1005 and RTCM 1001' shall be zero and the GPS Corrections layer of the TOM will be closed immediately (i.e., the RTCM 1005 Message Array and RTCM 1001 Message Array are omitted from the TOM).

Transmit Outgoing TOM Packet

Transmit the GPSC TOM using the specified DSRC channel and configured networking protocol (WSM or UDP). For any transmission errors, provide a descriptive message to the console or logging interface.

Format Outgoing WAVE Service Advertisement

If the GPSC SPA is configured to broadcast corrections on a DSRC service channel, it must also broadcast WAVE Service Advertisement (WSA) messages on the control channel. The WSA must be formatted according to Appendix A.30 – 'WSA OTA Message Definition' and contains the following data:

- TOM Framework Version
- GPSC Format Version
- Intersection ID of intersection originating the GPSC message
- GPS Status bit mask containing the GPS Status flags as described below in the Fault Handling section

If the GPSC SPA is being broadcast on a service channel, and the GPS status changes, the WAVE Service Advertisement message must be reformatted and used in subsequent broadcasts.

Transmit Outgoing WSA Packet

If using a DSRC service channel to broadcast GPS corrections, transmit the GPSC WSA message each control channel period (10 Hz).

A.25.1.3 Fault Handling

Recognize Missing or Invalid RTCM Data

If the RTCM data is not of message types 1005 or 1001, or the calculated CRC value does not match the received CRC value at the end of the message, the GPSC SPA shall discard the received message data and continue to seek valid RTCM data. If the RTCM data is invalid or missing for more than 60 seconds, both the "Unhealthy" GPS Status Flag and the "Unmonitored" GPS Status Flag shall be set to true.

Announce Base Station "Health" and Corrections Source

The GPS Status Flags in the TOM data object shall be set or cleared as follows:

- Unhealthy flag: Set to TRUE if the GPS status (as determined by arrival of NMEA messages) is unavailable for more than 8 seconds or the RTCM corrections data is invalid or unavailable for more than 60 seconds, cleared (set to FALSE) otherwise
- Unmonitored flag: Set to TRUE if the RTCM is invalid or unavailable for more than 60 seconds, cleared (set to FALSE) otherwise
- PDOP > 5 flag: Set to TRUE if the GPS PDOP value is more than 5.0, cleared (set to FALSE) otherwise
- Satellites < 5 flag: Set to TRUE if the count of the observable GPS satellites used in positioning and corrections is less than five, cleared (set to FALSE) otherwise
- Local GPS Corrections flag: Set to TRUE if the broadcast corrections are generated from GPS signals received by a local base station with an antenna in close proximity (e.g., < 1 km) to the point of DSRC broadcast, cleared (set to FALSE) otherwise
- Network GPS Corrections flag: Set to TRUE if the broadcast corrections are generated by a relatively remote base station (e.g., >> 1 km) from the point of DSRC broadcast, cleared (set to FALSE) otherwise
- Other GPS Corrections flag: Cleared unless the corrections are from another kind of source or to be handled in a special way

Resynchronize RTCM communications

If at any time the received RTCM data does not appear correct, the GPSC SPA shall attempt to find the start of the next available RTCM message. *Design and Implementation Guidance*: Raw RTCM may include bytes mimicking the header, so don't flush all received bytes on an error, rescan for next potential message header.

Shutdown for Unhandled Exceptions

The GPSC SPA may shutdown for any unhandled exceptions, but if it does, it shall attempt to display and/or log a descriptive message and perform other cleanup tasks.

A.25.1.4 Shutdown Functions

Release Communications Resources

During application shutdown, the GPSC SPA shall close the communication link with the GPS receiver. If possible, it shall restore any prior settings for the communication port used to interface with the GPS receiver. The GPSC SPA shall un-register the GPS Corrections DSRC service (including stopping the broadcast of GPSC Wave Service Advertisement messages, if applicable). The GPSC SPA shall close the DSRC communications channel, and if applicable and possible, it shall restore any prior settings for the communication port used to interface with the GPS receiver

Close Logging Interfaces

During application shutdown, the GPSC SPA shall flush and close any log file(s) and/or close any CAN channel or other system interfaces used for data logging.

Constraints and Performance Requirements

A.25.1.5 Design Constraints

GPS Data Inputs

GPS data inputs are constrained to use RTCM v. 3.0 and NMEA messages.

A.25.1.6 Specified Performance Requirements

Size of RTCM Data

If present, the RTCM v. 3.0 message type 1005 is 25 bytes. The current GPS Correction TOM data object allows a total of 255 bytes for both the 1005 message and the 1001 message, so the GPSC SPA must be able to handle RTCM 1001 messages up to 230 bytes long, although the longest expected size of the RTCM 1001 message is 101 bytes.

Rate of GPS Correction TOM Data Object Broadcast

The GPSC SPA must be capable of broadcasting the GPS correction TOM data object at 1.0 Hz with a jitter of less than 100 ms. The GPSC SPA may optionally support additional configurable broadcast rates.

Rate of DSRC WAVE Service Advertisement Broadcast

Any DSRC service provider application transmitting data on a service channel is expected to broadcast a WAVE Service Advertisement (WSA) each control channel period on the control channel frequency. The control channel period is approximately 50 milliseconds of every 100 milliseconds, so the broadcast rate is 10.0 Hz. If the GPSC SPA does not use a service channel, it should not broadcast WSAs.

Latency to update DSRC WAVE Service Advertisement Content

If the GPS Status Flags change so that the WSA content must be reformatted and updated, this must be accomplished within 2.0 seconds (20 control channel frames). The broadcast of the GPS Corrections TOM may be briefly interrupted during the update of the WSA content.

"Gypsy" Reference Implementation Overview

A.25.1.7 General Description

"Gypsy" is a Linux command-line executable developed as part of the CICAS-V RSE development activities. It has no configuration file. It takes a number of command-line options that can be used to override default parameters. It takes one argument, the intersection ID number for the primary intersection associated with this instance of Gypsy. It can provide various levels of text output, from very minimal (good for background mode execution) to quite verbose (useful for troubleshooting) as it receives GPS information and broadcasts it via WAVE WSMs or UPD packets. The broadcast rate is paced by the reception rate of the RTCM data, which should be tightly synchronized with the GPS coordinated time.

A.25.1.8 Input Processing Description

On the DENSO WSU platform, Gypsy registers two different callback functions with the Time and Positioning Services (TPS) software API to receive the NMEA and RTCM data respectively. The main callback receives a data structure containing general GPS position estimation and status information from TPS. Some minimal validation of this data is performed, and the passed data stored. If successful, flags indicating "GPZDA valid" and "GPGSA valid" are set to TRUE. The UTC time from the NMEA-based GPS data is stored by Gypsy. The time of data arrival (the time the callback was called) is also noted from the Linux system clock. The RTCM callback receives the binary data for the two RTCM message types (1005 and 1001). The data is passed through to a general routine that verifies the RTCM message type and the CRC code. If the messages pass these checks, the appropriate "RTCM 1005 valid" or "RTCM 1001 valid" flags are set and the raw binary RTCM data is buffered. For the 1001 message, the GPS time of week is extracted. Again, the time of arrival is noted from the system clock.

Gypsy expects the GPS receiver to be pre-configured to automatically transmit all needed NMEA message and the two needed RTCM message every second (1.0 Hz). While Gypsy only requires info from the NMEA $GPGSA and $GPZDA messages, TPS also expects $GPGGA, $GPRMC, and $GPGST. If any Gypsy-required message is missing for more than a pre-defined period, error conditions will be reflected in the overall GPS Status Flags of Gypsy.

Assuming the RTCM1001 and the NMEA $GPZDA (or the equivalent TPS callback) data arrive within 0.7 seconds of each other, Gypsy will compare the reported UTC time and GPS Time of Week to determine a current "leap seconds" discrepancy between the UTC and GPS time references. However, Gypsy will assume 14 leap seconds (which is currently valid) if the UTC/GPS comparison analysis is not possible.

A.25.1.9 Output Processing Description

The complete TOM data structure is fairly simple to construct. Small helper routines are used to incrementally add data elements to a memory buffer. If a special latency "metric" configuration is used with Gypsy, an additional "metric" timestamp data object is added to the GPS Corrections layer, before the main RTCM_3.0_L1_CORRECTION object.

The GPS Week is computed from the UTC date information, taking into account the "leap seconds" discrepancy between UTC and GPS time references. The status flags are updated based on current conditions, such as the PDOP and number of GPS satellites used in computations as reported in the latest NMEA-based information, whether any of the needed callbacks/messages have not arrived within the timeout periods. If the RTCM 1005 and 1001 messages are available, their total length is determined and placed into the buffer followed by the raw bytes for both RTCM messages. If the RTCM messages are not available (either both are available or neither is available, given the input processing specified above), a zero is inserted into the TOM object at the RTCM total length field, and then the GPS corrections layer is immediately closed (omitting the RTCM message arrays from the TOM object). After the close object bytes, a final TOM footer byte is appended to the TOM buffer.

When the buffer is completely formatted, the overall TOM size is patched into the TOM header. Finally, the TOM CRC is computed and also patched into the TOM header.

The complete TOM is passed to a UCOM message for transmission via WAVE WSM or WAVE UDP. A tally of transmitted TOMs is kept and perhaps displayed depending on configuration settings.

There are multiple levels of "verbose" text output available while Gypsy is running, configurable through the command line (e.g., –v option is somewhat verbose, -vv is more verbose, -vvv is quite verbose, etc.). When enabled, the verbose output shows which information is being placed in the outgoing TOM (and WSA, if applicable) data broadcasts and other execution parameters.

A.25.1.10 System Management

Gypsy is a Linux command-line application and allows many configuration settings to be defined as command-line options. The allowed options and defaults are:

General Options

- -L List required NovAtel OEMV GPS Receiver 'log' messages.
- -# channel Broadcast on DSRC channel (default: Control Channel=178).
- -s ms Override sleep duration (default: 50 ms)
- -P PSID Override DSRC WAVE PSID number for GPS Corrections (default: 0x01E00003)
- -m Toggle metric object on/off (default: OFF)
- -r Toggle RSE-DAS CAN logging on/off (default: OFF)
- -d Toggle debug on/off (currently ON)

- -v Increment verbosity per use (default: level 0)
- -! bad-xxx Force incorrect output after 30 sec, where xxx is one of: health healthwsa checksum length (default: none)
- -h Display this usage information and exit

UDP Options

- -U Use UDP-IP over DSRC instead of WSM (default: no, use WSM)
- -b addr Define broadcast IP address (default: none)
- -p port Override default UDP port (default: 6062)
- -w Swap UDP port bytes (default: network order)
- -B Toggle UDP bind() call (default: disabled)
- -n Enable IP address lookup by hostname - narrows def. broadcast address (default: no)

The following example command line launches Gypsy to broadcast using WSMs in the Service Channel #172, to include the latency metric object in the TOM data, and to associate this DSRC broadcast with CICAS-V intersection #4 in the WSA content. The ampersand places the process's execution in Linux background mode (not tied to any interactive console).

./wgypsy -# 172 –m 4 &

"Gypsy" Observed Performance

- CPU utilization: according to the utility 'top,' Gypsy uses significantly less than 10% of the CPU resources on average.
- Broadcast Jitter: A sample of more than 30 broadcast seconds showed an average period of 1.000 seconds with a standard deviation of 2.2 milliseconds. Therefore, a jitter of 6.6 milliseconds is estimated (3 standard deviations).
- Robustness: While uninterrupted uptime was not explicitly tracked, manual intervention (restarting of specific services) was not required for multi-week extended periods, which indicates that Gypsy is quite robust in typical usage scenarios.
- GPS Signal Interruption Handling: Within seconds of GPS antenna signal loss, Gypsy correctly reflects the changes in satellite count and PDOP in the GPS Status Flags. After the RTCM timeout period (60 seconds), Gypsy correctly sets the "Unmonitored" and "Unhealthy" GPS Status Flags. If the GPS antenna signal is restored, the GPS receiver takes several (~5 to 30) seconds to reacquire satellite observations and restart GPS corrections computations, after which Gypsy will resume receiving RTCM 1001 messages and properly indicate the normal GPS Status Flags.
- GPS Connectivity Interruption Handling: After the NMEA timeout period (8 seconds), Gypsy correctly indicates an "Unhealthy" GPS Status Flag if the serial

link to the GPS base station receiver if broken. Immediately after connectivity is restored, the status returns to normal.

- GPS Positioning Improvement: Generally, the GPS positioning accuracy is worse than 1.0 meters when using the non-local WAAS corrections mode, which is the fall-back mode if the CICAS-V vehicle does not receive local corrections via DSRC. Within 4 seconds of receiving locally-generated RTCM GPS corrections, the vehicle positioning accuracy almost always improves to better than 1.0 meters, and for a large percentage of the time the RTCM data is available, the positioning accuracy is better than 0.5 meters.

Enhancements

No specific enhancements were documented.

A.26 GID Service Provider Application Details

The Cooperative Intersection Collision Avoidance System (CICAS) for Violations (CICAS-V) overall system architecture requires vehicles to receive and retain information about intersection geometry before reaching an equipped intersection, to match themselves to the correct lane or approach, depending on available positioning accuracy, and to allow onboard applications to correlate vehicle presence with other more dynamic data such as Signal Phase and Timing (SPaT) information.

CICAS-V-equipped intersections broadcast information relevant to vehicles over Dedicated Short Range Communications (DSRC). Intersection geometry is communicated with the Geometric Intersection Description (GID) message. The GID is a core message of the CICAS-V architecture. It is the responsibility of the GID Service Provider Application (SPA), also known as the GID Server, to broadcast GIDs. Typically this server runs on signalized Roadside Equipment (RSE) installations, serving their particular intersection's GID and, potentially, other nearby stop sign GIDs.

A GID is a compact, binary representation of the critical portion of an intersection's geometry. The small size makes GIDs suitable for broadcast on a DSRC Service Channel (SCH), freeing bandwidth on the Control Channel (CCH). Access to the SCH is achieved through IEEE 1609.4 channel switching.

System Context

The GID SPA shall execute on a suitable computer host which is part of the RSE associated with a CICAS-V equipped intersection. The GID SPA requires no external interface except DSRC. However, a production version could potentially benefit from a network connection to a Traffic Management Center (TMC) or other management capability for GID updates.

Required and Optional Features

Required features:

- Ability to broadcast GID messages using the WAVE Short Message (WSM) protocol in Transportation Object Message (TOM) format over DSRC
- Ability to broadcast over selectable Service Channel or Control Channel

- Activation of WAVE Service Advertisement (WSA) messages on the DSRC Control Channel when GID SPA is broadcasting on a Service Channel
- Ability to alter broadcast period, if desired
- Ability to combine multiple intersection GIDs in any specified order into individual WSMs
- Ability to shutdown cleanly on command

Optional features:

- Optional inclusion of RSE system time and message counter in GIDs for engineering studies
- Optional logging of activity at multiple levels of detail to a local file
- Optional logging of DSRC activity to external Data Acquisition System

Functional Requirements

No functional requirements were explicitly documented.

Constraints and Performance Requirements

Rate of GID Broadcast

The GID SPA must be capable of broadcasting a GID message over DSRC every 500 ms with less than 100 ms latency. Other broadcast rates may be supported.

"Giddy" Reference Implementation Overview

A.26.1.1 General Description

"Giddy" is a 32-bit Linux program developed as part of the CICAS-V RSE development activities. It has no configuration file. Rather, command-line options can be used to override default parameters. The DSRC broadcast rate is controlled by a command-line option alone. The server can provide various levels of debug output on stdout for troubleshooting and can even be made silent for unattended operation. DSRC broadcast is on the default system SCH or another channel specified on the command line. The server is best invoked by a system startup script for consistent unattended operation. Only one server process at a time may operate on a given platform.

A.26.1.2 Initialization

A.26.1.2.1 Command Line Options

Giddy is a Linux command-line application and allows many configuration settings to be defined as command-line options. Default values are used for options that are not provided on the command line. The GID SPA supports the following command line options (default values in parentheses):

Options Overview:

-i iid	Select Intersection ID(s) (2)
-a aid	Enable and specify Area ID (currently 0, 0=disabled)
-j	Toggle Signalized vs. Stop Sign (OFF)

-P psid	Override default GID PSID (0x01E00002)
-C	Toggle channel type (SCH)
-S num	Override Service Channel Number (180)
-r	Toggle logging to DAS on/off (OFF)
-G cport	Override DAS CAN port number (2)
-m	Toggle Metric Object inclusion (OFF)
-s ms	Override sleep duration (500 ms)
-c CV	Override GID ContentVersion (0)
-f FV	Override GID FormatVersion (2)
-x	Allow ContentVersion to be dynamic
-y	Write libpcap header to output file (OFF)
-o file	Write first GID to file then quit ()
-l file	Get approach labels from file ()
-L	Enable label load
-u	Use approach numbers instead of ApproachIDs
-t mask	Specify test bit mask (0x00000000)
-d	Toggle debug on/off (ON)
-v	Increment verbosity per use (0)
-h	Display this usage information then quit

Giddy parses its command line options immediately on launch. Certain options have option arguments, which are processed in turn. Any command line arguments are not processed.

Options Details

Intersection Id (-i iid)

The most important option is **–i**. It specifies the intersection (or intersections, see below) to be broadcast. It takes one argument: a string containing either an integer representing a unique Intersection ID (sometimes shortened to IID) or a comma-separated list of Intersection IDs (with no embedded white space). It defaults to intersection 2.

The GID server supports the following intersections required by the CICAS-V Pilot-FOT system. Note that the Virginia intersections are in the vicinity of the Virginia Tech campus and Christiansburg, VA.

Supported Intersection IDs:

1	5th Ave and El Camino, Redwood City/Atherton, CA
2	10 Mile and Orchard Lake Rd, MI

3	11 Mile and Drake Rd, MI
4	12 Mile and Farmington Rd. MI
5	Smart Rd, VA
6	Depot St and Franklin St, VA
7	Elm St / Independence Blvd and Franklin St, VA
8	Peppers Ferry (114) and Franklin St (460), VA
9	Dumbarton x Oakwood and El Camino, CA
10	Columbia and Waverly, CA
11	Dexter and Waverly, CA
12	Hickok and First, VA
13	Sheltman and College, VA
14	College and Depot, VA
15	Magna Carta and Constitution, VA
16	Constitution and Liberty, VA
17	Constitution and Tranquility, VA
18	Tranquility and Independence, VA
19	Independence and Sapphire, VA
20	Sapphire and Diamond, VA
21	Diamond and Windmill, VA
22	Windmill and Cambria, VA
23	Juniper and Morning Star, VA
24	Juniper and Alder, VA
25	Market and Arbor, VA

Area GID (-a aid)

Create an Area GID by combining the **–a** option with **–i**. The -a option requires an Area ID integer (aid) which is stored in the PSC for inclusion in WSAs, instead of the Intersection ID. For example, specify -a1 -i2,3 in the 10 Mile RSE to wrap it and the 11 Mile intersection into an Area GID message.

Logging (-r)

The **–r** option, if specified, enables logging to the Infrastructure DAS. On every successful GID broadcast, code in the DASMSG module formats a $750 message and the timer handler transmits it over CAN to the Infrastructure DAS.

Miscellaneous (-C, -P, -j, -t, -G, -u)

The **–C** option can be used to force the server to broadcast on the Control Channel instead of a Service Channel. If specified, WSA broadcast on the Control Channel, if present, does not include an entry for the GID service. If both –C and –S are specified, -C takes priority.

The **–P** option overrides the default GID PSID with the option argument, which can be either a hex value prefixed with "0x" or a positive decimal integer.

The **–j** option can be used to turn a Signalized intersection into a Stop Sign one. Likewise, it can be used to turn a Stop Sign intersection into a Signalized one. It toggles a Boolean so only specify it once on the command line.

The **–t** option was designed for use during Task 10 System Testing. It was also used during the Task 11 Objective Testing as conducted by U.S. DOT personnel. The option takes one argument, a 32-bit unsigned integer called the testmask. It is usually specified as a long hex value for clarity: e.g., –t 0x00000001. It can be shortened to –t1 for brevity.

The **–G** and **–u** options were useful during early development but are no longer relevant.

A.26.1.2.2 Initialize DSRC Communications Channel

The GID SPA must open the default or specified DSRC channel to broadcast its messages. If the DSRC channel is not available, the GID SPA aborts and provides a descriptive error message. In this way only one GID server may operate on a given platform at a time.

A.26.1.2.3 Format Initial Outgoing GID Message

The outgoing GID message is formatted in accordance with Appendix A.29 – 'WSM OTA Message Definitions'. The message has a TOM Header, GID Layer, and blank Metric Object if enabled with the -m option, followed by a hierarchy of GID objects comprising GID content. The message is completed with a TOM Footer. Finally, the TOM Header's CRC is updated.

A.26.1.3 Normal Operations

A timer handler is called every broadcast interval. It first checks the radio transmission queue to see if it's full. If so, it returns and the GID message is dropped.

Next, if the Dynamic Content Version feature is enabled (**-x** option) the GID message's Content Version is incremented. If **–m** was specified, the GID message's Metric Object is updated. If the message was changed (with the exception of the Metric Object payload) since last broadcast, the message's CRC is updated.

The GID message is then sent over DSRC with a RIS (Radio Interface System) API call.

Each successful GID message transmission increments an internal unsigned 16-bit message counter which is allowed to wrap around to zero naturally. The Metric Object, if enabled, contains a copy of this counter.

If the GID SPA is configured to broadcast on a DSRC service channel, it must also broadcast WAVE Service Advertisement (WSA) messages on the control channel every control channel period (10 Hz) containing a GID entry. The WSA message's role is to direct receiving OBE radios to the particular service channel containing the GID WSM.

The entry includes up to 31 bytes of PSC information provided by the GID server during initial GID message formatting. The WSA must be formatted according to Appendix A.30 – 'WSA OTA Message Definition' and contains the following data:

- TOM Framework Version
- GID Content Version
- GID Format Version
- Number of Intersections in the GID Message
- Intersection ID or Area ID
- Latitude of Center of Intersection Reference Point
- Longitude of Center of Intersection Reference Point

A.26.1.4 Input Processing Description

Giddy takes no input from external devices. It obtains time from the system clock using standard POSIX routines.

The server declares a WSM handler should it receive a GID message from a different, nearby RSE, even though it is not designed for this purpose. The handler does nothing with the message but note it in debug output.

It supports optional GID Label Objects through the use of a label file. The –L option enables the label loading from file during program initialization. The –l option enables the naming of the label file. It is a simple text file containing only content lines, comment lines and blank lines. Standard Linux line termination is advisable. Comment lines begin with '#' and end at end of line. Each content line is a comma-separated list of three fields: (1) Intersection ID, (2) Approach ID, and (3) a string of no more than 31 characters beginning with the first after the comma-delimiter and ending with the last before end of line. No more than 100 labels can be used at a time. Note that this feature is deprecated for most purposes as it increases the size of the GID. If label loading fails, the program proceeds as normal.

A.26.1.5 Output Processing Description

The program outputs DSRC messages in TOM format as its main activity. It may output debug text to stdout – controlled by **–d** and **–v** options. It may output its first almost fully prepared GID message to a local capture file before terminating (**–o** option).

The **–r** option controls logging of DSRC activity to the Infrastructure DAS used at the intersections in VA. By default this feature is disabled. If enabled, upon successful transmission of a DSRC message, the DASCreateEventMsgMyTime routine is called to format a $750 CAN message with a 5-byte timestamp followed by 3 bytes of additional information (GID LayerType & MessageCounter). The message is then sent to the DAS with a VIS API call.

If the **–o** option was specified on the command line and an output file named, the GID_Capture routine is called to write the entire GID message to the file, whereupon the server exits. This is best used to compose binary capture files with various combinations and flavors of intersections for parsing and diagnostics by external software. The raw2c tool, for example, converts such output to usable C code.

The **–y** option, when used with **–o**, causes GID_Capture to first write a fake libpcap header to the capture file before writing the GID message.

While Giddy is operating, multiple levels of debug output are supported, though only one is typically set at a time. This is done with the **–v** option. Each time it is specified on the command-line, the verbosity level is incremented. At maximum verbosity, several lines of text are output per GID message broadcast. To turn off this output completely, specify the **–d** option on the command line.

A.26.1.6 Fault Handling

If an error occurs during process initialization, the server writes error text to stdout and terminates.

During normal operation, if for some reason, the radio transmission queue fills up and the GID server is not able to broadcast a message, it does not terminate. The GID message is dropped, debug text is output, and the server continues to operate. This has never been observed.

Similarly, if the RIS API call fails to transmit the DSRC message, the server does not terminate. The GID message is dropped, debug text is output, and the server continues to operate. This has never been observed.

A.26.1.7 Shutdown

The GID server continues to operate normally until told to stop. If signaled to terminate (SIGTERM or SIGINT), it disables its timer, closes open file descriptors, de-registers DSRC services, and releases any resources before termination. The GID service on the RSE ceases when the server de-registers. If the GID had been broadcast on a service channel, the GID entry no longer appears in the CCH WSA upon deregistration.

A.26.1.8 System Management

The GID server has no external resources to be managed. Upon launch it becomes a standard Linux process with threads. Use the ps command (e.g., “ps –ef”) to view the running process.

Note that Giddy does not detach itself from the controlling tty. To run in background it is invoked with a terminating ampersand (‘&’) on the command line as follows.

```
# ./giddy –i 38 –m &
```

This command launches a GID server in the background that broadcasts the GID associated with intersection 38, plus a Metric Object. Debug output is enabled, though minimal if things are going well.

To stop the running GID server:

```
# killall giddy
```

“Giddy” Observed Performance

CPU loading by Giddy, according to ‘top’ and ‘uptime’, barely registers. As GID data does not change between broadcasts, the server spends most of its time asleep.

During engineering studies when the server includes the Metric Object in the GID message, the Metric Object’s date, time and message counter change from message to message. This causes the CRC to be updated, however, because the Metric Object is not considered content worth triggering a reparse of the GID, Content Version is not incremented. This minor activity places negligible load on the CPU.

The GID server has been observed at live intersections to perform continuously for several months, as indicated by ‘uptime’ and ‘ps.’

Enhancements

Following are some enhancements that may be considered if there is another phase to the program:

1. The default Intersection ID of 2 should be changed to 0, making the –i option mandatory.
2. The –x option (Allow Dynamic Content Version) does not update the Content Version field in the PSC. The capability was added for unit testing only and was not fully developed.

RSE Interface Definitions

A.27 RSE – DAS Interface Definition

The purpose of this table is to define the CAN interface between the intersection RSE and DAS in order to provide a consistent data logging interface between these two devices. All CAN message data follow the 'Big Endian' data format. Following the table are some general and referenced notes.

Table 181: RSE – DAS Interface Definition

CAN Message	ID	Signal	Short Name	Start Byte	Start Bit	Len	Data Type	Range	Conversion	Update Rate (ms)
RSE-DAS 1[1,2]	$750	Years since 2000	YY	0	7	4	UNM	0 – 15	E = N * 1	Event Based
		Months since January	MM	0	3	4	UNM	0 – 11	E = N * 1	
		Day of the Month	DD	1	7	5	UNM	1 – 30	E = N * 1	
		Current Hour	hh	1	2	5	UNM	0 – 23	E = N * 1	
		Current Minute	mm	2	5	6	UNM	0 – 59	E = N * 1	
		Current Second	ss	3	7	6	UNM	0 – 59	E = N * 1	
		Current Millisecond	ms	3	1	10	UNM	0 – 999	E = N * 1	
		Layer Type (LSByte)	LType	5	7	8	UNM	0 – 255	E = N * 1	
		Message Counter	MsgCtr	6	7	16	UNM	0 – 65535	E = N * 1	

General Notes:

1. The 'Signal' text color indicates the following:

   BLUE: Data elements that are supported

Referenced Notes:

1 When a GID, SPaT or GPSC message is transmitted by the RSE, CAN message $750 is sent to DAS to log the event. A similar message ($751) is logged on the OBE where the message transmission elapsed time is determined by subtracting the Metric Object's timestamp from the OBE current system time at time of reception. LType is the least significant byte of the DSRC message's Layer Type and will indicate if the transmitted message is a GPSC, GID, or SPaT one.

2 The C code below demonstrates how to obtain and convert system time in UTC to the RSE-DAS CAN message format listed above.

```
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include <stdlib.h>
unsigned char timestamp[5];
unsigned int msec;
struct timeval tRSE;
struct tm utc;
```

```
// Get high precision current time
gettimeofday(&tRSE, NULL);
msec = tRSE.tv_usec / 1000;       // cvt microsec to millisec
// Parse the seconds portion of the current time in terms of UTC
gmtime_r(&tRSE.tv_sec, &utc);
// Fill in the TIMESTAMP
timestamp[0] = (((unsigned char) (utc.tm_year + 1900 - 2000)) << 4) | (unsigned char) utc.tm_mon;
timestamp[1] = ((unsigned char) utc.tm_mday << 3) | ((unsigned char) utc.tm_hour >> 2);
timestamp[2] = ((unsigned char) utc.tm_hour << 6) | (unsigned char) utc.tm_min;
timestamp[3] = ((unsigned char) utc.tm_sec  << 2) | (unsigned char) (msec >> 8);
timestamp[4] =  (unsigned char) msec;
```

A.28 RSE – OBE DSRC OTA Interface Definition

For the contents of this interface please refer to 0.

DSRC OTA Message Definitions

A.29 WSM OTA Message Definitions

The TOM Framework

Message frameworks provide a basic set of services for message transmission. A TOM frame begins each message with a Message Header and ends it with a Message Footer. Everything in between the header and the footer is considered message content. The message content is a hierarchical set of TOM objects. The framework provides message differentiation and a basic measure of integrity.

A.29.1.1 Message Header

The TOM Header has the structure shown in Figure 79. Each field is described separately below.

Message Type	unsigned 8-bit integer
TOM Framework Version	unsigned 8-bit integer
Message Length	unsigned 16-bit integer
CRC-16	unsigned 16-bit integer

Figure 79: TOM Header

A.29.1.1.1 Message Type

The first byte of every TOM frame is the Message Type field. It is an unsigned 8-bit integer or byte.

*The Message Type for **all** TOM DSRC messages is always 0xF1 (241 decimal).*

If Message Type is not 0xF1, TOM parsing must not proceed.

Message Type does *not* indicate *type of content* held within the message frame since there may be more than one type. Multiple layers may coexist serially within a TOM frame. The layer structure is described in more detail in section 0.

A.29.1.1.2 TOM Framework Version

This field is an unsigned 8-bit integer. It is the next byte after Message Type. Applications must support at least one version.

If an application does not support a particular version, it should not parse the message any further. This is a defensive mechanism to prevent message misinterpretation. To overcome this hurdle the parser would require an update.

A.29.1.1.3 Message Length

The Message Length field is an unsigned 16-bit integer. Its bytes are in network byte order (i.e., big endian). The most significant precedes the next significant byte, and so on. Message Length represents the number of bytes in the message frame from Message Type to Message Termination Flag (see Message Footer below).

The minimal conforming frame contains 7 bytes. A message length of fewer bytes must be rejected by receiving applications.

A.29.1.1.4 CRC-16

A standard 16-bit Cyclic Redundancy Check field is included as a rough measure of data integrity. While CRCs do not guarantee integrity they do help detect bit errors.

The CRC algorithm to be used is specified in the NTCIP Guide [9] which also provides corresponding C code. It implements CRC-16 CCITT (Normal) with the generator polynomial of $x^{16} + x^{12} + x^5 + 1$.

To correctly compute the CRC-16 value for a frame the entire frame must be completely filled and ready for transmission with one exception: The CRC-16 field itself needs to be initially set to 0 and then the CRC-16 value is computed on the entire frame. The resultant CRC is then stored in the CRC-16 field. The integer stored in the CRC field must be in network byte-order.

A.29.1.2 Message Footer

Message Termination Flag	unsigned 8-bit integer

Figure 80: TOM Footer

A.29.1.2.1 Message Termination Flag

The TOM frame is completed with a Message Termination Flag (MTF) byte (Figure 80). This byte should always be set to the same value as the frame's Message Type. (It must therefore also have the same data type as Message Type.) This technique makes it easier to recognize whole TOM messages at a glance in packet streams and captures and to reject partial, malformed or corrupted frames.

No attempt is made to *escape* (i.e., substitute with a special byte sequence) any occurrences of this value within the body of the frame. The end of the frame is not determined through inspection of the contents. Message Length indicates where to expect this field. Add Message Length to the address of the Message Type byte in the Message Header to find the MTF.

A.29.1.3 Object Tag

XML has a tagging mechanism for describing content.

<name>CICAS-V</name>

The value or content of 'name' is CICAS-V. The form of Binary XML described herein has a similar tagging mechanism. Figure 81 shows the Binary XML equivalent of <name>.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer

Figure 81: Object Tag

An Object Tag is composed of two fields: Object ID and Object Size. There is a special Object Tag called the Close Tag (see Section 0). Used together, they emulate XML's tagging mechanism.

A.29.1.3.1 Object ID

An Object ID is an unsigned 8-bit byte. It always appears first in an Object Tag. See Figure 81. The purpose of an Object ID is to distinguish one type of object from other types. Values must be unique within applicable 'numeric space.'

A.29.1.3.1.1 Reserved Object IDs

Table 182: Special Object IDs

CLOSE	0
LAYER	1
METRIC	255

> *Table 182 lists all reserved Object IDs. These values must not be given to any other Object IDs, nor can they be changed or reassigned.*

The range of valid Object IDs for other, ordinary objects begins at 2 and ends at 254. These numeric values are reused across Layers to avoid running out of Object IDs.

An Object ID defined for a layer is unique to that layer and limited in scope to that layer. It may not have the same meaning outside the layer. METRIC Object ID is a special case and may be used inside any layer.

A.29.1.3.2 Object Size

Object Size is an unsigned 8-bit byte. It always appears immediately after an Object ID field in an Object Tag.

> *Object Size must be set to the size, in bytes, of the object introduced by the Object Tag. This must not include child objects or a closing object.*

Object Size is used to locate the next Object. It is added to a pointer pointing at the current Object.

A.29.1.3.3 Object Payload

When defining a new kind of object, data fields are appended tightly to the Object Tag and an unused Object ID is allocated for it. The new object now has a payload.

The Close Tag

CLOSE Object ID	unsigned 8-bit integer
Which Object to Close	unsigned 8-bit integer

Figure 82: Close Object

The Close Tag (also called the Close Object) is special, just as XML's closing tag is. It is an Object Tag structure as defined above except that the Object Size field is reinterpreted as the Target Object ID or, more commonly, Which Object to Close. This is possible because the generic Object Tag size is always two.

> *The 'Which Object to Close' field must be set to the Object ID of the object being closed.*

A.29.1.4 Nesting and Encapsulation

Key to XML and Binary XML is *nesting*. Hierarchical information can be nested to create one-to-many associations. Information can belong to other information through context. This is analogous to parent-children relationships.

The act of bracketing child objects with a parent object at the beginning and a Close Tag at the end is herein called *encapsulation*. The child objects are nested in this fashion. The parent object should not keep track of how many children it has or risk errors when counter maintenance fails.

Child objects are objects in their own right. They may encapsulate or be parent objects to other child objects as well.

A.29.1.5 Proper Closure

> *The Close Object must be used when encapsulating child objects. It must immediately follow the list or hierarchy of encapsulated child objects.*

For byte efficiency, it is acceptable to omit a matching Close Object if the object does not encapsulate child objects. Such objects are considered standalone. A GID Reference Point Object (Appendix A.29.1.10) is an example of a standalone object as is a childless SPaT Approach Object (Appendix A.29.1.21).

Objects may not recurse or encapsulate children with like Object IDs within the same layer. This allows standalone objects like childless SPaT Approach Objects to be listed one after another without requiring Close SPaT Approach tags.

The Layer Object

> *The LAYER Object is the only object besides the Close Object that is explicitly common to all TOM message layers. Even though the object is a Layer Object, each is given a modified name based on its Layer Type (e.g., GID Layer).*

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Layer Type	unsigned 16-bit integer
Layer ID	unsigned 8-bit integer
Content Version	unsigned 8-bit integer
Format Version	unsigned 8-bit integer

Figure 83: Layer Object

All Layers must be closed with a Close Layer Object to ensure integrity in case multiple layers are stored in a message frame. The Layer Types that are defined in the following sections are listed in the following table.

Table 183: Layer Object – Layer Types

GID Layer Type	1
SPaT Layer Type	2
GPSC Layer Type	3
TSVWG Layer Type	4
RCMD Layer Type	5

The Layer Format Versions table below is the authoritative source for values to use in the Layer Object's Format Version field. Servers and applications must use the appropriate one for each layer type. Applications that do not support a particular Format Version may reject the layer.

Table 184: Layer Object – Format Versions

TOM Framework Version	1
GID Format Version	2
SPAT Format Version	2
GPSC Format Version	1
TSVWG Format Version	1
RCMD Format Version	1

GID Message Layer

The GID, or Geometric Intersection Description, defines a digital map of an intersection down to the lane level if necessary. The extent of the map in each direction depends on factors such as topology, signal reception probability, and other intersections in the vicinity. Tunnels, overpasses and thick canopies may render lane data moot if the intended vehicle population doesn't have an alternate positioning solution.

The GID is designed to provide vehicles (1) a local, geo-referenced coordinate system; (2) as efficiently and accurately as possible the location of drivable lanes; (3) some means of mapping lanes to signal phase and timing information received separately; and (4) an extensible scheme capable of incorporating future content with minimal negative impact.

A.29.1.6 GID Object IDs

Table 185: GID Object IDs

INTERSECTION	2
REFERENCE POINT	3
NODE CONFIG	4
APPROACH	5
EGRESS	6
LABEL	7
REFERENCE LANE	8
NODE LIST	9
COMPUTED LANE	10
AREA	11

Consult Table 185 for the GID Object ID values that correspond to the objects discussed within the GID section. Note that only the objects that are supported in the CICAS-V SW implementation are contained within the GID section.

A.29.1.7 GID Layout

The general layout of an ordinary single-intersection GID is as follows (indentation indicates relative nesting level):

TOM Header

Layer Object: GID

| GID Intersection Object

| | GID Reference Point Object

| | GID Approach Object

| | | GID Reference Lane Object

| | | | GID Node List Object

| | | Close GID Reference Lane Object

| | | GID Computed Lane Object

| | Close GID Approach Object

| Close GID Intersection Object

Close Layer Object

TOM Footer

See section A.29.1.16.1 for a GID Layout incorporating an Area GID and multiple intersections.

A.29.1.8 GID Layer Object

The GID itself is a Layer. It encapsulates all objects concerning the geometry of one or more intersections. The GID Layer Object must be closed with the Close Layer object immediately following all of its interior objects.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Layer Type	unsigned 16-bit integer
Layer ID	unsigned 8-bit integer
Content Version	unsigned 8-bit integer
Format Version	unsigned 8-bit integer

Figure 84: GID Layer Object

The Layer Type for GID is 1.

Layer ID is used to tell layers of the same type in the same message frame apart. If there is only a single GID layer in a message, then set Layer ID to 0.

Content Version is used to indicate a change in GID content. For example, adding a lane to an intersection's GID constitutes a content change. The nature of the GID objects hasn't changed but more objects were added to the GID to reflect the new lane. Any change in the data values contained in the message increases the value. Receiving applications that notice a change in content must re-parse the contents of the GID Layer. If Content Version hasn't changed, applications are free to disregard the layer. Since intersection geometry doesn't change that often, GID Content Version will likely change only rarely.

Table 184 specifies which value to use for the GID Layer Object's Format Version field.

A.29.1.9 GID Intersection Object

The GID Intersection Object is a required object. The GID Intersection Object uniquely identifies which intersection is described in the GID. The GID Intersection Object encapsulates a hierarchy of geometric information associated with a traffic intersection, and possibly other optional objects that may also apply to the intersection. One must be used per intersection. The GID Intersection Object must be properly closed with a Close GID Intersection object. That allows the possibility of multiple intersections per GID layer.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Intersection ID	unsigned 32-bit integer
Intersection Reference Point ID (IRPID)	unsigned 8-bit integer
Intersection Attributes	8-bit bit mask

Figure 85: GID Intersection Object

Note: Do not confuse this with the SPaT Intersection Object as the two differ in format and role. One encapsulates geometry information, the other traffic signal information.

The IRPID field explicitly indicates which Reference Point is the primary one for the intersection, thus, allowing multiple reference points to be employed for very large intersections or locales.

There exists no algorithm or registration facility that manages Intersection IDs so far. Appendix A.26.1.2.1 lists the intersection IDs that were defined the the CICAS-V program.

Table 186: Intersection Attributes

0x01	Signalized Intersection
0x02	LaneLevelAccuracyRequired

Intersection Attributes is used to indicate certain things about the entire intersection as a whole. Table 186 specifies the bit field format. To indicate an intersection is equipped with a traffic signal controller, set the Signalized Intersection bit ON. Conversely, to indicate it is a stop sign intersection, the Signalized Intersection bit must be OFF.

The Lane Level Accuracy Required attribute indicates if the GPS rover must lane match using locally corrected positioning to support CICAS and other high accuracy positioning dependent systems.

A.29.1.10 GID Reference Point Object

This object is used to indicate the location of a GPS reference point. This object is a standalone. It has no children so it doesn't need a close tag.

There may be more than one Reference Point Object in a GID, as when large areas are covered. Each reference point must have different Reference Point IDs. There must be at least one per Intersection.

The Reference Point with RPID 0 must lie within the closed polygon formed by connecting the stop nodes and ideally be the center point of the intersection.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Reference Point ID	unsigned 8-bit integer
Latitude (LSBit = 10^{-7} decimal degrees)	signed 32-bit integer
Longitude (LSBit = 10^{-7} decimal degrees)	signed 32-bit integer
Altitude (LSBit = 1 dm)	*partially signed* 16-bit integer

Figure 86: GID Reference Point Object

Latitude and Longitude are determined by obtaining the location's GPS lat/longs in decimal degrees to 7 or more digits of precision. The integer value is derived by multiplying each value by 10^7 and removing the mantissa.

Table 187: Altitude Mapping Table

Altitude	Rolled value
64,000 dm	64000
1 dm	1
0 dm	0
-1 dm	65535
-2 dm	65534
-1,535 dm	64001

Altitude is treated specially. Altitude is a *partially signed* 16-bit integer indicating decimeters above/below the WGS-84 geoid. Positive values up to 64,000 are used "as is." Negative values as low as -1,535 are "rolled over the top" by adding them to 65,536 (2^16). Max/min altitudes in feet: 6400m = 20997ft, -153.5m = -504ft.

A.29.1.11 GID Node Config Object

A Node Config Object gives the GID Designer greater freedom to change the definition of a Node (see Section A.29.1.14 for more on nodes). While it is optional, it is the only way to enable inclusion of altitudes in nodes. This object stands alone; it requires no closing object tag.

<table>
<tr><td colspan="4">Object ID</td><td>unsigned 8-bit integer</td></tr>
<tr><td colspan="4">Object Size</td><td>unsigned 8-bit integer</td></tr>
<tr><td>Z</td><td>W</td><td>C</td><td>Node Offset Granularity</td><td>bit fields spanning 1 byte</td></tr>
</table>

Figure 87: GID Node Config Object

A Node Offset is a Cartesian coordinate offset (X, Y, or Z) with respect to a Reference Point. X increases to the East, Y to the North, and Z as altitude increases.

Normally, Nodes consist of just X and Y offsets. The *Z* bit, if on, allows Nodes to have Z offsets. It informs Node List parsers to expect Z values in addition to X and Y values. The Z bit is not currently supported in the CICAS-V OBE SW implementation.

The *W* bit, if on, allows Nodes to have Lane Widths. It informs Node List parsers to expect W values in addition to X and Y and possibly Z values.

Each offset (X, Y or Z) is a signed 16-bit integer by default. The *C* bit, if on, changes it to 12-bits. Offsets are packed together; no pad bytes. The C bit is not currently supported in the CICAS-V OBE SW implementation.

See section A.29.1.14.2 for more detail about how nodes and node offsets are organized.

Use Node Offset Granularity (NOG), a 5-bit bit field, to override default 1 cm granularity of node offsets. Increase NOG to extend the range of offsets. A NOG of 0 means default, 31 is max.

Table 188, column 2, shows how the C bit and NOG interact to reduce maximum resolvable node offset distances.

Table 188: Max Offset Values at Various Granularities

16 bit offset	12 bit offset
max offset @ NOG	max offset @ NOG
327.67m @ 1cm	20.47m @ 1cm
1,638.35m @ 5cm	102.35m @ 5cm
3,276.70m @ 10cm	409.40m @ 20cm
4,915.05m @ 15cm	614.10m @ 30cm
6,553.40m @ 20cm	
9,830.10m @ 30cm	

The scope is bounded by the parent. This means that a Node Config can affect just a reference lane, an entire approach, or a complete intersection depending on its placement in the GID. If it is placed after the GID Layer Object and before a pair of intersections, it can affect both intersections. The same is true if encapsulated by a GID Area Object.

A.29.1.12 GID Approach Object

The Approach Object encapsulates Reference Lanes and Computed Lanes, if any. Its parent is the Intersection Object. This is a required object.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Approach ID	unsigned 8-bit integer

Figure 88: GID Approach Object

The role of this object is to facilitate the association of certain lanes coming from a given direction with the dedicated signal phase and timing data for those lanes. For example, if a left or right turn (which may span more than one lane) has separate, dedicated signal phase and timing, it is given its own approach.

Use one GID Approach Object per approach. After the approach's lane objects, complete the approach with a Close Approach object.

A.29.1.13 GID Reference Lane Object

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Lane Number	unsigned 8-bit integer
Reference Point ID	unsigned 8-bit integer
Lane Width	unsigned 16-bit integer
Lane Attributes	16-bit bit mask

Figure 89: GID Reference Lane Object

The Reference Lane Object provides important information about individual lanes of traffic. Only one per lane is permitted. Reference lanes in a message are used when actual X,Y node data exists. This node data is placed in a Node List Object and encapsulated. The Reference Lane Object must be closed with a Close Reference Lane object.

A.29.1.13.1 Lane Number

The Lane Number field indicates which lane is which. Lanes are numbered per-approach starting with 1. Beginning on the left, they increase to the right from the driver's perspective. Note that this is somewhat by convention.

Adjacency of lanes with sequential lane numbers is not guaranteed, meaning sometimes road geography calls for the interposition of a non-lane between drivable lanes.

A.29.1.13.2 Reference Point ID

The Reference Point ID (RPID) indicates which Reference Point was used to create the offsets found in the encapsulated GID Node List Object. Rather than make an assumption, the association is made explicit to allow for future use of multiple Reference Points within a GID and distant node lists based on them, perhaps used to encode nearby stop sign intersections.

A.29.1.13.3 Lane Width

Lane Width describes the mean physical width of the lane for the length of the lane. This assumes the lane is of fairly consistent width. Gutters, shoulders, margins segregated by paint, and marked, roadside parking spaces should not be considered part of the lane because these are not drivable areas of pavement controlled by traffic signals. Granularity is 1 centimeter.

A.29.1.13.4 Variable Lane Width

The Lane Width in the GID Reference Lane Object sets the default width for the lane. When an in-scope Node Config Object is employed to enable lane widths (W) in Node List Objects, any non-zero lane widths at the node level override this default.

The following XML example shows how to specify variable lane widths per-node. (1) The W-bit is turned on in a Node Config Object somewhere above and in a parent, grandparent or great-grandparent of the Reference Lane Object. (2) Each node in the Node List Object has a third value: either a zero, which means the lane width at that node is the Lane Width set in the Reference Lane Object (i.e., the default), or some positive integer value to use instead. (3) An empty Node Config Object resets node interpretation to default for the remainder of the scope. This step is optional if the scope ends naturally.

```
            <NodeConfig>
(1)             <W/>
            </NodeConfig>
            …
            <ReferenceLane>
                <Description>This is Lane A03</Description>
                <LaneNumber>3</LaneNumber>
                <LaneWidth>339</LaneWidth>
                <LaneAttributes>
                    <StraightAllowed/> <RightAllowed/>
                </LaneAttributes>
                <RPID>0</RPID>
                <NodeList>
(2)                 2107,-3547,450
                    10288,-10964,0
                    12732,-13208,0
                </NodeList>
            </ReferenceLane>
            …
```

(3) <NodeConfig>

</NodeConfig>

See section A.29.1.14 below for the ramifications in the Binary XML.

A.29.1.13.5 Lane Attributes

Table 189: GID Lane Attributes

0x0001	Straight Allowed
0x0002	Left Allowed
0x0004	Right Allowed
0x0008	U-Turn Allowed
0x0010	No U-Turn
0x0020	No Turn on Red
0x0040	No Stop
0x0100	Yield
0x1000	HOV Lane
0x2000	Two Way Left Turn Lane (TWLTL)
0x4000	Bike Lane

Lane Attributes is a combination of characteristics specific to a lane. Together they indicate what traffic may and/or may not do while in the lane and indicate what type of lane it is.

Lane attribute bits are Exclusive-ORed together to produce the Lane Attributes bit mask value. Allowed and disallowed vehicle movements for the lane are specified in this manner. For example, if a lane has both Left Allowed and Straight Allowed bits enabled, vehicles in the lane may turn left or go straight-through the intersection when the applicable signal phase permits. That lookup is performed in the most current SPaT received, if still valid. If, however, that SPaT indicates both Red Left Arrow and Green Ball are lit then vehicles in the lane are not permitted to turn left and may only proceed straight.

To achieve Right Turn on Red (RTOR), specify Right Allowed but not No Turn on Red. U-Turn Allowed and No U-Turn are mutually exclusive. Only Left Allowed and Right Allowed lanes may specify U-Turn attributes. No Turn on Red may apply to Left Allowed, Right Allowed and U-Turn Allowed lanes only. The Yield attribute is for lanes which peel off from normal approach lanes that are not signalized, and yet come in conflict with intersection traffic.

The GID Lane Attributes are currently not supported in the CICAS-V OBE SW implementation.

A.29.1.14 GID Node List Object

The Node List Object provides node data. A *node* is a spot on the ground. It has an X and a Y coordinate indicating its position on a local, geographic North-oriented, arbitrary grid. These are typically X and Y offsets from some reference point, but they may include Z if the closest in-scope Node Config Object has its Z bit enabled. Z is an offset based on the Reference Point Altitude. There is currently no support for the Z offset in the CICAS-V OBE SW implementation.

This is a required object when using a GID Reference Lane Object, which encapsulates it. It is a standalone object and does not have a close object.

Object Size varies based on number of nodes in the payload, size of each node, and it must also account for the two-byte Object Tag. A GID Node List Object containing two default-sized X,Y nodes, for example, has an Object Size of 10 bytes. One with three such nodes is 14 bytes, etc.

A.29.1.14.1 Node Placement

The first node must be on the Stop Line if the Node List Object belongs to an Approach Object, or where an egress lane begins if the Node List Object belongs to an Egress Object. Each node that follows is increasingly distant from the intersection, following the centerline of the lane. Append subsequent nodes to the object as needed.

A.29.1.14.2 Node List Organizations

A.29.1.14.2.1 Basic Node Lists

There are four basic node list organizations. The presence of Z and W fields (see section A.29.1.13.4 above) at this level is controlled by the closest Node Config Object in-scope above the Node List Object. The tuples (i.e., ordered sets) are listed below with references to corresponding figures.

1. X,Y (default) – see Figure 90
2. X,Y,Z – Not Supported in the CICAS-V OBE SW implementation
3. X,Y,W – see Figure 91
4. X,Y,Z,W – Not Supported in the CICAS-V OBE SW implementation

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
X Offset of Stop/Start Line Node (LSBit = 1 cm)	signed 16-bit integer
Y Offset of Stop/Start Line Node (LSBit = 1 cm)	signed 16-bit integer
X Offset of Subsequent Node (LSBit = 1 cm)	signed 16-bit integer
Y Offset of Subsequent Node (LSBit = 1 cm)	signed 16-bit integer

Figure 90: Default GID Node List Object (X,Y)

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
X Offset of Stop/Start Line Node (LSBit = 1 cm)	signed 16-bit integer

Y Offset of Stop/Start Line Node (LSBit = 1 cm)	signed 16-bit integer
Lane Width (W) of Stop/Start Node (LSBit = 1 cm)	unsigned 16-bit integer
X Offset of Subsequent Node (LSBit = 1 cm)	signed 16-bit integer
Y Offset of Subsequent Node (LSBit = 1 cm)	signed 16-bit integer
Lane Width (W) of Subsequent Node (LSBit = 1 cm)	unsigned 16-bit integer

Figure 91: GID Node List Object with Optional Node Level Lane Widths (X,Y,W)

A.29.1.15 GID Computed Lane Object

The Computed Lane Object provides the opportunity to save some bytes in certain circumstances. It never contains any node data. It's a standalone object. Since it has no children, it requires no Close Tag. This is an optional object.

A Computed Lane may only be used in a GID where a Reference Lane parallels it accurately to the distance desired. They must both be in the same Approach. Reference Lane Number is the lane number of the Reference Lane whose node data is used to determine the location of the computed lane's nodes. Center Line Offset indicates the lateral offset of the computed lane's center line vs. the reference lane's lane segment. This is a signed integer; negative is a leftward offset, positive is a rightward offset. Orientation is the driver's perspective.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Lane Number	unsigned 8-bit integer
Lane Width	unsigned 16-bit integer
Lane Attributes	16-bit bit mask
Reference Lane Number	unsigned 8-bit integer
Center Line Offset (LSBit = 1 cm)	signed 16-bit integer

Figure 92: GID Computed Lane Object

Lane Number and Lane Attributes are as defined above in Section A.29.1.13. As is Lane Width except that it may be zero to force Lane Width inheritance from the Computed Lane's Reference Lane lane width.

A.29.1.16 GID Area Object

The GID Area Object uniquely identifies a collection of intersections. It does this by encapsulating one or more GID Intersection Objects and their child objects. If included, the GID Area Object must be properly closed with a Close GID Area object.

The intent of this object is to provide a way to identify a number of intersections with one ID. That ID is then used in place of what would otherwise have been a variable length list of Intersection IDs. The primary use case is in the WAVE Service Announcement (WSA) PSC payload, which is limited in size.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Area ID	unsigned 32-bit integer

Figure 93: GID Area Object

As shown in Figure 93, the object begins with the familiar Object Tag. The object's payload consists of the Area ID field. The size of the Area ID field and its range of values must be large enough to span half the range of possible Intersection ID values, for that is the maximum-use case scenario: one Area ID for every two Intersection IDs.

A.29.1.16.1 GID Layout Incorporating Area GID

The Area GID assumes that one or more intersections are contained within it. Its object layout is shown below:

TOM Header

Layer Object: GID

GID Area Object

| GID Intersection Object

| | GID Reference Point Object

| | GID Approach Object

| | | GID Reference Lane Object

| | | | GID Node List Object

| | | Close GID Reference Lane Object

| | | GID Computed Lane Object

| | Close GID Approach Object

| Close GID Intersection Object

| *GID Intersection Object*

...

| *Close GID Intersection Object*

Close GID Area Object

Close Layer Object

TOM Footer

SPaT Message Layer

The SPaT, or Signal Phase and Timing message layer, is designed to provide traffic signal phase and timing information organized in such a way that a vehicle can reliably determine (1) whether it has right-of-way or must stop; (2) what movements are allowed from a given lane; and (3) an extensible scheme capable of incorporating future content with minimal negative impact.

A.29.1.17 SPaT Object IDs

Table 190: SPaT Object IDs

INTERSECTION	2
APPROACH	3
PEDESTRIAN	4
PREEMPT	5
LOCATION	6
LABEL	7
SENSOR	8
CURRENTTIME	9

Consult Table 190 for the SPaT Object ID values that correspond to the objects discussed within the SPaT section. Note that only the objects that are supported in the CICAS-V SW implementation are contained withing the SPaT section.

A.29.1.18 SPaT Layout

Typically, the general layout of a SPaT is organized as follows. Indentation indicates relative nesting levels. Only one object repeats: SPaT Approach Object. An intersection with 8 approaches has 8 SPaT Approach Objects in it. The relationship is one approach object for each approach. SPaT Approach Objects in this scenario do not need matching Close SPaT Approach Objects because they do not encapsulate other objects (i.e., they are standalone). Standalone objects do not require closure. See section A.29.1.5 for more about standalone objects.

The layout of required objects in a SPaT Layer is as shown below. Indentation implies encapsulation.

```
TOM Header
        Layer Object: SPaT
        |       SPaT Intersection Object
        |       |       SPaT Approach Objects
        |       Close SPaT Intersection Object
        Close Layer Object
TOM Footer
```

Optional SPaT objects may appear elsewhere in the message layer at the points indicated below.

TOM Header

Layer Object: SPaT

| *Optional SPaT Object*

| SPaT Intersection Object

| | *Optional SPaT Object*

| | SPaT Approach Object

| | | *Optional SPaT Object*

| | Close SPaT Approach Object

| Close SPaT Intersection Object

Close Layer Object

TOM Footer

Standalone and encapsulating Approach Objects may be mixed as needed. Note the addition of the Close SPaT Approach Object in this scenario. This is now necessary because the SPaT Approach Object is no longer standalone. It encapsulates something.

A.29.1.19 SPaT Layer Object

The SPaT Layer encapsulates one or more SPaT Intersections. The SPaT Layer Object must be closed with the Close Layer object immediately following all of its interior objects.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Layer Type	unsigned 16-bit integer
Layer ID	unsigned 8-bit integer
Content Version	unsigned 8-bit integer
Format Version	unsigned 8-bit integer

Figure 94: SPaT Layer Object

The Layer Type for SPaT is 2.

Content Version is used to indicate a change in SPaT content. For example, the change of a Countdown Timer is a content change. The Content Version for that SPaT layer must differ from prior broadcast SPaT message layers for the intersection. Receiving applications that notice a change in content version must parse the contents of the SPaT Layer. If Content Version hasn't changed, the application is free to disregard the layer, using cached data. Assuming SPaT Content Version changes on the order of ten times per second, wrapping around will occur about once every 25.5 seconds.

Layer ID is used to tell layers of the same type in the same message frame apart. If there is only a single SPaT layer in a message then the Layer ID is set to 0.

Table 184 specifies which value to use for the SPaT Layer Object's Format Version field.

A.29.1.20 SPaT Intersection Object

This is a required object. The SPaT Intersection Object uniquely identifies the intersection it corresponds to. It encapsulates all SPaT Approach Objects at the intersection, and possibly other optional objects that may apply to the entire intersection. Encapsulated SPaT Approach Objects may occur in any order. The SPaT Intersection Object must be closed with the Close Intersection object. It is possible to have multiple intersections per SPaT layer but this is not recommended.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Intersection ID	unsigned 32-bit integer

Figure 95: SPaT Intersection Object

A.29.1.21 SPaT Approach Object

The SPaT Approach Object provides signal phase and timing for an individual approach. Its parent is the SPaT Intersection Object. This is a required object.

> *The role of this object is to convey all relevant signal phase and timing information needed by drivers in controlled lanes. There must be one Approach Object for every unique signal phase from a given direction. An Approach ID is provided to allow vehicles to match geometry data in the GID to signal phase data. This ID must match its counterpart in the GID Approach Object.*

The SPaT Approach Object may encapsulate other objects that apply to a given approach, except other Approach objects. When encapsulation occurs there must be a Close SPaT Approach object. Otherwise, the Close SPaT Approach object should be omitted for byte efficiency as the object is intended to be standalone under normal circumstances.

<table>
<tr><td colspan="2">Object ID</td><td>unsigned 8-bit integer</td></tr>
<tr><td colspan="2">Object Size</td><td>unsigned 8-bit integer</td></tr>
<tr><td colspan="2">Approach ID</td><td>unsigned 8-bit integer</td></tr>
<tr><td colspan="2">Signal Phase Indications</td><td>32-bit bit mask</td></tr>
<tr><td>Countdown Timer Confidence</td><td>Yellow Duration Confidence</td><td>2 x unsigned 4-bit integer (two nibbles)</td></tr>
<tr><td colspan="2">Time until next signal phase change
(in hundredths of a second)
AKA Countdown Timer</td><td>unsigned 16-bit integer</td></tr>
<tr><td colspan="2">Yellow Duration</td><td>unsigned 8-bit integer</td></tr>
</table>

Figure 96: SPaT Approach Object

A.29.1.21.1 Signal Phase Indications

The Signal Phase Indications field is a bit mask to facilitate combinations of various indications or signal lights. 'OR' the bits together before storing the value in the field (in network order).

Table 191: Signal Indication Bit Values

All 0	Dark
0x00000001	Green Ball
0x00000002	Yellow Ball
0x00000004	Red Ball
0x00000010	Green Left Arrow
0x00000020	Yellow Left Arrow
0x00000040	Red Left Arrow
0x00000100	Green Right Arrow
0x00000200	Yellow Right Arrow
0x00000400	Red Right Arrow
0x00001000	Soft Green Left Arrow
0x00002000	Soft Yellow Left Arrow
0x00004000	Soft Red Left Arrow
0x00010000	Soft Green Right Arrow
0x00020000	Soft Yellow Right Arrow
0x00040000	Soft Red Right Arrow
0x00100000	Straight Green Arrow
0x00200000	Straight Yellow Arrow
0x00400000	Straight Red Arrow
0x01000000	Flashing Ball
0x02000000	Flashing Left Arrow
0x04000000	Flashing Right Arrow
0x08000000	Flashing Soft Left Arrow
0x10000000	Flashing Soft Right Arrow
0x20000000	Flashing Straight Arrow

The Flashing bits are modifiers. To indicate a Flashing Red Ball, for example, the Red Ball and Flashing Ball bits must be on. Soft arrows are oblique or angled arrow indications.

Only the Green Ball, Yellow Ball, Red Ball, and Flashing Red Ball are supported by the CICAS-V OBE SW implementation.

A.29.1.21.2 SPaT Confidences

Countdown Timer Confidence (high) and Yellow Duration Confidence (low) are 4-bit fields within the same byte. These fields are associated with like-named fields described below. Each is set independently to one of the values in Table 192. Any other value is undefined and should be treated as 'disregard.'

A receiving application should check a field's Confidence before consulting the associated field.

Table 192: Confidence

Value	Confidence
0	Disregard associated field
1	Associated field is inexact (at least)
2	Associated field is inexact (at most)
3	Associated field is exact

Note: The CICAS-V OBE SW implementation ignores the Countdown Timer Confidence and Yellow Duration Confidence values and considers the Countdown Timer value to be exact.

A.29.1.21.3 SPaT Timers

Countdown Timer indicates either exactly how much time is left until the signal phase changes, at least that much time, or at most that much time. If, however, Countdown Timer Confidence is zero, the meaning of the Countdown Timer value is undefined. Countdown Timer specifies time remaining in hundredths of a second.

Yellow Duration specifies how long a yellow light lasts on that approach, in tenths of a second. This is typically a static value in the traffic signal. Each phase or overlap may have a different value from its counterparts.

A.29.1.22 SPaT Current Time Object

The SPaT Current Time is an optional object. It is encapsulated by a SPaT Intersection to convey the current RSE system time at that intersection.

This is a standalone, optional object. No close tag required.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Year	unsigned 16-bit integer
Month	unsigned 8-bit integer
Day	unsigned 8-bit integer
Hour	unsigned 8-bit integer
Minute	unsigned 8-bit integer
Millisecond	unsigned 16-bit integer

Figure 97: SPaT Current Time Object

The intent of this object is that recipients may add this time to any in-scope future time offset (e.g., Countdown Timer) and, assuming both producer and consumer sync their system clocks to the same reference, can determine, for example, when a signal light will change.

Time values reflect UTC.

Valid value ranges:

Year	1..9999
Month	1..12
Day	1..31
Hour	0..23
Minute	0..59
Millisecond	0..60999 (including 60000..60999 for 1 leap second)

The value of 0 is undefined for the Year, Month and Day fields. Any date/time field within the object that is set to an undefined value renders the timestamp invalid.

GPSC Message Layer

The GPSC message layer format is largely described below but the inner workings of GPS Correction, the RTCM data format, etc., are described in Appendix A.25 – 'GPS Corrections Service Provider Application Details.'

A.29.1.23 GPSC Object IDs

Table 193: GPSC Object IDs

RTCM_3.0_L1_CORRECTION	2

Table 193 lists Object ID values to use within the GPSC message layer.

A.29.1.24 GPSC Layer Object

The GPSC Layer encapsulates one or more GPSC correction objects.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Layer Type	unsigned 16-bit integer
Layer ID	unsigned 8-bit integer
Content Version	unsigned 8-bit integer
Format Version	unsigned 8-bit integer

Figure 98: GPSC Layer Object

The Layer Type for GPSC is 3.

It is not unusual for RTCM correction data to change on a regular basis. The fact that the correction data has changed is communicated through Content Version. This field is incremented on change from previously published content. The new value doesn't matter to receiving applications so much as the fact that it differs from the previous value seen. This allows the unsigned field to wrap around naturally.

Layer ID is used to tell layers of the same type in the same message frame apart. If there is only a single GPSC layer in a message then set Layer ID to 0.

Table 184 specifies which value to use for the GPSC Layer Object's Format Version field.

A.29.1.25 GPSC RTCM 3.0 L1 Correction Object

This object conveys GPS time and RTCM version 3.0 L1 correction data.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
GPS Status	16-bit bit mask
GPS Week Number	unsigned 16-bit integer
GPS Milliseconds in Week	unsigned 32-bit integer
Length of both RTCM 1005 and RTCM 1001 data	unsigned 8-bit integer
RTCM 1005 Message Array	25 bytes of data
RTCM 1001 Message Array	up to 101 bytes of data

Figure 99: GPSC RTCM 3.0 L1 Correction Object

All integer fields have their bytes in network order. Object ID is set to the RTCM_3.0_L1_CORRECTION Object ID value from Table 193. Object Size varies with the size of the RTCM data. It must be set to the size in bytes of the entire object.

Table 194: GPSC GPS Status

0x0001	Unhealthy
0x0002	Unmonitored
0x0004	*(reserved)*
0x0008	PDOP > 5
0x0010	Satellites < 5
0x0020	Local GPS Corrections
0x0040	Network GPS Corrections
0x0080	Other GPS Corrections

GPS Status is a 16-bit bit mask composed of bit values from Table 194 and stored in network order. OR them together to obtain the field value.

GPS Week Number and GPS Milliseconds in Week provide the current GPS time.

The two RTCM message array fields provide the core data of the correction object. The sum of their sizes must be stored in the 'Length of both RTCM 1005 and RTCM 1001 data' field.

TSVWG Message Layer

The TSVWG Message Layer alerts the infrastructure that a Traffic Signal Violation Warning alert has been given to a vehicle's driver. Message flow is OBE to RSE.

A.29.1.26 TSVWG Object IDs

Table 195: TSVWG Object IDs

Warning Given	2

Table 195 lists Object ID values to use within the TSVWG message layer.

A.29.1.27 TSVWG Layer Object

The TSVWG Layer encapsulates one or more TSVWG Warning Given objects.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Layer Type	unsigned 16-bit integer
Layer ID	unsigned 8-bit integer
Content Version	unsigned 8-bit integer
Format Version	unsigned 8-bit integer

Figure 100: TSVWG Layer Object

The Layer Type for TSVWG is 4.

Layer ID is used to tell layers of the same type in the same message frame apart. If there is only a single TSVWG layer in a message then Layer ID is set to 0.

Table 184 specifies which value to use for the TSVWG Layer Object's Format Version field.

TSVWG Warning Given Object

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Intersection ID	unsigned 32-bit integer
Approach ID0	unsigned 8-bit integer
Approach ID1	unsigned 8-bit integer
Approach ID2	unsigned 8-bit integer

Figure 101: TSVWG Warning Given Object

Intersection ID must be set to the Intersection ID (from GID or SPaT) of the applicable intersection.

If it is highly likely that a vehicle is on a specific approach, its Approach ID is stored in Approach ID0, and Approach ID1 and ID2 are both set to zero. However, if the vehicle may be on more than one approach, up to two additional Approach IDs (i.e., ID1 and ID2) may be used. The values must be ordered most likely to least likely.

RCMD Message Layer

THIS MESSAGE LAYER IS ONLY FOR TESTING. This is a tentative message for research purposes only.

The RCMD Message Layer provides a means for remote command of road-side equipment.

This message is not currently designed with any further functionality in mind.

A.29.1.28 RCMD Object IDs

Table 196: RCMD Object IDs

Preempt Signal	2

Table 196 lists Object ID values to use within the RCMD message layer.

A.29.1.29 RCMD Layer Object

The RCMD Layer encapsulates one or more RCMD Objects. The RCMD Layer Object must be closed with the Close Layer object immediately following all of its interior objects.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Layer Type	unsigned 16-bit integer
Layer ID	unsigned 8-bit integer
Content Version	unsigned 8-bit integer
Format Version	unsigned 8-bit integer

Figure 102: RCMD Layer Object

The Layer Type for RCMD is 5.

Layer ID is used to tell layers of the same type in the same message frame apart. If there is only a single RCMD layer in a message then Layer ID is set to 0.

Table 184 specifies which value to use for the RCMD Layer Object's Format Version field.

A.29.1.30 RCMD Preempt Signal Object

The RCMD Preempt Signal Object is used to command the traffic signal controller associated with Intersection ID to perform signal preemption. The command affects all vehicle signaling at the intersection.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Intersection ID	unsigned 32-bit integer
Approach ID	unsigned 8-bit integer
Preemption Type	unsigned 8-bit integer

Figure 103: RCMD Preempt Signal Object

Since the command may be given within broadcast range of more than one intersection, the Intersection ID of the intended intersection must be filled in with the correct value.

The preempting vehicle sets the Approach ID to the Approach ID of the approach it is driving on at the time.

Table 197 defines Preemption Type values to use.

Table 197: Preemption Types

Clear Preemption	0
Turn Green	1
Turn Red	2

If the light is red and Turn Green is asserted, the light will turn green as soon as all other conflicting approaches have a red.

If the light is green and Turn Red is asserted, the light will immediately begin the proper sequence to red.

Multiple intersections (e.g., back to back signalized intersections) are handled by including in the same message layer as many RCMD Preempt Signal Objects as are needed.

Metric Object

The Metric Object was created for engineering purposes. It is mainly used to provide basic metrics for downstream monitoring applications to track. By monitoring these objects an application may detect message sequence gaps and know when they occurred.

The Metric Object is an optional object. It is a standalone object. No close tag is required.

This object may be used inside any Layer. It has a special Object ID to keep it distinct from other layer objects. Only the Metric Object ID may have the value of 255.

Object ID	unsigned 8-bit integer
Object Size	unsigned 8-bit integer
Year	unsigned 16-bit integer
Month	unsigned 8-bit integer
Day	unsigned 8-bit integer
Hour	unsigned 8-bit integer
Minute	unsigned 8-bit integer
Millisecond	unsigned 16-bit integer
Message Counter	unsigned 16-bit integer

Figure 104: Metric Object

As a meta-object, this object shall not be used as the basis for a future time in a production object.

Time values reflect UTC.

Valid value ranges:

Year	1..9999
Month	1..12
Day	1..31
Hour	0..23

Minute 0..59

Millisecond 0..60999 (including 60000..60999 for 1 leap second)

The value of 0 is undefined for the Year, Month and Day fields. Any date/time field within the object that is set to an undefined value renders the timestamp invalid.

Message Counter is a sequence counter. It is incremented with every transmission. After max value it wraps around to 0. This counter is useful for packet drop detection when conducting messages traffic studies.

A.30 WSA OTA Message Definition

A service registers itself (initiate a WBSS) on the WAVE network as defined in the IEEE Standard 1609.3. As part of the initiation a *ProviderServiceIdentifer* (PSID) and a *ProviderServiceContext* (PSC) is configured and sent for each specific service. The PSID uniquely identifies the service and the PSC provides context information about the service so a user can determine if the service is relevant to them. The PSID and the PSC is contained in the WAVE Service Advertisement (WSA) in the *ProviderServiceInfo* section. The WSA also contains the information about which channel to switch to in the service channel interval. The current definition of the WSA does not include information about which service channel interval will contain the WAVE Short Message for a desired service. However, since the user registers and joins a WBSS it will follow the CCH and SCH channel switching intervals until it receives the WSM from the provider.

For the CICAS-V the GID and GPSC services utilized the SCH. The SPaT was broadcast on the CCH. The PSID assignments for each service are documented in Appendix A.31. The following describes the PSC for each of the CICAS-V services utilizing the SCH and how the data values should be used.

GID Service PSC Definition

Table 198: PSC for GID Service

Value Name	Description	Data Type
Field Length	The number of bytes contained in the PSC excluding this byte	Unsigned 8-bit Integer
TOM Framework Version	The version of the TOM Framework as described in the TOM Header	Unsigned 8-bit Integer
GID Content Version	The version of the content in the GID as described in the GID Layer Object	Unsigned 8-bit Integer
GID Format Version	The version of the format for the GID as described in the GID Layer Object	Unsigned 8-bin Integer
Number of Intersections	The total number of intersections described in this message. This will determine whether or not this message describes a collection of intersections which is also referred to as an Area GID.	Unsigned 8 bit Integer

Value Name	Description	Data Type
Intersection ID / Area ID	If the number of intersections equals 1 then this value is an Intersection ID. If the number of intersections greater than 1 this value is an Area ID. For an Intersection ID this is the same as described in the GID Intersection Object. For an Area ID this is an ID that references the intersections contained within this message and is described in the Area Object.	Unsigned 32-bit Integer
Reference Point Latitude	This reference point refers to the intersection where this message is being broadcast. This value is the latitude of the reference point for the center of the intersection (RPID = 0) as described in the GID Reference Point Object.	Signed 32-bit Integer
Reference Point Longitude	This reference point refers to the intersection where this message is being broadcast. This value is the longitude of the reference point for the center of the intersection (RPID = 0) as described in the GID Reference Point Objects.	Signed 32-bit Integer

Data Value Usage:

- Field Length: Used to assist in parsing the message into the individual value fields
- TOM Framework Version: Used to determine if the parser is programmed to read the TOM format that the GID is contained within
- GID Content Version: Used to determine if the GID has been updated since the last receipt of the GID for the specific intersection. If this is a new GID (updated or not in the database) the OBE shall switch to the appropriate service channel (SCH) to receive the new GID
- GID Format Version: Used to determine if the parser is programmed to read the format of the GID, and thus parse the GID
- Number of Intersections: Used to determine if this message is for one intersection or a collection of intersections
- Intersection ID (or Area ID): Used to determine if the OBE already has the GID or set of GIDs in its database
- Reference Point Latitude and Longitude: Used in cases where the OBE is within range of more than one RSE advertising the CICAS-V GID Service and the GID database does not contain the GIDs being sent out. The OBE shall determine, based on the most relevant approaching intersection, which service channel to switch to in order to receive the necessary GID

GPSC Service PSC Definition

Table 199: PSC for GPS Corrections Service

Value Name	Description	Data Type
Field Length	The number of bytes contained in the PSC excluding this byte	Unsigned 8-bit Integer
TOM Framework Version	The version of the TOM Framework as described in the TOM Header	Unsigned 8-bit Integer
GPSC Format Version	The version of the format for the GPSC message as described in the GPSC Layer Object	Unsigned 8-bit Integer
Intersection ID	ID of the intersection in which the GPSC message is being sent from	Unsigned 32-bit Integer
GPS Status	A status bit mask containing a bit that determines the health of the GPS receiver as well as other status information. This is the same field as described in the GPSC RTCM 3.0 L1 Correction Object	16-bit bit mask

Data Value Usage:

- Field Length: Used to assist in parsing the message into the individual value fields.
- TOM Framework Version: Used to determine if the parser is programmed to read the TOM format that the GPSC message is contained within.
- GPSC Format Version: Used to determine if the parser is programmed to read the format of the GPSC message and, thus, parse the message.
- Intersection ID: Used in cases where an OBE receives GPS Corrections from more than one RSE. The Intersection ID shall be used in these cases to determine which RSE a vehicle is heading towards and thus is more relevant to receive this message from.
- GPS Status: Used to determine the health of the GPS receiver. If the GPS Status bit mask states that the GPS receiver is both Healthy and Monitored the OBE shall switch to the appropriate SCH for receiving GPS Correction data.

A.31 PSID Definitions

The following table lists the PSIDs that were defined for each of the OTA DSRC WSMs defined for the CICAS-V program. Note that in the event of their being overlap with another competing service each of these values is configurable in both the RSE and OBE.

Table 200: CICAS-V PSID Assignments

CICAS-V Message	VII Service Categorization			PSID (Hex Values)			
	Service Category	**Service Sub Category**	**Application Type**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**
SPaT	Safety	CICAS-V	SPaT	01	E0	00	01
GID	Safety	CICAS-V	GID	01	E0	00	02
GPSC	Safety	CICAS-V	Local Corrections	01	E0	00	03
TSVWG	Probe Data	CICAS-V	Traffic Safety Violation	03	E0	00	01
RCMD	System Test	CICAS-V	Remote Command	07	E0	00	01

System Test Case Details

The purpose of the Task 10 system testing effort was to verify that all of the system components were operational, integrated, and working properly through a combination of component level and intersection level system testing efforts. The execution of these tests was seen as a precursor to the execution of the Task 11 Objective Test procedures. However, there was some overlap between this testing effort and the Task 11 testing effort.

Test procedures were created to test not only the OBE as a whole but also the individual SW components that comprised the OBE. The component test case procedures were based on the SW specification, while the whole OBE test case procedures consisted of a set of intersection driving scenarios intended to verify the OBE was operating as intended.

Following are the details of the MI intersections pertinent to a number of the intersection test cases followed by the individual test cases and test case status summary.

A.32 MI Intersection Layout, Lane and Approach Details

Simple Traffic Signal Approach Layout

From the CICAS-V Concept of Operations [4] the Simple Traffic Signal Approach Conditions are:

> *The CICAS-V enabled vehicle is approaching a CICAS-V enabled traffic signal at a simple intersection with no dedicated turn lanes, where all vehicles on the same approach have the same traffic signal indication.*

An example CICAS-V test intersection that exhibited these characteristics was the W. 10 Mile Rd. & Orchard Lake Rd. intersection located in Farmington, MI. By its nature this type of intersection only requires “which-road” positioning accuracy. While the intersection was equipped with an RSE providing the GPSC message this functionality was able to be disabled to test the “Limited Positioning Services” scenario. Following is a diagram of this intersection.

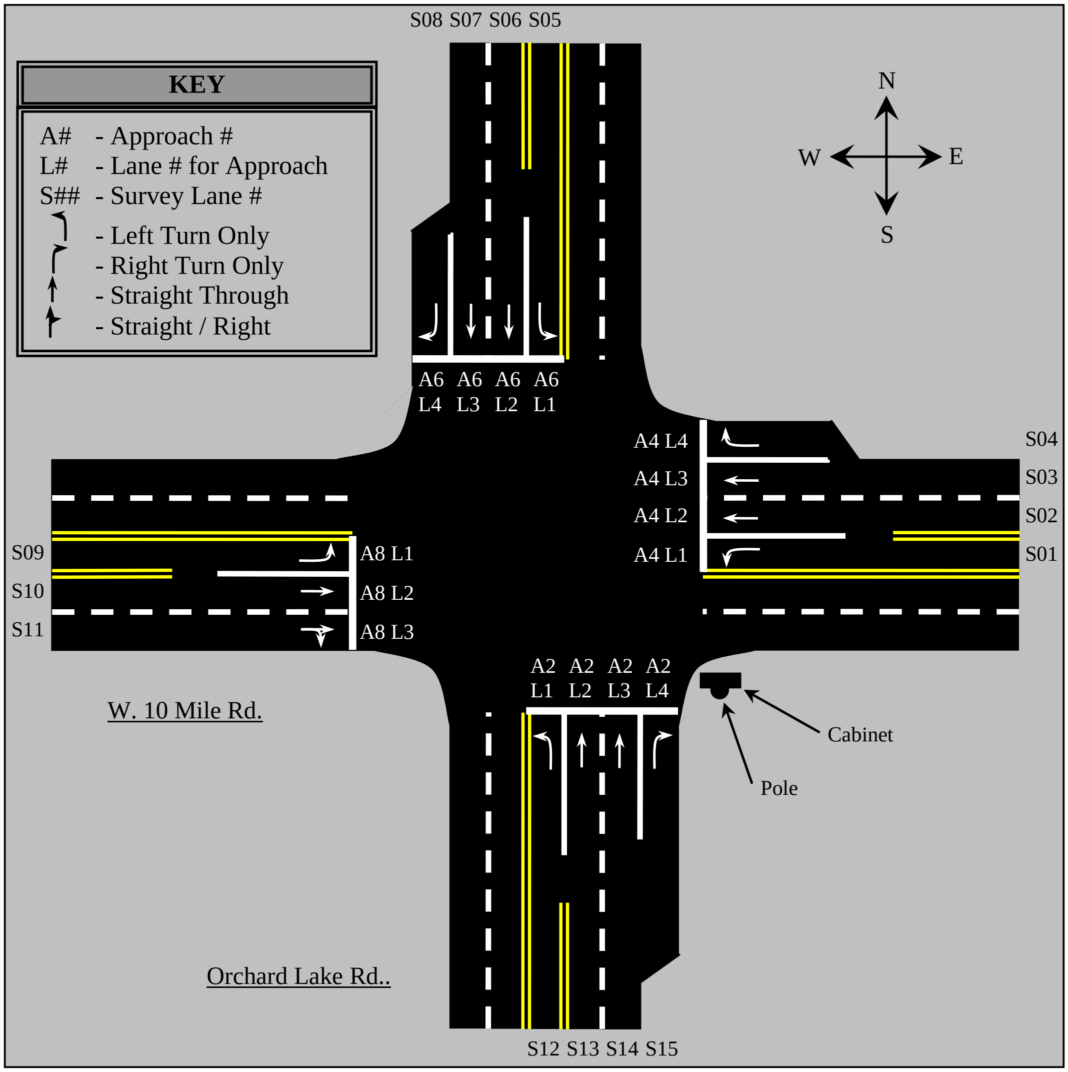

Figure 105: W. 10 Mile Rd. & Orchard Lake Rd. Intersection Layout

Table 201: W. 10 Mile Rd. & Orchard Lake Rd. Lane Data

Positioning Requirement:			Road Level – Requires limited positioning services		
Flashing Red Available:			No		
Survey Lane #	Approach #	Lane # for Approach	Furthest Lane Node to GID Ref. Point Distance (m)	Furthest Lane Node to Stop Bar Distance (m)	Lane Attributes
S01	A4	1	88.16	66.45	Left Allowed
S02	A4	2	250.03	227.66	Straight Allowed
S03	A4	3	314.69	292.32	Straight Allowed
S04	A4	4	84.99	62.93	Right Allowed
S05	A6	1	81.37	60.47	Left Allowed
S06	A6	2	300.82	279.16	Straight Allowed
S07	A6	3	278.79	257.07	Straight Allowed
S08	A6	4	91.26	69.76	Right Allowed
S09	A8	1	83.04	61.03	Left Allowed
S10	A8	2	295.50	273.00	Straight Allowed
S11	A8	3	220.94	198.54	Straight Allowed Right Allowed
S12	A2	1	84.42	64.25	Left Allowed
S13	A2	2	252.99	232.41	Straight Allowed
S14	A2	3	261.69	241.06	Straight Allowed
S15	A2	4	196.50	176.19	Right Allowed

Dedicated Turn Lane Traffic Signal Approach Layout

From the CICAS-V Concept of Operations, the Dedicated Turn Lane Traffic Signal Approach Conditions are:

> *The CICAS-V enabled vehicle is approaching a CICAS-V enabled intersection with multiple traffic signal indications on the approach.*

An example CICAS-V test intersection that exhibited these characteristics was the W. 12 Mile Rd. & Farmington Rd. intersection located in Farmington, MI which has dedicated left turn lanes. One of the phase cycles for the dedicated turn lanes was a flashing red with allowed the “Flashing Traffic Signal” scenario to be tested. Following is a diagram of this intersection.

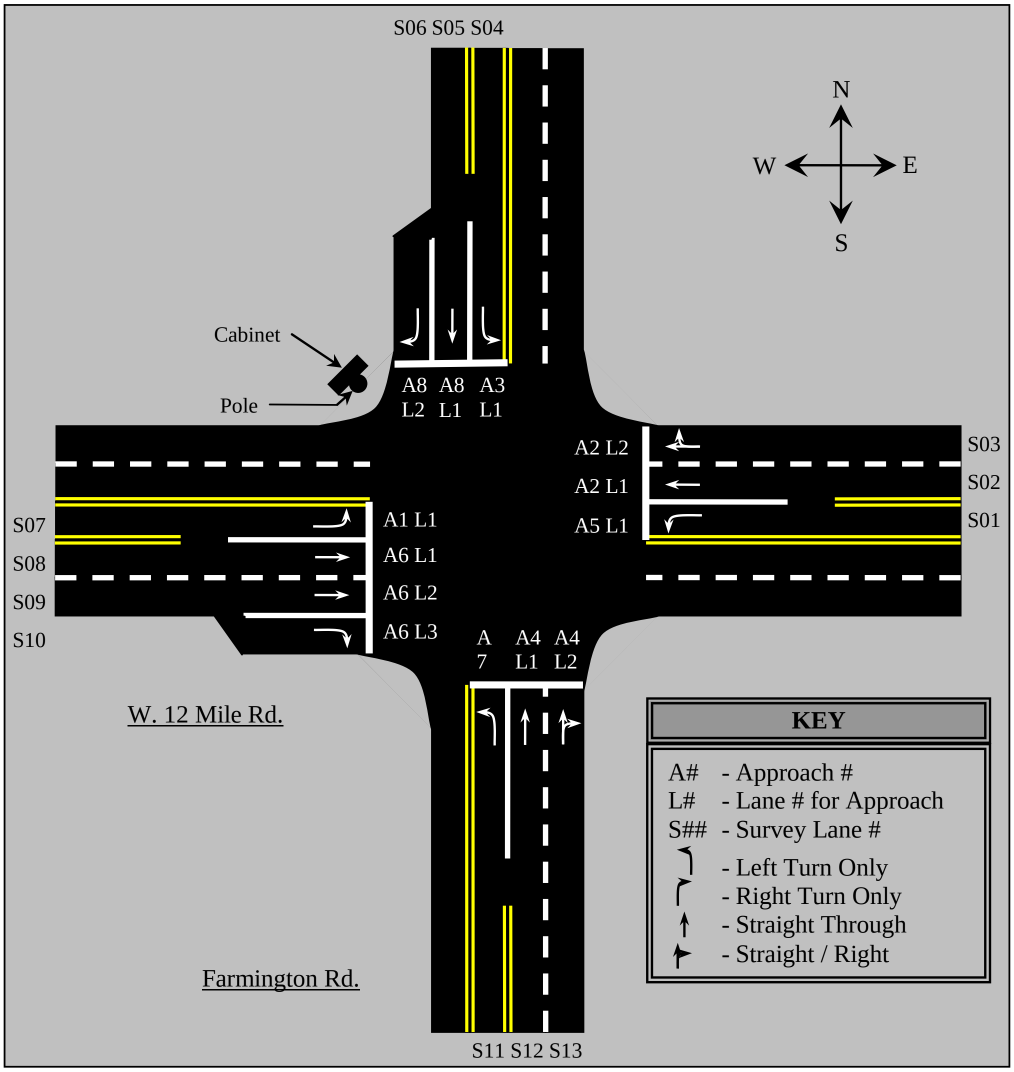

Figure 106: W. 12 Mile Rd. & Farmington Rd. Intersection Layout

Table 202: W. 12 Mile Rd. & Farmington Rd. Lane Data

Positioning Requirement:			Lane Level – Requires GPS correction messages		
Flashing Red Available:			Yes		
Survey Lane #	Approach #	Lane # for Approach	Furthest Lane Node to GID Ref. Point Distance (m)	Furthest Lane Node to Stop Bar Distance (m)	Lane Attributes
S01	A5	1	92.02	72.19	Left Allowed
S02	A2	1	238.44	218.29	Straight Allowed
S03	A2	2	239.15	218.95	Straight Allowed Right Allowed
S04	A3	1	112.41	93.10	Left Allowed

Positioning Requirement:			Lane Level – Requires GPS correction messages		
Flashing Red Available:			Yes		
Survey Lane #	Approach #	Lane # for Approach	Furthest Lane Node to GID Ref. Point Distance (m)	Furthest Lane Node to Stop Bar Distance (m)	Lane Attributes
S05	A8	1	295.98	276.51	Straight Allowed
S06	A8	2	94.94	75.52	Right Allowed
S07	A1	1	92.12	72.34	Left Allowed
S08	A6	1	288.24	268.07	Straight Allowed
S09	A6	2	265.09	244.83	Straight Allowed
S10	A6	3	92.65	72.79	Right Allowed
S11	A7	1	73.52	53.64	Left Allowed
S12	A4	1	324.80	304.40	Straight Allowed
S13	A4	2	260.23	239.97	Straight Allowed Right Allowed

Simple Stop Sign Approach Layout

From the CICAS-V Concept of Operations, the Simple Stop Sign Approach Conditions are:

> *The CICAS-V enabled vehicle is approaching a CICAS-V enabled, simple stop sign controlled intersection. It is presumed that the vehicle has previously obtained GID for the intersection.*

An example CICAS-V stop-controlled test intersection was the W. 11 Mile Rd. & Drake Rd. intersection located in Farmington, MI. There was no RSE located at this intersection which allowed the "Limited Positioning Services" scenario to be tested here as well. Following is a diagram of this intersection.

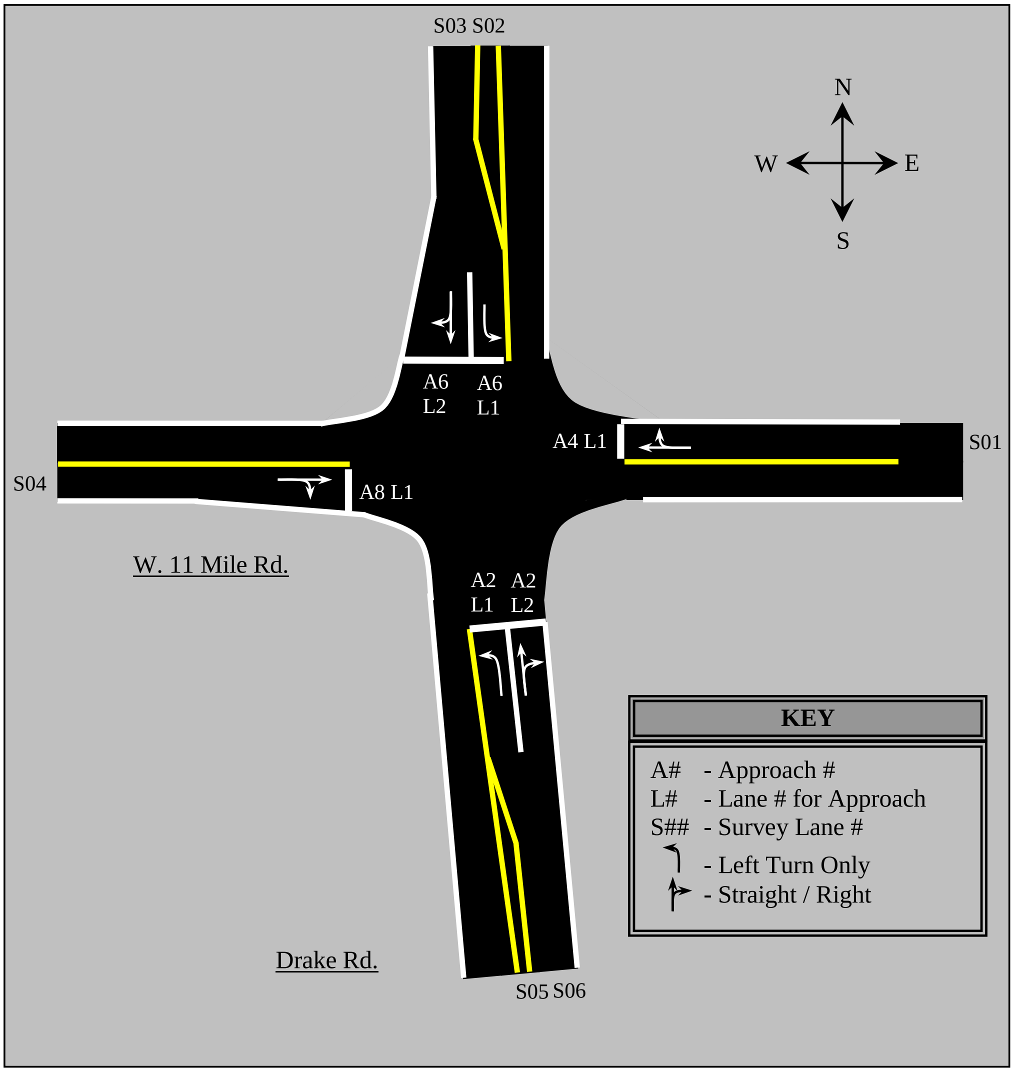

Figure 107: W. 11 Mile Rd. & Drake Rd. Intersection Layout

Table 203: W. 11 Mile Rd. & Drake Rd. Lane Data

Positioning Requirement:			Road Level		
Survey Lane #	Approach #	Lane # for Approach	Furthest Lane Node to GID Ref. Point Distance (m)	Furthest Lane Node to Stop Bar Distance (m)	Lane Attributes
S01	A4	1	274.85	255.99	Straight Allowed Right Allowed
S02	A6	1	51.77	33.50	Left Allowed
S03	A6	2	293.78	275.14	Straight Allowed Right Allowed
S04	A8	1	320.62	301.60	Straight Allowed

Positioning Requirement:			Road Level		
Survey Lane #	Approach #	Lane # for Approach	Furthest Lane Node to GID Ref. Point Distance (m)	Furthest Lane Node to Stop Bar Distance (m)	Lane Attributes
					Right Allowed
S05	A2	1	57.04	38.37	Left Allowed
S06	A2	2	322.80	312.67	Straight Allowed Right Allowed

A.33 Test Cases

Component Test Case Procedures

A.33.1.1 Vehicle Message Handler (VEH)

The Vehicle Message Handler is responsible for receiving and processing the CAN messages from the Netway6 device.

A.33.1.1.1 Test Cases

Test 1 – CMP-VEH-01: $600-$605 Batch CAN Processing

Test Case #: CMP-VEH-01	
Test Description:	To verify the OBE supports batch processing of the $600 - $605 Vehicle to OBE CAN messages.
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Send the complete set of $600 - $605 vehicle CAN messages into CAN1.	#1) Verify that the $600 - $605 messages are logged on the OBE

Test 2 – CMP-VEH-02: Incomplete $600-$605 CAN Message Set

Test Case #: CMP-VEH-02	
Test Description:	To verify that the OBE ignores an incomplete set of Vehicle to OBE CAN messages.
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Send an incomplete set of vehicle CAN messages (e.g., $600 - $604) into CAN1 preferably with a way to distinguish this set of messages.	#1) Verify that these messages are not logged on the OBE #2) Verify that an error indicating that 'Incomplete CAN data received' is logged
2) Send the complete set of $600 - $605 vehicle CAN	#1) Verify that this set of $600 - $605 CAN messages

Test Case #: CMP-VEH-02	
Test Description:	To verify that the OBE ignores an incomplete set of Vehicle to OBE CAN messages.
messages into CAN1, preferably with a way to distinguish this set of messages.	are logged on the OBE

Test 3 – CMP-VEH-03: Incomplete CAN Message Set $605 Processing

Test Case #: CMP-VEH-03	
Test Description:	To verify, for an incomplete set of Vehicle to OBE messages, that the $605 message is not a trigger to constitute a complete set.
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Send an incomplete set of vehicle CAN messages that includes the $605 message (e.g., $600, $601, $605) into CAN1.	#1) Verify that these messages are not logged on the OBE #2) Verify that an error indicating that 'Incomplete CAN data received' is logged
2) Send the complete set of $600 - $605 vehicle CAN messages into CAN1.	#1) Verify that this set of $600 - $605 CAN messages are logged on the OBE

Test 4 – CMP-VEH-04: CAN Timeout Support

Test Case #: CMP-VEH-04	
Test Description:	To verify that the OBE supports a default time out value for indicating that no Vehicle to OBE CAN messages have been received.
Test Setup / Equipment:	OBE with the default 'CANExpirationTime' configured time.
Test Steps	**Expected Results for Corresponding Test Step**
1) Send one or more complete set(s) of $600 - $605 vehicle CAN messages into CAN1	#1) Verify that these messages are logged on the OBE
2) Stop sending vehicle CAN data for more than 'CANExpirationTime'	#1) Verify that an error indicating 'CAN expired' is logged
3) Send one or more complete set(s) of $600 - $605 vehicle CAN messages into CAN1	#1) Verify that these messages are logged on the OBE

Test 5 – CMP-VEH-05: CAN Timeout Configuration Support

Test Case #: CMP-VEH-05	
Test Description:	To verify that the time out value, for indicating that no Vehicle to OBE CAN messages have been received, can be configured to specified minimum and maximum time values.
Test Setup / Equipment:	NOTE: The default, min, and max values all have the same value of 400ms. Will need to change the allowed min and max values in cicas-v.dflt to run this test.
Test Steps	**Expected Results for Corresponding Test Step**
1) Modify 'CANExpirationTime' to the minimum configurable time and restart the WSU	NA
2) Send one or more complete set(s) of $600 - $605 vehicle CAN messages into CAN1	#1) Verify that these messages are logged on the OBE
3) Stop sending vehicle CAN data for more than the newly configured 'CANExpirationTime'	#1) Verify that an error indicating 'CAN expired' is logged
4) Modify 'CANExpirationTime' to the maximum configurable time and restart the WSU	NA
5) Repeat steps #2-#3	Expected results from #2 and #3

Test 6 – CMP-VEH-06: CAN Reception Rate

Test Case #: CMP-VEH-06	
Test Description:	To verify that the OBE supports the nominal reception rate of 10Hz for the Vehicle to OBE CAN messages.
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Send multiple complete set(s) of $600 - $605 vehicle CAN messages into CAN1 at 10Hz.	#1) Verify that these messages are logged on the OBE

Test 7 – CMP-VEH-07: Message Reception Rate Variability

Test Case #: CMP-VEH-07	
Test Description:	To verify that the OBE can support rates other than the nominal rate of 10 Hz for the Vehicle to OBE CAN messages.
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Send multiple complete set(s) of $600 - $605 vehicle CAN messages into CAN1 at something other than 10 Hz but within the 'CANExpirationTime'	#1) Verify that these messages are logged on the OBE

Test 8 – CMP-VEH-08: Message $650 Support

Test Case #:	CMP-VEH-08	
Test Description:	To verify that the OBE supports receiving the $650 Message.	
Test Setup / Equipment:	Netway configured to indicate which data items are and are not supported.	
Test Steps		**Expected Results for Corresponding Test Step**
1) Send in the $650 vehicle CAN message with bits set to indicate which data items are and are not supported.		#1) Verify that the data items that are supported are indicated that way #2) Verify that the data items that are not supported are indicated that way

Test 9 – CMP-VEH-09: Heartbeat Support

Test Case #:	CMP-VEH-09	
Test Description:	To verify that the OBE supports heartbeats between the OBE and Netway.	
Test Setup / Equipment:	OBE with a properly configured Netway connected that supports heartbeat.	
Test Steps		**Expected Results for Corresponding Test Step**
1) OBE application starts up		#1) $704 message should be sent with the first three bytes indicating a value of one.
2) $606 Netway heartbeat messages are sent to OBE with the last three bytes indicating the correct ‘OBE to Netway6 Heartbeat Sequence’.		#1) As long as the $606 messages are transmitted at the proper rate and configured properly, the $704 message should be sent at the configured periodic rate with the sequence numbers increasing by one with each transmission.

Test 10 – CMP-VEH-10: $606 Heartbeat Error Processing

Test Case #:	CMP-VEH-10	
Test Description:	To verify that the appropriate $606 Heartbeat errors are handled properly	
Test Setup / Equipment:	OBE with a properly configured Netway connected that supports modifications to the heartbeat protocol for test purposes.	
Test Steps		**Expected Results for Corresponding Test Step**
1) Configure the Netway to not send the $606 message to the OBE.		#1) After the configurable time out time the OBE should log a CAN message $606 timeout. #2) After the configurable time out time the OBE should send a $704 message with the ‘Netway6 to OBE Heartbeat Error’ bit set
2) Configure the Netway to send the $606 message with an incorrect ‘Netway6 to OBE Heartbeat Sequence’.		#1) Upon reception of $606 message the OBE should log in invalid Netway to OBE heartbeat error

Test Case #: CMP-VEH-10	
Test Description:	To verify that the appropriate $606 Heartbeat errors are handled properly
3) Configure the Netway to send the $606 message with the 'OBE to Netway6 Heartbeat Error' flag set.	#1) Upon reception of $606 message the OBE should log an OBE to Netway heartbeat error
4) Configure the Netway to send the $606 message with the 'Vehicle CAN Data Timeout' set to a 1, 2, 4, or 8	#1) Upon reception of $606 message the OBE should log a vehicle CAN timeout error.

Test 11 – CMP-VEH-11: Logging Enable / Disable

Test Case #: CMP-VEH-11	
Test Description:	To verify that the logging flag for CVIP can be enabled and disabled.
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Start up the WSU and make sure the CVI log file does not exist. If it does delete it.	NA
2) Enable only the CVIDebugFlag	#1) Verify that CVI data is logged to the console
3) Enable only the CVIInfoFlag	#1) Verify that a smaller amount of CVI data is logged to the console
4) Enable the CVILogFlag	#1) Verify that a persistent CVI log file is created on the WSU
5) Repeat steps #2-#3	Expected results from #2 and #3

Radio Handler / Data Demux (RAD)

The Radio Handler / Data Demux (radiodemux) is primarily responsible for DSRC message traffic on the WSU. It interfaces with the radio through the WSU Software Services API (WSS API) and the Radio Services module. Radiodemux provides WSM data that passes its message validation filter to applications on the OBE that register as WSM Users with specific supported PSIDs. These include the GID DB Handler which receives GID WSMs, the SPaT Handler which receives SPaT WSMs, and the GPS Handler which receives GPSC WSMs.

Radiodemux also drives radio configuration on launch, receives radio stats from the WSS API, and requests WSAs. But these functions are not explicitly tested here except insofar as they are needed for basic radio/API operation.

A.33.1.1.2 Setup

To perform the tests below the traffic signal controller is not needed by the RSE unless testing with live SPaT messages is desired. Likewise, on the OBE, neither the Netway6 nor DAS devices are needed. But both OBE and RSE must have functional GPS because the DSRC radios require consistent PPS.

Many of the tests below require the tester to perform the following setup steps first.

On the OBE:

a. Log into the OBE as root via telnet or serial console, if available.
b. Change the CICAS-V Interface Config file, /rwflash/configs/CICAS_IF_config.txt, to enable output to CONSOLE and enable VIIC compatibility if the RSE requires it. (Save the config file first for easy restoration later.)
c. On the OBE run radiodemux (Do not hit ^C to quit or you will have to reboot.).
d. Separately, log into the RSE as root via telnet or serial console, if available.
e. Ensure no other DSRC message activity is going on. Kill User/Provider processes on OBE and RSE as needed.
f. Set the VII compatibility flag if necessary. (This is not necessary on the WSU RSE.)
g. Verify that PPS is being received on both machines and increments once per second.

For some test cases you do not need to run radiodemux. Start the CICAS Application suite and do not have radiodemux running for those tests.

When testing is complete, restore the config file if necessary.

On the RSE:

a. Use the sling tool to send WSMs for a service, or in some cases, use the appropriate server (GID, SPaT, or GPSC).
b. Ensure that the installed test programs are the correct versions for the system/application release.

A test program other than radiodemux, sling or whatever is noted, may be substituted so long as it performs the indicated operations and displays evidence of success or failure to perform the operations. The test step required to cause the expected result may differ from the given test case but the operations and results should be identical.

A.33.1.1.3 Test Cases

A.33.1.1.3.1 Radio Handling Test Cases

Test 12 – CMP-RAD-RH-01: Verify WBSS Join

Test Case #: CMP-RAD-RH-01	
Test Description:	Verify WBSS Join
Test Setup / Equipment:	See setup steps above
Test Steps	**Expected Results for Corresponding Test Step**

On the OBE, run radiodemux to verify that the CICAS-V WBSS can be joined.	Radiodemux joins the WBSS and does not indicate failure.

Test 13 – CMP-RAD-RH-02: Verify WBSS Detach

Test Case #: CMP-RAD-RH-02		
Test Description:	Verify WBSS Detach	
Test Setup / Equipment:	See setup steps above.	
Test Steps		**Expected Results for Corresponding Test Step**
On the OBE, while running radiodemux, hit the Enter key to verify that the CICAS-V WBSS can be detached.		Radiodemux terminates normally and does not indicate failure to detach from WBSS.

Test 14 – CMP-RAD-RH-03: Verify Periodic Radio Stats Requests

Test Case #: CMP-RAD-RH-03		
Test Description:	Verify Periodic Requests for Radio Stats	
Test Setup / Equipment:	See setup steps above (except RSE-related steps).	
Test Steps		**Expected Results for Corresponding Test Step**
On the OBE, verify that radiodemux periodically requests radio statistics from Radio Services.		Radiodemux periodically requests radio statistics without error or warning.

Test 15 – CMP-RAD-RH-04: Verify Radio Stats Polling Rate Configuration

Test Case #: CMP-RAD-RH-04		
Test Description:	Verify Radio Stats Polling Rate is Configurable	
Test Setup / Equipment:	See setup steps above (except RSE-related steps). NOTE: The default, min, and max values all have the same value of 1000ms. Will need to change the allowed min and max values in cicas-v.dflt to run this test.	
Test Steps		**Expected Results for Corresponding Test Step**
On the OBE, verify that the Radio Statistics polling rate configuration parameter (RadioStatisticsPollingInterval) supports the minimum configurable value.change.		Radiodemux polls radio statistics at the minimum rate as specified.
On the OBE, verify that the Radio Statistics polling rate configuration parameter supports the maximum configurable value.		Radiodemux polls radio statistics at the maximum rate as specified.

A.33.1.1.3.2 Radio Service Configuration Test Cases

Test 16 – CMP-RAD-RSC-01: Verify PSID Configuration

Test Case #: CMP-RAD-RSC-01	
Test Description:	Verify PSIDs are Configurable
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
1) On the OBE, temporarily configure the GID PSID (GIDPSID) to 01010101 (or anything other than the default value). 2) On the RSE, use the GID Server to transmit valid GIDs to the service with the specified PSID.	Radiodemux routes the GIDs to the GID Database Handler and logs the GID data.
3) Repeat steps #1-#2 for the SPaT PSID (SPATPSID)	Radodemux routes the SPaT messages to the SPaT Database Handler and logs the SPaT data.
4) Repeat steps #1-#2 for the GPSC PSID (GPSCPSID)	Radiodemux routes the GPSC messages to the GPS Database Handler and logs the GPSC data.

Test 17 – CMP-RAD-RSC-02: Unrecognized PSID Handling

Test Case #: CMP-RAD-RSC-02	
Test Description:	Verify Handling of Unrecognized PSID
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
On the RSE, use any server to transmit otherwise valid WSMs to a PSID not specified in the CICAS configuration file.	Radiodemux discards them and no data is logged

A.33.1.1.3.3 TOM Header Failure Tests

Test 18 – CMP-RAD-HDR-01: Invalid Message Type (TOM Header)

Test Case #: CMP-RAD-HDR-01	
Test Description:	Invalid Message Type (TOM Header)
Test Setup /	See setup steps above.

Equipment:	

Test Steps	Expected Results for Corresponding Test Step
On the RSE, use *sling* to transmit otherwise valid WSMs with TOM Header Message Type incorrectly set to any value other than 0xF1.	Radiodemux discards them and indicates that the TOM header check failed.

Test 19 – CMP-RAD-HDR-02: Unsupported TOM Framework Version

Test Case #:	CMP-RAD-HDR-02
Test Description:	Unsupported TOM Framework Version (TOM Header)
Test Setup / Equipment:	See setup steps above.

Test Steps	Expected Results for Corresponding Test Step
On the RSE, use *sling* to transmit otherwise valid WSMs with unsupported TOM Framework Version value(s).	Radiodemux discards them and indicates that the header check failed.

Test 20 – CMP-RAD-HDR-03: Incorrect / Zero Message Length

Test Case #:	CMP-RAD-HDR-03
Test Description:	Incorrect, Zero Message Length (TOM Header)
Test Setup / Equipment:	See setup steps above.

Test Steps	Expected Results for Corresponding Test Step
On the RSE, use *sling* to transmit otherwise valid WSMs with data in the payload but a zero Message Length.	Radiodemux discards them

Test 21 – CMP-RAD-HDR-04: Incorrect / Too Small Message Length

Test Case #:	CMP-RAD-HDR-04
Test Description:	Incorrect, Too Small Message Length (TOM Header)
Test Setup / Equipment:	See setup steps above.

Test Steps	Expected Results for Corresponding Test Step
On the RSE, use *sling* to transmit otherwise valid WSMs with data in the payload but with a non-zero Message Length that is too small.	Radiodemux discards them

Test 22 – CMP-RAD-HDR-05: Incorrect / Too Large Message Length

Test Case #:	CMP-RAD-HDR-05	
Test Description:	Incorrect, Too Large Message Length (TOM Header)	
Test Setup / Equipment:	See setup steps above.	
Test Steps		**Expected Results for Corresponding Test Step**
On the RSE, use *sling* to transmit otherwise valid WSMs with data in the payload but with a Message Length that is too large.		Radiodemux processes valid data as normal and discards the rest.

Test 23 – CMP-RAD-HDR-06: Incorrect CRC-16

Test Case #:	CMP-RAD-HDR-06	
Test Description:	Incorrect CRC-16 (TOM Header)	
Test Setup / Equipment:	See setup steps above.	
Test Steps		**Expected Results for Corresponding Test Step**
On the RSE, use *sling* to transmit otherwise valid WSMs with incorrect CRC-16 value.		Radiodemux discards them and indicates that a header check failed

A.33.1.1.3.4 TOM Footer Failure Tests

Test 24 – CMP-RAD-FTR-01: Invalid Message Termination Flag

Test Case #:	CMP-RAD-FTR-01
Test Description:	Invalid Message Termination Flag (TOM Footer)
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
On the RSE, use *sling* to transmit otherwise valid WSMs with Message Termination Flag incorrectly set to any value other than 0xF1.	Radiodemux discards them

A.33.1.1.3.5 Null Cases

Test 25 – CMP-RAD-NULL-01: Empty Message

Test Case #:	CMP-RAD-NULL-01
Test Description:	Empty Message
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
On the RSE, use *sling* to transmit WSMs containing TOM Header and TOM Footer, no payload.	Radiodemux discards them

Test 26 – CMP-RAD-NULL-02: Empty Layer

Test Case #:	CMP-RAD-NULL-02
Test Description:	Empty Layer
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
On the RSE, use *sling* to transmit WSMs containing TOM Header, a Layer Object of any supported type, no payload, a Close Layer Object, and TOM Footer.	Radiodemux discards them

A.33.1.1.3.6 Success Test Cases

Test 27 – CMP-RAD-NML-01: Normal GID on CCH

Test Case #: CMP-RAD-NML-01	
Test Description:	Normal GID on CCH
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
On the RSE, use the GID Server to transmit a valid GID WSM on the CCH.	Radiodemux passes it to the GID Database Handler without error or warning.

Test 28 – CMP-RAD-NML-02: Normal GID on SCH

Test Case #: CMP-RAD-NML-02	
Test Description:	Normal GID on SCH
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
On the RSE, use the GID Server to transmit a valid GID WSM on the SCH.	Radiodemux switches to the SCH, receives the GID message, and passes it to the GID Database Handler without error or warning.

Test 29 – CMP-RAD-NML-03: Normal SPaT

Test Case #: CMP-RAD-NML-03	
Test Description:	Normal SPaT
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
On the RSE, use the SPaT Server to transmit a valid SPaT WSM on the CCH.	Radiodemux passes it to the SPaT Handler without error or warning.

Test 30 – CMP-RAD-NML-04: Normal GPSC on CCH

Test Case #: CMP-RAD-NML-04	
Test Description:	Normal GPSC on CCH
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**

On the RSE, use the GPSC Server to transmit a valid GPSC WSM on the CCH.	Radiodemux passes it to the GPS Handler without error or warning.

Test 31 – CMP-RAD-NML-05: Normal GPSC on SCH

Test Case #: CMP-RAD-NML-05	
Test Description:	Normal GPSC on SCH
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
On the RSE, use the GPSC Server to transmit a valid GPSC WSM on the SCH.	Radiodemux switches to the SCH, receives the GPSC message, and passes it to the GPS Handler without error or warning.

Test 32 – CMP-RAD-NML-06: Maximum DSRC Message Receipt Support

Test Case #: CMP-RAD-NML-06	
Test Description:	Maximum size DSRC message receipt support
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
On the RSE, use the GPSC Server to transmit a DSRC message that is equal to the maximum size allowed by the standards. NOTE: This could be an existing GID where the existing approach data is duplicated over and over while increasing the approach id's until the message size is as close to the max size as possible. To account for the remaining bytes an unsupported object could be placed somewhere prior to the end of the message.	The message should be received and processed by the OBE.

A.33.1.1.3.7 Layer Test Cases

Test 33 – CMP-RAD-LAY-01: Unsupported Layer

Test Case #:	CMP-RAD-LAY-01
Test Description:	Unsupported Layer
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
On the RSE, use *sling* to transmit WSMs containing TOM Header, a Layer Object of any unsupported type, a payload, a Close Layer Object, and TOM Footer.	Radiodemux discards them

Test 34 – CMP-RAD-LAY-02: Dangling Layer

Test Case #:	CMP-RAD-LAY-02
Test Description:	Dangling Layer
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
On the RSE, use *sling* to transmit WSMs containing TOM Header, a Layer Object of any supported type, a payload, *no* Close Layer Object, and TOM Footer.	Radiodemux discards them

A.33.1.1.3.8 Radio Request to Transmit Messages

~~Test 35 – CMP-RAD-RHTX-01: TSVWG Transmission~~

~~**Test Case #:**~~	~~CMP-RAD-RHTX-01~~
~~**Test Description:**~~	~~Radio Handler transmits TSVWG based upon a request from the Warning Algorithm~~
~~**Test Setup / Equipment:**~~	~~CICAS application and WSU software services are executed on the OBE.~~ ~~CICAS application is configured to send a TSVWG.~~
~~**Test Steps**~~	~~**Expected Results for Corresponding Test Step**~~
~~1) CICAS vehicle runs a test where a Traffic Signal Violation Warning is given.~~ ~~2) Warning Algorithm requests to send a TSVWG.~~	~~Logfile entry has a TSVWG sent included.~~

NOTE: The SW supports sending this message, however, there is no longer a need for it. Thus this functionality will not be tested.

~~Test 36 – CMP-RAD-RHTX-02: RCMD Transmission~~

~~**Test Case #:**~~	~~CMP-RAD-RHTX-02~~	
~~**Test Description:**~~	~~Radio Handler transmits RCMD based upon a request from the Warning Algorithm~~	
~~**Test Setup / Equipment:**~~	~~CICAS application and WSU software services are executed on the OBE.~~ ~~CICAS application is configured to send RCMD at specified time to stop bar.~~	
~~**Test Steps**~~		~~**Expected Results for Corresponding Test Step**~~
~~1) CICAS vehicle runs a test where a Traffic Signal Violation Warning is given.~~ ~~2) Warning Algorithm requests to send a RCMD.~~		~~Logfile entry has a RCMD sent included.~~

NOTE: The SW supports sending this message, however, there is no longer a need for it. Thus this functionality will not be tested.

A.33.1.1.3.9 Radio Handler Heartbeat Message

~~Test 37 – CMP-RAD-HB-01: Heartbeat transmission to Error Handler~~

~~**Test Case #:**~~	~~CMP-RAD-HB-01~~	
~~**Test Description:**~~	~~Successful heartbeat messages are sent out to error handler~~	
~~**Test Setup / Equipment:**~~	~~CICAS application and WSU software services are running.~~ ~~Configure the error handler watchdog timer and radio handler heartbeat message configuration values so that heartbeat messages are sent within watchdog timeouts.~~	
~~**Test Steps**~~		~~**Expected Results for Corresponding Test Step**~~
~~Start CICAS application.~~		~~Check log file to see if the Error Handler is receiving heartbeat messages.~~

NOTE: This is being tested as part of SW Watchdog Monitor test cases.

~~Test 38 – CMP-RAD-HB-02: Heartbeat Error~~

~~**Test Case #:**~~	~~CMP-RAD-HB-02~~	
~~**Test Description:**~~	~~Configuration of heartbeat messages to create error~~	
~~**Test Setup / Equipment:**~~	~~CICAS application and WSU software services are running.~~ ~~Configure the error handler watchdog timer and radio handler heartbeat message configuration values so that heartbeat messages are not sent within watchdog timeouts.~~	
~~**Test Steps**~~		~~**Expected Results for Corresponding Test Step**~~
~~Start CICAS application.~~		~~Check log file to see if the Error Handler is logging an error in the radio handler heartbeat.~~

NOTE: This is being tested as part of SW Watchdog Monitor test cases.

A.33.1.2 GPS Handler (GPSH)

The GPS Handler interfaces to the NovAtel OEMV GPS receiver through the Time/Positioning Services (TPS) provided on the WSU. The TPS interfaces to a GPS receiver to obtain time and position updates. The TPS also receives RTCM corrections which it sends to the GPS receiver.

A.33.1.2.1 Test Cases

Test 39 – CMP-GPSH-01: GPS Handler Registration

Test Case #:	CMP-GPSH-01	
Test Description:	Register the GPS Handler module with the WSU TPS to receive GPS data updates. (wsuTpsInit, wsuTpsRegister)	
Test Setup / Equipment:		
Test Steps		**Expected Results for Corresponding Test Step**
1) Enable logging of GPS data to the console. The application should call wsuTpsInit first, and then call wsuTpsRegister.		#1) Verify that data appears on the console, and that the GPS coordinates represent a valid position consistent with surveyed or reliable previous recorded position data.

Test 40 – CMP-GPSH-02: GPS Healthy Status Check

Test Case #:	CMP-GPSH-02	
Test Description:	Upon receipt of a GPSC WSA, check if the GPS Status Flag indicates Healthy. (Error indications are listed in CMP-GPSH-08.)	
Test Setup / Equipment:		
Test Steps		**Expected Results for Corresponding Test Step**
1) Enable logging of GPS data to the console.		#1) Verify data appearing on the console
2) Modify the code running on the RSE side to set the GPS Healthy status flag bit in the GPS Status field of the GPS Corrections TOM message to indicate that the base station GPS receiver is not healthy. Verify the RSE transmission includes the "unhealthy" status bit (run Gypsy with debug + verbose mode). [This could alternately be done by (1) creating a GPS corrections message indicating "Not Healthy," and using *sling* to send it; or (2) adding a command line option to gypsy that will cause it to send a message indicating "Not Healthy"]		#2) The LED on the vehicle GPS Receiver's serial port that is connected to the WSU will flash red when data is being sent from the WSU to the vehicle GPS receiver. It will flash green when the vehicle GPS receiver is sending data to the WSU. Verify that the COM port LED on the vehicle GPS receiver does NOT flash RED while the RSE broadcasts the "unhealthy" status. (The LED may appear yellow/orange if the transmit and receive events are simultaneous) #3) Verify that the data no longer appears on the console.

Test 41 – CMP-GPSH-03: NMEA Data Processing

Test Case #:	CMP-GPSH-03	
Test Description:	Upon receipt of a NMEA input, check if a checksum error has occurred or no solution is available. If a solution is available, output the parsed NMEA data to other modules. (Intersection Identification [II] and Map Matching/Lane Identification [MM/LI]). (Error indications are listed in CMP-GPSH-08.)	
Test Setup / Equipment:		

Test Steps	**Expected Results for Corresponding Test Step**
1) Enable logging of the II New approaching intersection event. Go to a "functioning intersection" (as described in 1.4.2 Dependencies) from a non-functioning intersection, from a non-functioning intersection to a functioning intersection, or from one functioning intersection to a different functioning intersection.	#1) Examine the log to verify that a change of approaching intersection ID (or change to none available) event appears in the log.
2) Enable logging of the MM/LI Algorithm results (successful lane identification) and MM/LI Algorithm results (unsuccessful lane identification) events. Go to a "functioning intersection" (as described in 1.4.2 Dependencies) from a non-functioning intersection, or from a non-functioning intersection to a functioning intersection.	#2) Examine the log to verify that either MM/LI event appears in the log.
3) Remove the OBE GPS from the OBE WSU. Plug in a PC and use a terminal program or similar mechanism to transmit a NMEA message with an invalid checksum.	#3) Examine the log to verify the bad checksum was caught.

Test 42 – CMP-GPSH-04: GPS Data Timer Expiration

Test Case #:	CMP-GPSH-04
Test Description:	If no GPS input has been received or no GPS solution has been received for a configurable period of time, the previous data shall be considered expired. Output a data invalid indication to other modules. (Error indications are listed in CMP-GPSH-08.)
Test Setup / Equipment:	

Test Steps	Expected Results for Corresponding Test Step
1) Enable the logging of the GPS Handler's GPS data timeout (no fix) event, and GPS data timeout (no data) error. (a) Disconnect the antenna of the vehicle's GPS receiver to generate the "no fix" event. (b) Disconnect the serial port of the vehicle's GPS receiver to generate the "no data" error.	#1) Examine the log to verify that the appropriate event or error appears in the log.
2) Repeat steps 1 and 2 of Test Case CMP-GPSH-03 after performing step 1 of this Test Case	#2) Examine the log file to verify that under these error conditions none of the indicated events are logged.
3) Double the period of time. (a) Enable the logging of the GPS Handler's GPS data timeout (no fix) event, and GPS data timeout (no data) error. (b) Disconnect the antenna of the vehicle's GPS receiver to generate the "no fix" event. (c) Disconnect the serial port of the vehicle's GPS receiver to generate the "no data" error.	#3) Examine the log to verify that the appropriate event or error appears in the log, happening in a timeframe consistent with the doubled period.
4) Halve the period of time. (a) Enable the logging of the GPS Handler's GPS data timeout (no fix) event, and GPS data timeout (no data) error. (b) Disconnect the antenna of the vehicle's GPS receiver to generate the "no fix" event. (c) Disconnect the serial port of the vehicle's GPS receiver to generate the "no data" error.	#4) Examine the log to verify that the appropriate event or error appears in the log, happening in a timeframe consistent with the halved period.

Test 43 – CMP-GPSH-05: DGPS Metric Object Processing

Test Case #: CMP-GPSH-05		
Test Description:	Upon receipt of a TOM with GPS corrections: If the TOM contains a Metric Object, calculate the elapsed time since the time in the Metric Object and output the elapsed time data to the DAS Handler/Logger module	
Test Setup / Equipment:		
Test Steps		**Expected Results for Corresponding Test Step**
1) Create a TOM with a Metric Object to send to the GPS Handler. 2) Enable logging of the GPS Metric Object in the DAS Handler. 3) Send the TOM to the GPS Handler.		#1) Examine the output of the DAS Handler to confirm the occurrence of the Metric Object - CAN message $751.

Test 44 – CMP-GPSH-06: DGPS Unhealthy Check

Test Case #: CMP-GPSH-06		
Test Description:	Upon receipt of a TOM with GPS corrections: Check if the GPS Status flag indicates Healthy. If not, discard the data. (Error indications are listed in CMP-GPSH-08.)	
Test Setup / Equipment:		
Test Steps		**Expected Results for Corresponding Test Step**
1) Modify the code running on the RSE side to set the GPS Healthy status flag bit in the GPS Status field of the GPS Corrections TOM message to indicate that the base station GPS receiver is not healthy. Verify the RSE transmission includes the "unhealthy" status bit (run Gypsy with debug + verbose mode). [This could alternately be done by (1) creating a GPS corrections message indicating "Not Healthy", and using *sling* to send it; or (2) adding a command line option to gypsy that will cause it to send a message indicating "Not Healthy"]		#1) The LED on the vehicle GPS Receiver's serial port that is connected to the WSU will flash red when data is being sent from the WSU to the vehicle GPS receiver. It will flash green when the vehicle GPS receiver is sending data to the WSU. Verify that the COM port LED on the vehicle GPS receiver does NOT flash RED while the RSE broadcasts the "unhealthy" status. (The LED may appear yellow/orange if the transmit and receive events are simultaneous)
2) Connect a serial terminal to the open COM port of the Vehicle GPS (e.g., use COM2 if the receiver COM1 is connected to the WSU or vice-versa). Request a log of NMEA GPGGA to the terminal (same as being transmitted to the WSU)		#2)† The "GPS Qual" field should indicate a value of 1 (GPS-Fix / not-corrected) or 9 (WAAS-corrected)

† The sixth value of the $GPGGA message, following the "$GPGGA" itself, is the Fix/Quality Indicator. A value of **0** indicates a fix was not available or invalid, a value of **1** indicates an uncorrected GPS fix, a value of **5** indicates use of RTCM 1001 and 1005 corrections, and a value of **9** indicates use of WAAS

corrections. For the test cases above which are only interested in RTCM corrections, any Fix/Quality value other than a 5 will be considered “non-corrected” NMEA data!

Test 45 – CMP-GPSH-07: RTCM Checksum Check

Test Case #: CMP-GPSH-07	
Test Description:	Upon receipt of a TOM with GPS corrections: Validate the Radio technical Commission for Maritime Services (RTCM) 1005 and 1001 checksums. If correct, output the corrections to the GPS receiver. (Error indications are listed in CMP-GPSH-08.)
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Enable the logging of the GPS Handler’s GPSC RTCM checksum failure. Modify the code on the RSE side to guarantee a bad checksum. Verify the RSE transmission includes the incorrect checksum values (run Gypsy with debug + verbose mode).	#1) Verify that the COM port LED on the vehicle GPS receiver does NOT flash RED while the RSE broadcasts the “unhealthy” status. Examine the log file to verify the presence of the GPSC RTCM checksum failure.
2) Connect a serial terminal to the open COM port of the Vehicle GPS. Request a log of NMEA GPGGA to the terminal (same as being transmitted to the WSU) [This could alternately be done by (1) creating a GPS corrections message indicating “Not Healthy”, and using *sling* to send it; or (2) adding a command line option to gypsy that will cause it to send a message indicating “Not Healthy”]	#2)† The “GPS Qual” field should indicate a value of 1 (GPS-Fix / not-corrected) or 9 (WAAS-corrected), after approximately 60 seconds of invalid RTCM data.
3) On the RSE, remove the code modification that causes the checksum to be incorrect. Connect a serial terminal (or emulator) to the GPS receiver’s serial port of the OBE WSU.	#3) Verify on the RSE that data is being transmitted with a valid checksum. Binary RTCM 1001 and 1005 corrected NMEA data should be visible on the OBE terminal.
4) Reconnect the GPS receiver to the WSU	#4) Verify that the COM port LED on the vehicle GPS receiver DOES flash RED while the RSE broadcasts the “unhealthy” status.
5) Connect a serial terminal to the open COM port of the Vehicle GPS. Request a log of NMEA GPGGA to the terminal (same as being transmitted to the WSU)	#5) †The “GPS Qual” field should indicate a value of 5 (RTK floating ambiguity solution)

† The sixth value of the $GPGGA message, following the “$GPGGA” itself, is the Fix/Quality Indicator. A value of **0** indicates a fix was not available or invalid, a value of **1** indicates an uncorrected GPS fix, a value of **5** indicates use of RTCM 1001 and 1005 corrections, and a value of **9** indicates use of WAAS corrections. For the test cases above which are only interested in RTCM corrections, any Fix/Quality value other than a 5 will be considered “non-corrected” NMEA data!

~~Test 46 – CMP-GPSH-08: Error Reporting to Error Handler~~

~~**Test Case #:**~~ ~~CMP-GPSH-08~~	
~~**Test Description:**~~	~~Report an error indication to the Error Handler upon any of the following conditions: (a) GPSC WSA indicates GPS status is not Healthy; (b) WSU TPS indicates a NMEA checksum error has occurred; (c) No GPS input; (d) No GPS solution; (e) GPS data expiration; (f) GPSC indicates RSE GPS status is not Healthy; and (g) RTCM checksum error.~~
~~**Test Setup / Equipment:**~~	
~~**Test Steps**~~	~~**Expected Results for Corresponding Test Step**~~
~~1) The Error Handler is responsible for the logging of these errors. If the capability to log these errors does not currently exist, it must be added.~~ ~~2) These error conditions can be created either by modifying the code running on the RSE to set a specific value, or by changing the OBE hardware (for example, disconnecting the GPS antenna), as described in the preceding test cases.~~	~~1) Any of the seven errors indicated above should appear in the Error Handler's log, or be displayed on the console, as configured.~~

NOTE: Due to Phase 2 of the CICAS-V project being put on hold not all of the Error Handling functionality was implemented. This will need to be revisited if there is a Phase 2.

~~Test 47 – CMP-GPSH-09: GPS Handler Heartbeat Support~~

~~**Test Case #:**~~ ~~CMP-GPSH-09~~	
~~**Test Description:**~~	~~Output a periodic software heartbeat message to the Error Handler at a configurable interval.~~
~~**Test Setup / Equipment:**~~	
~~**Test Steps**~~	~~**Expected Results for Corresponding Test Step**~~
~~1) Configure the Error Handler to log or record the arrival of periodic software heartbeat messages.~~ ~~2) Change the periodicity to a value approximately one half the Error Handler timeout value. Verify the timing by examining the Error Handler output.~~ ~~3) Change the periodicity to a value slightly less than the Error Handler timeout value. Verify the timing by examining the Error Handler output.~~ ~~4) Change the periodicity to a value slightly more than the Error Handler timeout value. Verify the reboot cause by examining the Error Handler output.~~	~~1) Check the Error Handler output to verify that the arrival of the periodic software Heartbeat message from this handler correlates to the configured periodicity.~~ ~~2) A reboot should only occur when the periodic rate exceeds the timeout value of the Error Handler. When a reboot occurs, examine the Error Handler log to verify the heartbeat not received was that of the GPS Handler.~~

NOTE: The SW as implemented does not support this test. This functionality can be confirmed via other tests.

A.33.1.3 GID Database Handler (GID)

A.33.1.3.1 Setup

An RSE is required to send the GIDs. The RSE shall execute the GID Server software and have configurations (input data) to execute for specific tests.

The OBE shall execute the WSU SW Services and the CICAS-V application.

The CDIP Log file will be used to check for appropriate log messages.

A.33.1.3.2 Test Cases

Test 48 – CMP-GID-01: GID Object Parsing

Test Case #:	CMP-GID-01
Test Description:	To verify that the supported Intersection GID objects are properly parsed.
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
Send supported GID from RSE.	CDIP Log has entry for New GID received.

Test 49 – CMP-GID-02: Un-Supported GID Object Parsing

Test Case #:	CMP-GID-02
Test Description:	To verify that the supported Intersection GID objects are properly parsed when the GID also contains un-supported objects
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
Send GID with un-supported objects.	CDIP Log has entry for New GID received.

Test 50 – CMP-GID-03: Un-supported GID Format Version Processing

Test Case #:	CMP-GID-03
Test Description:	To verify that a received GID with an un-supported Format Version is discarded.
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
Send GID with un-supported Format Version	CDIP Log contains no new record for GID received.

Test 51 – CMP-GID-04: Multiple GID Storage

Test Case #:	CMP-GID-04	
Test Description:	To verify that multiple GIDs can be stored in NVM up to the configurable GID storage allocation.	
Test Setup / Equipment:	See setup steps above.	
Test Steps		**Expected Results for Corresponding Test Step**
1) Configure the GID storage allocation. 2) Send multiple GIDs (all with unique Intersection ID) to fill up storage space.		CDIP Log has entry for Unexpired GID deleted.

Test 52 – CMP-GID-05: GID Storage Configuration

Test Case #:	CMP-GID-05	
Test Description:	To verify that the GID storage allocation is configurable.	
Test Setup / Equipment:	See setup steps above.	
Test Steps		**Expected Results for Corresponding Test Step**
1) Configure the GID storage allocation. 2) Send multiple GIDs (all with unique Intersection ID) to fill up storage space.		CDIP Log has entry for Unexpired GID deleted.

Test 53 – CMP-GID-06: GID Content Version Processing

Test Case #:	CMP-GID-06	
Test Description:	To verify that if the content version changes for a previously received Intersection GID that the updated content is stored in NVM.	
Test Setup / Equipment:	See setup steps above.	
Test Steps		**Expected Results for Corresponding Test Step**
1) Clear GID Database 2) Send GID 3) Send updated content version GID		CDIP Log has entry for two New GID received.

Test 54 – CMP-GID-07: Expired GID Deletion

Test Case #: CMP-GID-07	
Test Description:	To verify that, at start-up, stored Intersection GID records older than the GID expiration period are deleted.
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
1) Send a GID that will expire shortly 2) Restart CICAS Application	CDIP Log has entry for Expired GID record deleted.

Test 55 – CMP-GID-08: GID Load Time Updates

Test Case #: CMP-GID-08	
Test Description:	To verify that the load time for stored Intersection GIDs is updated whenever the GID is received and that this is used in the expired GID determination
Test Setup / Equipment:	See setup steps above. Configure GIDExpirationTime to be 1 day
Test Steps	**Expected Results for Corresponding Test Step**
1) Send GID 1 2) Send GID 2 3) Send GID 1 Again 4) Wait one day	1) Verify GID 1 has been received and stored 2) Verify GID 2 has been received and stored 3) Verify that after a day has passed that GID 2 is deleted before GID 1. If so that would indicate that Load times are updated when GIDs are received.

Test 56 – CMP-GID-09: GID Expiration Period Configuration

Test Case #: CMP-GID-09	
Test Description:	To verify that the GID expiration period is configurable.
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
1) Have GIDs already in database. 2) Update configuration file to have expiration period be set at a small value. 3) Restart CICAS Application	CDIP Log has entry for Expired GID record deleted.

Test 57 – CMP-GID-10: New GID Received, Oldest GID Deleted

Test Case #: CMP-GID-10	
Test Description:	To verify that, if a new Intersection GID is received and there is not enough remaining NVM to store the GID, the oldest GID in NVM is deleted and the new GID is stored. NOTE: The list of Intersection Reference points needs to be updated as well.
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
1) Configure the GID storage allocation. 2) Send multiple GIDs (all with unique Intersection ID) to fill up storage space. 3) Clear log file. 4) Send new GID (unique Intersection ID)	CDIP Log has entry for Unexpired GID record deleted.

Test 58 – CMP-GID-11: Area GID Processing

Test Case #: CMP-GID-11	
Test Description:	To verify that an Area GID is parsed and several GIDs are entered into the database
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
1) Clear GID Database. 2) Start CICAS application. 3) Send Area GID	CDIP Log has entries for New GIDs received that match all GIDs within area.

Test 59 – CMP-GID-12: GID Metric Object Processing

Test Case #: CMP-GID-12	
Test Description:	To verify that, if a Metric Object is in a GID the elapsed time is sent to the DAS.
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
Send GID with Metric Object	DAS records have data on the elapsed time for sending and receiving the GID as well as a message counter.

~~Test 60 – CMP-GID-13: GID Handler Heartbeat Processing~~

~~**Test Case #:**~~ ~~CMP-GID-13~~	
~~**Test Description:**~~	~~Successful heartbeat messages are sent out to error handler~~
~~**Test Setup / Equipment:**~~	~~See setup steps above.~~
~~**Test Steps**~~	~~**Expected Results for Corresponding Test Step**~~
~~Start CICAS application.~~	~~Check log file to see if the Error Handler is receiving heartbeat messages.~~

NOTE: This is being tested as part of SW Watchdog Monitor test cases.

~~Test 61 – CMP-GID-14: GID Heartbeat Error~~

~~**Test Case #:**~~ ~~CMP-GID-14~~	
~~**Test Description:**~~	~~Configuration of heartbeat messages to create error~~
~~**Test Setup / Equipment:**~~	~~See setup steps above.~~
~~**Test Steps**~~	~~**Expected Results for Corresponding Test Step**~~
~~Start CICAS application.~~	~~Check log file to see if the Error Handler is logging an error in the GID DB heartbeat.~~

NOTE: This is being tested as part of SW Watchdog Monitor test cases.

Test 62 – CMP-GID-15: Intersection Reference Point Updates

Test Case #: CMP-GID-15	
Test Description:	To verify that when a new Intersection GID is received that the list of Intersection Reference Points is updated
Test Setup / Equipment:	See setup steps above.
Test Steps	**Expected Results for Corresponding Test Step**
1) Delete all GIDs in storage	
1) Drive the loop of local intersections	1) GIDs are received and stored (check via the CDIGidRxLog and CDINewGidRx logs) 2) New Intersection Reference Point is sent to IILM (check via Intersection being identified as it is approached or via IILM log if it exists.)

Test 63 – CMP-GID-16: Intersection Reference Point List via Stored GIDs

Test Case #:	CMP-GID-16	
Test Description:	To verify that the correct list of Intersection Reference Points is determined from the stored Intersection GIDs	
Test Setup / Equipment:	See setup steps above.	

Test Steps	**Expected Results for Corresponding Test Step**
1) Drive the loop of local intersections	1) GIDs are received and stored
2) Disable GID transmission from the RSEs	
3) Turn off vehicle and then re-start	1) GIDs are stored on OBE (check via CDIStartupGIDDBConts log)
4) Drive by each of the intersections w/ disable GID transmission	1) Intersection is identified as it is approached

A.33.1.4 SPaT Handler (SPaT)

A.33.1.4.1 Setup

CICAS-V SPaT Test Setup and Instructions

Hardware Required:

1 – DENSO WSU preloaded with the latest revision of the CICAS-V software buil**d**

1 – WSU DSRC antenna and cabling

1 – WSU GPS unit (NovAtel OEMV or similar) and cabling

1 – WSU GPS antenna and cabling

1 – RSE preloaded with the latest revision of the CICAS-V SPaT server software build

1 – RSE GPS antenna with cabling

1 – 12V power supply (5A minimum)

Optional Equipment:

1 – Eagle EPAC 3108 Model 52 traffic controller preloaded with custom CICAS-V firmware

1 – Crossover Ethernet cable (connection of RSE to traffic controller)

Hardware Preparation:

1. Begin connecting all appropriate power and communication cabling to the devices.
2. Power all devices and verify normal operation.
3. If a traffic controller is available, perform steps 4 – 7. Else skip to step 8.
4. Set the EPAC internal IP address to 192.168.0.2 netmask 255.255.255.0 gateway 192.168.0.1 and reboot the unit to ensure the settings have taken effect.
    a. See EPAC manual for network setup instructions.
5. Using a crossover type Ethernet cable, connect the EPAC to an available Ethernet port on the RSE.
6. Set the IP address of the appropriate LAN port on the RSE to 192.168.0.3 netmask 255.255.255.0 gateway 192.168.0.1 and activate the connection.
7. If the connection from the RSE to the EPAC is functional, the EPAC should respond to a ping request from the RSE using the command **ping 192.168.0.2.**
8. For proper channel switching operation from both the RSE and WSU, verify that all GPS equipment is properly configured and connected and the antennas have a clear view of the sky for clock synchronization between the RSE and WSU.

9. If available, use a test utility such as DENSO WTA to verify that a connection is established with GPS and the appropriate data is being received.
10. Execute the SPaT server software on the RSE and set the default transmission rate to 10Hz.
11. Assuming the server and client software running on the RSE and WSU configure the DSRC radios to the proper channel, data rate, and power level, the equipment should now be ready to test.
12. Test Cases
13. Test 64 – CMP-SPaT-01: Verify DSRC Link and Basic SPaT Message Reception
14. Using one of many methods, verify that the DSRC link between the RSE and WSU is active and fully operational.
15. This can be accomplished using a WSU application such a WDA designed to test basic radio operation. Static SPaT frames could also broadcast from the RSE using a test application such as "sling." Note that the EPAC traffic controller is not necessary to complete this step.
16. Using a SPaT message parsing application such as radiodemux, verify that the SPaT messages are being received at the rate of broadcast.
17. Check timestamp intervals and verify messages are being received and parsed periodically. For example, for a 10Hz RSE broadcast rate, messages should be available to the WSU application at roughly 100ms intervals +/- some small error.
18. Check range of broadcast rates from 1 – 20Hz and verify successful reception and proper timestamps.
19. Verify no CRC or checksum errors exist.
20. Suggestion: Forcefully change CRC calculation on the RSE to create error and verify WSU detects an improper CRC.
21. Other suggestions…
22. Change SPaT PSID and verify WSU no longer parses.
23. Change TOM identifier (0xF1) and verify WSU no longer parses.

Test 65 – CMP-SPaT-02: Verify Correct Operation of WSU SPaT Parser (static frames)

24. Using a WSU application such as radiodemux, verify the correct parsing of static SPaT broadcast messages from the RSE.
25. Begin WSU SPaT parser testing using static SPaT frames to verify basic reception and parsing. Note that and EPAC traffic controller is not necessary when using

static SPaT frames. A test application such as “sling” can be used to broadcast static SPaT frames at various broadcast rates.

26. Verify parsed SPaT data correctly matches information contained within static frame. Includes the following:
27. Layer Type
28. Layer ID
29. Content Version (should not be changing)
30. Format Version
31. Intersection ID
32. Approach ID’s (all)
33. Signal Phase Indications
34. Time Confidence
35. Signal Phase Change Time
36. Yellow Duration
37. Create additional static frames that verify the correct parsing at the extents of the useable range for each SPaT parameter.
38. Create additional static frames that verify correct parser operation at the extents of the data range for each SPaT parameter.
39. Verify no CRC/Checksum errors exist throughout the static frame tests (also verifies correct SPaT server operation on the RSE).
40. Verify correct reception intervals based on the RSE broadcast rate using internal WSU timestamp.
41.
42. Test 66 – CMP-SPaT-03: Verify Correction Operation of WSU SPaT Parser (live frames)
43. Using WSU application such as radiodemux, verify the correct parsing of live SPaT broadcast messages from the RSE:
44. Parsing live SPaT frames requires that a traffic controller (or emulator) be attached to the RSE running the SPaT server application.
45. Verify Content Version is incrementing for each updated SPaT message (assuming the SPaT content has actually changed).
46. Using WSU logging application, save SPaT data log of complete traffic cycle to Compact Flash and verify correct phase and timer operation.
47. Verify all approaches for the current intersection are represented in the data log .

48. Verify correct start and stop time of phase time counters through complete phase cycle (ex. 4.3 4.2 4.1 4.0 3.9……0.2 0.1 0.0).
49. Verify correct phase color for the current approach.
50. Verify no CRC/Checksum errors exist throughout the live frame tests (also verifies correct SPaT server operation on the RSE).
51. Verify correct reception intervals based on the RSE broadcast rate using internal WSU timestamp.
52.
53. Test 67 – CMP-SPaT-04: Verify SPaT is Parsed and Available to Applications in a Timely Manner
54. Using WSU logging application, save SPaT data log of complete traffic cycle to Compact Flash and verify correct phase and timer operation.
55. Verify broadcast GPS timestamp contained within the SPaT message closely matches the internal timestamp provided by the WSU logger.
56. Should be << 100ms
57. Verify throughout the traffic cycle, paying particularly close attention to the timestamps at the onset of yellow and red phase changes.
58. Using a video camera overlaid with GPS timestamp information, capture video data for each intersection approach.
59. Compare phase change GPS times extracted from the video data with the time stamped SPaT data collected by the WSU. The delta between the actual phase change time recorded in the video and the phase change time reported by the WSU should no more than 50ms.

A.33.1.5 DAS Handler / Logger (DASH)

A.33.1.5.1 Test Cases

In order to catch failures that may be related to buffer overflows, processing limitations, or currently unknown factors, the following tests should be run under each of these test conditions:

Use Case Number	Test vehicle speed (sl = speed Limit)	Warning received from WA	Signalized or Stop Sign
1	0 mph (stopped)	n/a	n/a
2	< WA threshold	No	Stop Sign
3	< WA threshold	No	Signalized
4	sl-5 up to sl	No	Stop Sign
5	sl-5 up to sl	No	Signalized
6	sl-5 up to sl	Yes	Stop Sign
7	sl-5 up to sl	Yes	Signalized

Test 68 – CMP-DASH-01: CAN Message $701 Registration

Test Case #: CMP- DASH -01	
Test Description:	Register with the WSU VIS to receive CAN message $701
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Connect a DAS unit to CAN 2 of the WSU (two wire high-speed CAN bus operating at 500 kbps, standard [11 bit message identifier] format). 2) Connect a CAN monitoring device to CAN 2 of the WSU (two wire high-speed CAN bus operating at 500 kbps, standard [11 bit message identifier] format). 3) Configure the monitoring device to record or display incoming data. 4) Run the CICAS-V software, utilizing inputs from the Netway6 CAN box and the GPS receiver.	1) Verify that the monitoring device receives CAN message $701, which is issued by the DAS in response to receiving a CAN message $606.

Test 69 – CMP-DASH-02: CAN $606 OBE-DAS Heartbeat Transmission

Test Case #: CMP- DASH -02	
Test Description:	Upon start-up, initialize the OBE to DAS Heartbeat Sequence counter to 1, and increment the counter by 1 for each subsequent transmission of CAN message $606 to the DAS
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Connect a DAS unit to CAN 2 of the WSU (two wire high-speed CAN bus operating at 500 kbps, standard [11 bit message identifier] format). 2) Connect a CAN monitoring device to CAN 2 of the WSU (two wire high-speed CAN bus operating at 500 kbps, standard [11 bit message identifier] format). 3) Configure the monitoring device to record or display incoming data, and begin recording prior to running the CICAS-V software. 4) Run the CICAS-V software, utilizing inputs from the Netway6 CAN box and the GPS receiver.	1) Examine bytes 5, 6, and 7 of the first monitored or recorded CAN message $606. It should have a value of 1 in the 24 bits made up of bytes 5, 6 and 7. 2) Examine bytes 5, 6, and 7 of a series of monitored or recorded CAN message $606. Each successive set of bytes should be 1 greater than the previous set of bytes.

Test 70 – CMP-DASH-03: CAN $701 DAS-OBE Heartbeat Processing

Test Case #: CMP- DASH -03	
Test Description:	Upon receipt of CAN message $701: (a) Check for an OBE to DAS heartbeat error indication; also check if the DAS heartbeat sequence counter matches the last number output in the $606 message; (b) Check for any other error indication; (c) Upon a heartbeat error or DAS error, output an error indication to the Error Handler.
Test Setup / Equipment:	

Test Steps	**Expected Results for Corresponding Test Step**
1) Connect a CAN device to CAN 2 of the WSU (two wire high-speed CAN bus operating at 500 kbps, standard [11 bit message identifier] format). 2) Run the CICAS-V software, utilizing inputs from the Netway6 CAN box and the GPS receiver. 3) Configure/program the connected CAN device to respond to a CAN message $606 with a proper CAN message $701. 4) Configure/program the CAN simulator to respond to a CAN message $606 with a CAN message $701 having combinations of the following: (1) incorrect heartbeat sequence counter; (2) DAS System Error; (3) DAS Boot Up Error; (4) DAS Shutdown Error; and (5) OBE to DAS Heartbeat Error. 5) Repeat Test Step 4 as required.	1) Verify that the CAN simulator responds to each CAN message $606 by sending a CAN message $701 with the 4 error bits indicating Healthy, and the DAS Heartbeat Sequence in bytes 5, 6 and 7. 2) Confirm that the Error Handler properly logs each of the errors.

Test 71 – CMP-DASH-04: OBE-DAS Data Frame Composition

Test Case #: CMP-DASH-04	
Test Description:	Receive input data from other modules.
Test Setup / Equipment:	CAN recording or monitoring device
Test Steps	**Expected Results for Corresponding Test Step**
1) Connect a CAN monitoring device to CAN 2 of the WSU (two wire high-speed CAN bus operating at 500 kbps, standard [11 bit message identifier] format). 2) Configure the monitoring device to record or display incoming data. 3) Run the CICAS-V software, utilizing inputs from the Netway6 CAN box and the GPS receiver. NOTE: For Use Case #1 (vehicle stopped), repeat this test by changing the turn signal position, the brake usage, and or by stopping at another location. This will allow the tester to observe changes in the data (Left Turn Signal, Right Turn Signal, Brakes Active, and Vehicle Latitude).	1) Verify that the monitoring device has captured CAN messages. Valid messages will have the following hexadecimal message identifiers: $600, $601, $602, $603, $604, $605, $606, $610, $611, $612, $613, $615, $615, $616, $617, $618, $650, $701 and $751 2) Spot check that: (a) bit 6 of byte 0 of message $600 correctly reflects the “Brakes Active” status; (b) bytes 3 and 4 of message $601 correctly reflect the vehicle speed; (c) bits 3 and 2 of byte 2 of message $602 correctly reflects the “Left Turn Signal” and “Right Turn Signal”, respectively; and (d) bytes 0 through 3 of message $614 correctly reflect the “Vehicle Latitude”

Test 72 – CMP-DASH-05: OBE-DAS $751 Event Triggered Transmission

Test Case #: CMP- DASH -05	
Test Description:	If the data must be output to the DAS: Check if the data is for DAS message $751. If so, output the message immediately
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Insure that RSE equipment is broadcasting Metric Object for all CICAS services. 2) Connect a CAN monitoring device to CAN 2 of the WSU (two wire high-speed CAN bus operating at 500 kbps, standard [11 bit message identifier] format). 3) Configure the monitoring device to record or display incoming data with a timestamp of when it was received. 4) Run the CICAS-V software within DSRC range, utilizing inputs from the Netway6 CAN box and the GPS receiver.	1) Monitor that the event based CAN message $751 can be received independently of the 10 Hz timing interval of the other DAS CAN messages.

Test 73 – CMP-DASH-06: OBE-DAS Periodic Data Transmission

Test Case #:	CMP- DASH -06
Test Description:	If the data must be output to the DAS: If the data is not for message $751, buffer the data. Output DAS messages $600-$606, $610-$618 and $650 at the next 10 Hz interval.
Test Setup / Equipment:	CAN recording or monitoring device
Test Steps	**Expected Results for Corresponding Test Step**
1) Connect a CAN monitoring device to CAN 2 of the WSU (two wire high-speed CAN bus operating at 500 kbps, standard [11 bit message identifier] format). 2) Configure the monitoring device to record or display incoming data with a timestamp of when it was received. 3) Run the CICAS-V software, utilizing inputs from the Netway6 CAN box and the GPS receiver.	1) Verify that the monitoring device has captured bursts of CAN messages. Valid messages will have the following hexadecimal message identifiers: $600 to $606, $610 to $618, $650, $701 and $751. 2) Verify that the timestamp indicates the bursts are timed 100 msec apart. 3) Spot check that: (a) bit 6 of byte 0 of message $600 correctly reflects the “Brakes Active” status; (b) bytes 3 and 4 of message $601 correctly reflect the vehicle speed; (c) bits 3 and 2 of byte 2 of message $602 correctly reflects the “Left Turn Signal” and “Right Turn Signal”, respectively; and (d) bytes 0 through 3 of message $614 correctly reflect the “Vehicle Latitude”

Test 74 – CMP-DASH-07: Data Logging

Test Case #:	CMP- DASH -07
Test Description:	If the data must be logged (enabled by a user configurable log mask), write a record to the log file.
Test Setup / Equipment:	CAN recording or monitoring device
Test Steps	**Expected Results for Corresponding Test Step**
1) Connect a CAN monitoring device to CAN 2 of the WSU (two wire high-speed CAN bus operating at 500 kbps, standard [11 bit message identifier] format). 2) Configure the monitoring device to record or display incoming data with a timestamp of when it was received. 3) Select type of data to log, and enable its logging by selecting it in the cicas-v.conf file. 4) Run the CICAS-V software, utilizing inputs from the Netway6 CAN box and the GPS receiver. 5) Disable logging of DAS data on the WSU. 6) Run the CICAS-V software, utilizing inputs from the Netway6 CAN box and the GPS receiver.	1) Verify that correct records (records containing the data selected in step 3) are written to the log file when logging is enabled, and that no records are written when logging is not enabled.

~~Test 75 – CMP-DASH-08: Heartbeat Transmission to Error Handler~~

~~**Test Case #:**~~ ~~CMP- DASH -08~~	
~~**Test Description:**~~	~~Output a periodic software heartbeat message to the Error Handler at a configurable interval.~~
~~**Test Setup / Equipment:**~~	
~~**Test Steps**~~	~~**Expected Results for Corresponding Test Step**~~
~~1) Connect a DAS unit to CAN 2 of the WSU (two-wire high-speed CAN bus operating at 500 kbps, standard [11 bit message identifier] format).~~ ~~2) Run the CICAS-V software, utilizing inputs from the Netway6 CAN box and the GPS receiver.~~ ~~3) Stop the CICAS-V software and change the periodicity of the heartbeat message.~~ ~~4) Restart the CICAS-V software.~~ ~~5) If there is no way to see (via an IPC call, etc.) whether or not the software periodic heartbeat message is being received, it would be necessary to modify the Error Handler code to provide feedback upon the arrival of software heartbeat messages.~~	~~1) Verify that the software periodic heartbeat message is being sent to the Error Handler, by no watchdog timer expiration errors being reported by the Error Handler.~~ ~~2) Verify that the software periodic heartbeat message is being sent to the Error Handler at the correct periodicity.~~

NOTE: The SW as implemented does not support this test. This functionality can be confirmed via other tests.

A.33.1.6 Intersection Identification (IID)

A.33.1.6.1 Setup

OBE with latest software and configuration values set to default except:

- Distance to antenna value set to appropriate value for vehicle
- “OffGID” value set to 5 meters

A.33.1.6.2 Test Cases

Test 76 – CMP-IID-01: Directional Intersection Approach Test

Test Case #: CMP-IID-01	
Test Description:	Directional intersection approach test
Test Setup / Equipment:	Approach intersection from each of the individual roadways. For example, for 10 Mile and Orchard lake, approach from north, south, west, and east-bound.
Expected Results	The intersection the vehicle is approaching should be identified within 10-meters of the range specified in “GCDDistance”

Test 77 – CMP-IID-02: Distance from Intersection Identification Verification

Test Case #: CMP-IID-02	
Test Description:	"GCDDistance" effectiveness test.
Test Setup / Equipment:	Identify permanent physical object 200m and 300m away from the center of a test intersection using Google earth. Approach intersection from each of two approaches and verify intersection is identified with 10 meters of the physical object.
Expected Results	The intersection the vehicle is approaching should be identified within 10-meters of the range specified in "GCDDistance"

Test 78 – CMP-IID-03: Zero Speed Velocity Behavior

Test Case #: CMP-IID-03	
Test Description:	Verify zero speed velocity behavior
Test Setup / Equipment:	Approach intersection and stop and stop bar. Verify that intersection selection remains locked on until GPS antenna passes the stop bar.
Expected Results	The intersection selection should remain selected as nearby until GPS antenna passes the stop bar.

Test 79 – CMP-IID-04: Non-pre-loaded GID Selection

Test Case #: CMP-IID-04	
Test Description:	Verify GID selection without pre-loaded GID
Test Setup / Equipment:	Erase the GID database. Drive Detroit circular circuit of all three intersections from northbound Orchard Lake Rd to eastbound Farmington Rd to Southbound Drake Rd.
Expected Results	The intersection the vehicle is approaching should be identified within 10-meters of the range specified in "GCDDistance" and stop being identified once the vehicle passes the center of the intersection.

Test 80 – CMP-IID-05: Pre-loaded GID Selection

Test Case #: CMP-IID-05	
Test Description:	Verify GID selection with pre-loaded GID
Test Setup / Equipment:	GID database containing the three Detroit GIDs. Drive Detroit circular circuit of all three intersections from northbound Orchard Lake Rd to eastbound Farmington Rd to Southbound Drake Rd.
Expected Results	The intersection the vehicle is approaching should be identified within 10-meters of the range specified in "GCDDistance". Verify that "no intersection" is reported once the vehicle passes the center of the intersection.

A.33.1.7 Map Matching / Lane Identification (LID)

A.33.1.7.1 Setup

OBE with latest software and configuration values set to default except:

- Distance to antenna value set to appropriate value for vehicle
- "OffGID" value set to 5 meters

A.33.1.7.2 Test Cases

Test 81 – CMP-LID-01: Quick Lane Shifts

Test Case #:	CMP-LID-01
Test Description:	Quick lane shift test
Test Setup / Equipment:	Conduct at least one left to right lane shift for each intersection approach (not GID approach) at 10 Mile and Orchard Lake Rd; and 12 Mile and Farmington Road. Lane shift should take approximately 2-seconds. This test should be conducted when GPS coverage is predicted to be poor.
Expected Results	Lane identified should change to the appropriate. Confidence should be >80% within 200mS of steady state in new lane.

Test 82 – CMP-LID-02: Multiple Lane Shifts

Test Case #:	CMP-LID-02
Test Description:	Multiple Lane Shifts
Test Setup / Equipment:	Conduct a lane shift from rightmost lane to left turn lane. Should be completed at VTTI. Lane shift should take approximately 4-seconds.
Expected Results	Lane identified should change to the appropriate. Confidence should be >80% within 200mS of steady state in new lane.

Test 83 – CMP-LID-03: Early Lane Shift

Test Case #:	CMP-LID-03
Test Description:	Early lane shift test
Test Setup / Equipment:	Conduct a left to right lane shift as early as possible from lane 2 to lane 3 while driving northbound at 10 Mile and Orchard Lake Rd. Conduct a right to left lane shift from lane 1 to the left turn lane as early as possible while driving west bound at 10 Mile Rd and Orchard Lake Rd. Lane shift should take approximately 3-seconds.
Expected Results	Lane identified should change to the appropriate. Confidence should be >80% within 200mS of steady state in new lane.

Test 84 – CMP-LID-04: GPS Antenna Offset Verification

Test Case #: CMP-LID-04	
Test Description:	Check distance to stop bar with vehicle bumper at stop bar
Test Setup / Equipment:	Distance between GPS antenna and front bumper must be verified and entered in cicas-v.conf. Drive to intersection, stop at stop bar. Conduct test at left turn lane and in right turn lane.
Expected Results	Distance to stop bar is zero.

Test 85 – CMP-LID-05: Lane Segment Transition Test

Test Case #: CMP-LID-05	
Test Description:	Lane Segment Transition Distance Check
Test Setup / Equipment:	Drive complete segment from beginning to end. Recorded distance to stop bar value
Expected Results	Verify that transition between lane segments is smooth.

Test 86 – CMP-LID-06: Off Lane but On GID Threshold

Test Case #: CMP-LID-06	
Test Description:	Off Lane but on GID threshold' parameter
Test Setup / Equipment:	Set "OffGID" value in cicas-v.conf file to 5 meters. Drive northbound toward 10 Mile Rd and Orchard Lake Rd in rightmost lane. Turn into Walgreen's parking lot and continue north. Drive eastbound on 10 Mile Rd in rightmost lane, turn right into small parking lot paralleling 10 Mile Rd.
Expected Results	Lane matching should state that vehicle is at an intersection but off GID. Lane matching confidence should transition from a high value (>90%) to 0%.

A.33.1.8 Warning Algorithm (WARN)

A.33.1.8.1 Test Cases

Test 87 – CMP-WARN-01: SPaT Requirement for Warning

Test Case #: CMP-WARN-01	
Test Description:	Verify the necessity of valid SPaT data for a warning.
Test Setup / Equipment:	Direct access to input and output of Warning Algorithm.
Test Steps	**Expected Results for Corresponding Test Step**
1.) Provide WA with approach ID data.	N/A
2.) Withhold valid SPaT data (at least) for the given approach.	N/A

Test Case #: CMP-WARN-01		
Test Description:	Verify the necessity of valid SPaT data for a warning.	
3.) Check “warning-generated” output flag.		warning-generated == false

Test 88 – CMP-WARN-02: Approach ID Requirement for Warning

Test Case #: CMP-WARN-02	
Test Description:	Verify the necessity of approach ID data for a warning,
Test Setup / Equipment:	Direct access to input and output of Warning Algorithm.
Test Steps	**Expected Results for Corresponding Test Step**
1.) Provide WA with valid SPaT data for all approaches at intersection.	N/A
2.) Withhold approach ID data.	N/A
3.) Check “warning-generated” output flag.	warning-generated == false

Test 89 – CMP-WARN-03: WA Sleep Frequency

Test Case #: CMP-WARN-03	
Test Description:	Verify sleep frequency correctness.
Test Setup / Equipment:	Ability to add observable task into WA flow. (probably changing source and re-compiling)
Test Steps	**Expected Results for Corresponding Test Step**
1.) Instruct WA to perform some observable task (i.e., output to console) every time the algorithm awakes.	N/A
2.) Check console output frequency.	Verify correctly chosen operational frequency.

Test 90 – CMP-WARN-04: Equipped Logic at Stop Sign Intersection

Test Case #: CMP-WARN-04	
Test Description:	Verify correctness of equipped logic at stop sign intersection.
Test Setup / Equipment:	Direct access to input and output of Warning Algorithm.
Test Steps	**Expected Results for Corresponding Test Step**
1.) Provide WA with approach ID data.	N/A
2.) Check “intersection-in-range” output flag.	intersection-in-range == true

Test 91 – CMP-WARN-05: Equipped Logic at Signalized Intersection

Test Case #:	CMP-WARN-05	
Test Description:	Verify correctness of equipped logic at signalized intersection.	
Test Setup / Equipment:	Direct access to input and output of Warning Algorithm.	

Test Steps	**Expected Results for Corresponding Test Step**
1.) Provide WA with approach ID data.	N/A
2.) Provide WA with valid SPaT data.	N/A
3.) Check "intersection-in-range" output flag.	intersection-in-range == true

Test 92 – CMP-WARN-06: Warning Logic Approach ID Input Verification

Test Case #:	CMP-WARN-06	
Test Description:	Assure correctness of warning generation logic based on approach ID data input.	
Test Setup / Equipment:	Direct access to input and output of Warning Algorithm.	

Test Steps	**Expected Results for Corresponding Test Step**
1.) Check "warning-generated" output flag.	Verify: warning-generated == true
2.) Withhold input of approach ID data.	N/A
3.) Check "warning-generated" output flag.	warning-generated == false

NOTE: This test requires access to WA at a level that is not available. The Functionality can be shown by other higher level tests.

Test 93 – CMP-WARN-07: Warning Logic SPaT Input Verification

Test Case #:	CMP-WARN-07	
Test Description:	Assure correctness of warning generation logic based on valid SPaT data input.	
Test Setup / Equipment:	Direct access to input and output of Warning Algorithm.	

Test Steps	**Expected Results for Corresponding Test Step**
1.) Check "warning-generated" output flag.	Verify: warning-generated == true
2.) Withhold input of valid SPaT data.	N/A
3.) Check "warning-generated" output flag.	warning-generated == false

NOTE: This test requires access to WA at a level that is not available. The Functionality can be shown by other higher level tests.

Test 94 – CMP-WARN-08: Is Braking Logic Verification

Test Case #: CMP-WARN-08	
Test Description:	Assure correctness of isBraking logic.
Test Setup / Equipment:	Knowledge of internal algorithm variable values (most likely through logging).
Test Steps	**Expected Results for Corresponding Test Step**
1.) Check output of isBraking logic.	Verify isBraking == false
2.) Provide braking input so that: press time > configurable time OR intended braking value > configurable value OR vehicle speed < configurable speed	N/A
3.) Check output of isBraking logic.	isBraking == true

Test 95 – CMP-WARN-09: Is Braking Transition

Test Case #: CMP-WARN-09	
Test Description:	Assure correct effects of isBraking transition.
Test Setup / Equipment:	Direct access to input and output of Warning Algorithm. Knowledge of internal algorithm variable values (most likely through logging).
Test Steps	**Expected Results for Corresponding Test Step**
1.) Check output of "warning-generated" output flag.	warning-generated == true
2.) Follow steps described in CMP-WARN-08.	N/A
3.) Check output of "warning-generated" output flag.	warning-generated == false

~~Test 96 – CMP-WARN-10: Time to Red Logic Verification~~

~~**Test Case #:** CMP-WARN-10~~	
~~**Test Description:**~~	~~Verify correctness of timeToRed logic.~~
~~**Test Setup / Equipment:**~~	~~Direct access to input and output of Warning Algorithm.~~ ~~Knowledge of internal algorithm variable values (most likely through logging).~~
~~**Test Steps**~~	~~**Expected Results for Corresponding Test Step**~~
~~1.) Withhold valid SPaT data or approach ID data.~~	~~N/A~~
~~2.) Check timeToRed value.~~	~~Value must be invalid.~~

NOTE: Without SPaT info, WA is not run and wa.log does not include timeToRed info. The Functionality can be shown by other higher level tests.

Test 97 – CMP-WARN-11: Time to Stop Bar Verification

Test Case #: CMP-WARN-11	
Test Description:	Verify correctness of timeToStopBar logic.
Test Setup / Equipment:	Direct access to input and output of Warning Algorithm. Knowledge of internal algorithm variable values (most likely through logging).
Test Steps	**Expected Results for Corresponding Test Step**
1.) Withhold distanceToStopBar from WA.	N/A
2.) Check timeToStopBar value.	Value must be invalid.

NOTE: Without distanceToStopBar info, WA is not run and wa.log does not include timeToStopBar info. The Functionality can be shown by other higher level tests.

Test 98 – CMP-WARN-12: Time Comparison Verification

Test Case #: CMP-WARN-12	
Test Description:	Verify correctness of time comparison logic.
Test Setup / Equipment:	Knowledge of Warning Algorithm output values. Knowledge of internal algorithm variable values (most likely through logging).
Test Steps	**Expected Results for Corresponding Test Step**
1.) Modify timeToRed and timeToStopBar values (via whatever means available) so that timeToRed >= timeToStopBar.	N/A
2.) Check “warning-generated” output flag.	warning-generated == false

NOTE: This functionality exercised in INT-SI-08/09/10.

Test 99 – CMP-WARN-13: Distance Comparison Verification

Test Case #: CMP-WARN-13	
Test Description:	Verify correctness of distance comparison logic.
Test Setup / Equipment:	Knowledge of Warning Algorithm output values. Knowledge of internal algorithm variable values (most likely through logging).
Test Steps	**Expected Results for Corresponding Test Step**
1.) Modify distanceToStopBar and distanceToWarn (distanceToWarn comes from VTTI table) values (via whatever means available) so that distanceToStopBar >= distanceToWarn.	N/A
2.) Check “warning-generated” output flag.	warning-generated == false

NOTE: This functionality exercised in INT-SI-08/09/10.

Test 100 – CMP-WARN-14: Braking Exception Logic Verification

Test Case #: CMP-WARN-14	
Test Description:	Verify correctness of braking exception logic.
Test Setup / Equipment:	Knowledge of Warning Algorithm output values. Knowledge of internal algorithm variable values (most likely through logging).
Test Steps	**Expected Results for Corresponding Test Step**
This tests the internal logic of the OBE and requires special tools and set up. In addition the data analyzed are internal SW variables and states that on their own provide no real benefit to the external reader. Thus the test steps and expected results are not provided. The test results, however, will still be documented.	

Test 101 – CMP-WARN-15: Minimum Warning Distance Threshold

Test Case #: CMP-WARN-15	
Test Description:	Verify that warnings are not given below minimum distance threshold.
Test Setup / Equipment:	Knowledge of Warning Algorithm output values. Knowledge of internal algorithm variable values (most likely through logging).
Test Steps	**Expected Results for Corresponding Test Step**
This tests the internal logic of the OBE and requires special tools and set up. In addition the data analyzed are internal SW variables and states that on their own provide no real benefit to the external reader. Thus the test steps and expected results are not provided. The test results, however, will still be documented.	

Test 102 – CMP-WARN-16: SPaT Freewheeling Logic

Test Case #: CMP-WARN-16	
Test Description:	Verify SPaT freewheeling logic
Test Setup / Equipment:	Modified SPaT server to send SPaT at 2 Hz (NOTE: Default configuration allows for missing up to 4 SPaT messages before terminating WA logic.) A way to monitor time to next phase, current signal phase, WA state information
Test Steps	**Expected Results for Corresponding Test Step**
1) Approach intersection with a SPaT server transmitting at 2 Hz.	1) The Current Signal Phase remains consistent with the approach / lane traffic controller 2) The Time to Next Phase counts down un-interrupted 3) The WA state remains consistent with what is expected

A.33.1.9 DVI Notifier (DVI)

A.33.1.9.1 Test Cases

Test 103 – CMP-DVI-01: State Machine Input/Output Verification I

Test Case #:	**CMP-DVI-1**
Test Description:	**S**tate **M**achine Test: Input/Output Verification
Test Setup / Equipment:	Direct access to input and output of State Machine.
Test Steps	**Expected Results for Corresponding Test Step**
Provide the following inputs: "intersection-in-range" = false "warning-generated" = false	Verify the following output: DVI Icon == none

Test 104 – CMP-DVI-02: State Machine Input/Output Verification II

Test Case #:	**CMP-DVI-2**
Test Description:	SM Test: Input/Output Verification
Test Setup / Equipment:	Direct access to input and output of State Machine.
Test Steps	**Expected Results for Corresponding Test Step**
Provide the following inputs: "intersection-in-range" = true "warning-generated" = false	Verify the following output: DVI Icon == solid blue

Test 105 – CMP-DVI-03: State Machine Input/Output Verification III

Test Case #:	**CMP-DVI-3**
Test Description:	SM Test: Input/Output Verification
Test Setup / Equipment:	Direct access to input and output of State Machine.
Test Steps	**Expected Results for Corresponding Test Step**
Provide the following inputs: “intersection-in-range” = true “warning-generated” = true	Verify the following output: DVI Icon == flashing red (OR solid blue/none if brakes are pressed before pre-warn -> warn transition)

Test 106 – CMP-DVI-04: State Machine Input/Output Verification IV

Test Case #:	**CMP-DVI-4**
Test Description:	SM Test: Input/Output Verification (this test is valuable and unique in respect to CMP-DVI-3 because it shows that logic is correct and even when no intersection in range, if a warning is generated that takes priority)
Test Setup / Equipment:	Direct access to input and output of Warning Algorithm.
Test Steps	**Expected Results for Corresponding Test Step**
Provide the following inputs: “intersection-in-range” = false “warning-generated” = true	Verify the following output: DVI Icon == flashing red

Test 107 – CMP-DVI-05: Flexible Trigger Warning Message Verification

Test Case #:	CMP-DVI-5
Test Description:	Verify correctness of Flexible Trigger message through logs (or scope/logic analyzer/CANalyzer/etc?).
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
This tests the internal logic of the OBE and requires special tools and set up. In addition the data analyzed are internal SW variables and states that on their own provide no real benefit to the external reader. Thus the test steps and expected results are not provided. The test results, however, will still be documented.	

Test 108 – CMP-DVI-06: Pre-Warning State Transitions

Test Case #: CMP-DVI-6	
Test Description:	Verify transitions out of pre-warn state (possible transitions: timer based to warn state or brake status based to equipped or none).
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
This tests the internal logic of the OBE and requires special tools and set up. In addition the data analyzed are internal SW variables and states that on their own provide no real benefit to the external reader. Thus the test steps and expected results are not provided. The test results, however, will still be documented.	

A.33.1.10 Power Moding (PWR)

A.33.1.10.1 Test Cases

Test 109 – CMP-PWR-01: Power Cycle Disk Resiliency

Test Case #: CMP-PWR-1	
Test Description:	Power Cycle Disk Structure Resiliency
Test Setup / Equipment:	WSU must be connected to Batt+, GND and IGN signal. Tester must have ability to change state of IGN signal.
Test Steps	**Expected Results for Corresponding Test Step**
1) The IGN signal should be high	1) The WSU is operating normally
2) Bring IGN low and then high again before WSU has been able to fully shut down	2) The WSU should shutdown and re-start without any harm coming to the file system (no check disk process run) on the internal storage

Test 110 – CMP-PWR-02: Correct Power Mode Choice

Test Case #: CMP-PWR-2	
Test Description:	Correct Power Mode Choice
Test Setup / Equipment:	WSU must be connected to Batt+, GND and IGN signal. Tester must have ability to change state of IGN signal.
Test Steps	**Expected Results for Corresponding Test Step**
1) The IGN signal should be low	1) The WSU should be off
2) Bring IGN high and then low again before WSU has been able to fully boot	1) The WSU should boot fully and automatically shutdown upon completion. (Also acceptable is for the WSU to shutdown before the full but process. What is not acceptable is for the WSU to remain on indefinitely while the IGN signal is low.)

A.33.1.11 SW Watchdog Monitor (WDG)

A.33.1.11.1 Test Cases

Test 111 – CMP-WDG-01: Application Heartbeat Message Transmission

Test Case #: CMP-WDG-1	
Test Description:	Verify that Heartbeat messages are being transmitted from each of the applications
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Start up the WSU and let it run	
2) Check the log files	1) Verify that heartbeat messages are being logged every configurable heartbeat time period.

Test 112 – CMP-WDG-02: Missed Heartbeats WSU Reset

Test Case #: CMP-WDG-2	
Test Description:	Verify that missed heartbeats leads to WSU reset
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Start up the WSU and let it run	
2) Kill Vehicle Message Handler	1) Verify that the WSU Resets 2) Upon re-start verify that information is logged indicating reason for reset.
3) Repeat #2 for: GPS Handler	1) Verify that the WSU Resets 2) Upon re-start verify that information is logged indicating reason for reset.
4) Repeat #2 for: Radio Handler / Data Demux	1) Verify that the WSU Resets 2) Upon re-start verify that information is logged indicating reason for reset.
5) Repeat #2 for: GID DB Handler	1) Verify that the WSU Resets 2) Upon re-start verify that information is logged indicating reason for reset.
6) Repeat #2 for: SPaT Handler	1) Verify that the WSU Resets 2) Upon re-start verify that information is logged indicating reason for reset.
7) Repeat #2 for: Intersection Identification	1) Verify that the WSU Resets 2) Upon re-start verify that information is logged indicating reason for reset.
8) Repeat #2 for: Map Matching / Lane Identification	1) Verify that the WSU Resets 2) Upon re-start verify that information is logged indicating reason for reset.
9) Repeat #2 for: Warning Algorithm	1) Verify that the WSU Resets 2) Upon re-start verify that information is logged indicating reason for reset.
10) Repeat #2 for: DVI Notifier	1) Verify that the WSU Resets 2) Upon re-start verify that information is logged indicating reason for reset.

Intersection Test Case Procedures

A.33.1.12 System Integration Procedures (SI)

The following tests should be run to verity that some combination of the RSE, OBE, Netway, DVI Icon / Audio, and Warning Algorithm table are all working properly in conjunction with one-another.

A.33.1.12.1 Test Cases

Test 113 – INT-SI-01: GID, GPSC, SPaT Tx / Rx

Test Case #: INT-SI-1	
Test Description:	Verify GID, GPSC, and SPaT message transmission / reception
Test Setup / Equipment:	Message reception logging on the OBE needs to be enabled.
Test Steps	**Expected Results for Corresponding Test Step**
1) Drive to the W. 10 Mile Rd. and Orchard Lake Rd. intersection	1) The GID, GPSC, and SPaT messages corresponding to this intersection are received by the OBE 2) The GID, GPSC, and SPaT message corresponding to this intersection are properly parsed by the OBE 3) The GID for the W. 11 Mile Rd. and Drake Rd. stop sign intersection is received and properly parsed by the OBE.
2) Drive to the W. 12 Mile Rd. and Farmington Rd. intersection	1) The GID, GPSC, and SPaT messages corresponding to this intersection are received by the OBE 2) The GID, GPSC, and SPaT message corresponding to this intersection are properly parsed by the OBE 3) The GID for the W. 11 Mile Rd. and Drake Rd. stop sign intersection is received and properly parsed by the OBE.

Test 114 – INT-SI-02: GID, GPSC, SPaT Tx Rate

Test Case #: INT-SI-2	
Test Description:	Verify GID, GPSC, and SPaT message transmission rate
Test Setup / Equipment:	Message reception logging on the OBE needs to be enabled
Test Steps	**Expected Results for Corresponding Test Step**
1) Drive to the W. 10 Mile Rd. and Orchard Lake Rd. intersection	1) Verify that the GID is received on the SCH at 2 Hz 2) Verify that the GPSC is received on the SCH at 1 Hz 3) Verify that the SPaT is received on the CCH at 10 Hz
2) Drive to the W. 12 Mile Rd. and Farmington Rd. intersection	1) Verify that the GID is received on the SCH at 2 Hz 2) Verify that the GPSC is received on the SCH at 1 Hz 3) Verify that the SPaT is received on the CCH at 10 Hz

Test 115 – INT-SI-03: W. 10 Mile Rd. & Orchard Lake Rd. Delta Position Verification

Test Case #: INT-SI-3	
Test Description:	W. 10 Mile Rd. and Orchard Lake Rd. corrected position delta from actual position verification
Test Setup / Equipment:	Ability to accurately measure vehicle position via L1/L2 or some other measurement GPS corrections have had time to take effect and settle out Ability to get vehicle corrected position Ability to get the delta between the actual and corrected position Refer to Figure 105 in this document for the intersection layout
Test Steps	**Expected Results for Corresponding Test Step**
1) Obtain delta position measurements for Approach #2	1) Delta distance should be within 0.5 meters
2) Repeat step #1 for Approach #4	1) Delta distance should be within 0.5 meters
3) Repeat step #1 for Approach #6	1) Delta distance should be within 0.5 meters
4) Repeat step #1 for Approach #8	1) Delta distance should be within 0.5 meters

Test 116 – INT-SI-04: W. 12 Mile Rd. & Farmington Rd. Delta Position Verification

Test Case #:	INT-SI-4
Test Description:	W. 12 Mile Rd. and Farmington Rd. corrected position delta from actual position verification
Test Setup / Equipment:	Ability to accurately measure vehicle position via L1/L2 or some other measurement GPS corrections have had time to take effect and settle out Ability to get vehicle corrected position Ability to get the delta between the actual and corrected position Refer to Figure 106 in this document for the intersection layout

Test Steps	**Expected Results for Corresponding Test Step**
1) Obtain delta position measurements for Approach #2	1) Delta distance should be within 0.5 meters
2) Repeat step #1 for Approach #4	1) Delta distance should be within 0.5 meters
3) Repeat step #1 for Approach #6	1) Delta distance should be within 0.5 meters
4) Repeat step #1 for Approach #8	1) Delta distance should be within 0.5 meters

Test 117 – INT-SI-05: W. 10 Mile Rd. & Orchard Lake Rd. Approach/Lane/Stop Bar Verification

Test Case #: INT-SI-5	
Test Description:	W. 10 Mile Rd. and Orchard Lake Rd. Approach / Lane / Stop bar identification / distance verification
Test Setup / Equipment:	GPS antenna offset from bumper needs to be properly configured. The ability to inspect live the selected approach and lane information. GPS corrections have had time to take effect and settle out Refer to Figure 105 in this document for a intersection layout Refer to Table 201 in this document for lane data

Test Steps	**Expected Results for Corresponding Test Step**
1) Drive the full length of Surveyed Lane S01 and stop at the stop bar.	1) **A4 L1** should begin being indicated close to the 'GID Ref. Pt. to Furthest Lane Node Distance' from the intersection reference point listed in the lane data table referenced in the 'Test Setup / Equipment' portion of this table 2) When the approach and lane is identified confirm that the distance to centerline is less than 1 m. 3) From where the lane and approach are identified up to the stop bar verify that there are no dis-continuities in the distance to stop bar 4) Distance to stop bar should indicate close to zero when stopped at the stop bar
2) Drive the full length of Surveyed Lane S02 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A4 L2**
3) Drive the full length of Surveyed Lane S03 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A4 L3**
4) Drive the full length of Surveyed Lane S04 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A4 L4**
5) Drive the full length of Surveyed Lane S05 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A6 L1**
6) Drive the full length of Surveyed Lane S06 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A6 L2**
7) Drive the full length of Surveyed Lane S07 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A6 L3**
8) Drive the full length of Surveyed Lane S08 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A6 L4**
9) Drive the full length of Surveyed Lane S09 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A8 L1**
10) Drive the full length of Surveyed Lane S10 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A8 L2**
11) Drive the full length of Surveyed Lane S11 and stop	1) Step #1 results for furthest lane node distance and

Test Case #: INT-SI-5	
Test Description:	W. 10 Mile Rd. and Orchard Lake Rd. Approach / Lane / Stop bar identification / distance verification
at the stop bar.	distance to stop bar for: **A8 L3**
12) Drive the full length of Surveyed Lane S12 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A2 L1**
13) Drive the full length of Surveyed Lane S13 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A2 L2**
14) Drive the full length of Surveyed Lane S14 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A2 L3**
15) Drive the full length of Surveyed Lane S15 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A2 L4**

Test 118 – INT-SI-06: W. 12 Mile Rd. & Farmington Rd. Approach/Lane/Stop Bar Verification

Test Case #: INT-SI-6	
Test Description:	W. 12 Mile Rd. and Farmington Rd. Approach / Lane / Stop bar identification / distance verification
Test Setup / Equipment:	GPS antenna offset from bumper needs to be properly configured. The ability to inspect live the selected approach and lane information. GPS corrections have had time to take effect and settle out Refer to Figure 106 in this document for a intersection layout Refer to Table 202 in this document for lane data

Test Steps	**Expected Results for Corresponding Test Step**
1) Drive the full length of Surveyed Lane S01 and stop at the stop bar.	1) **A5 L1** should begin being indicated close to the 'GID Ref. Pt. to Furthest Lane Node Distance' from the intersection reference point listed in the lane data table referenced in the 'Test Setup / Equipment' portion of this table 2) When the approach and lane is identified confirm that the distance to centerline is less than 1 m. 3) From where the lane and approach are identified up to the stop bar verify that there are no dis-continuities in the distance to stop bar 4) Distance to stop bar should indicate close to zero when stopped at the stop bar
2) Drive the full length of Surveyed Lane S02 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A2 L1**
3) Drive the full length of Surveyed Lane S03 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A2 L2**
4) Drive the full length of Surveyed Lane S04 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A3 L1**
5) Drive the full length of Surveyed Lane S05 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A8 L1**
6) Drive the full length of Surveyed Lane S06 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A8 L2**
7) Drive the full length of Surveyed Lane S07 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A1 L1**
8) Drive the full length of Surveyed Lane S08 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A6 L1**
9) Drive the full length of Surveyed Lane S09 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A6 L2**
10) Drive the full length of Surveyed Lane S10 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A6 L3**
11) Drive the full length of Surveyed Lane S11 and stop	1) Step #1 results for furthest lane node distance and

Test Case #: INT-SI-6	
Test Description:	W. 12 Mile Rd. and Farmington Rd. Approach / Lane / Stop bar identification / distance verification
at the stop bar.	distance to stop bar for: **A7 L1**
12) Drive the full length of Surveyed Lane S12 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A4 L1**
13) Drive the full length of Surveyed Lane S13 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A4 L2**

Test 119 – INT-SI-07: W. 11 Mile Rd. & Drake Rd. Approach/Lane/Stop Bar Verification

Test Case #: INT-SI-7	
Test Description:	W. 11 Mile Rd. and Drake Rd. Approach / Lane / Stop bar identification / distance verification
Test Setup / Equipment:	GPS antenna offset from bumper needs to be properly configured. The ability to inspect live the selected approach and lane information. Refer to Figure 107 in this document for a intersection layout Refer to Table 203 in this document for lane data
Test Steps	**Expected Results for Corresponding Test Step**
1) Drive the full length of Surveyed Lane S01 and stop at the stop bar.	1) **A4 L1** should begin being indicated close to the 'GID Ref. Pt. to Furthest Lane Node Distance' from the intersection reference point listed in the lane data table referenced in the 'Test Setup / Equipment' portion of this table 2) From where the lane and approach are identified up to the stop bar verify that there are no dis-continuities in the distance to stop bar 3) Distance to stop bar should indicate close to zero when stopped at the stop bar
2) Drive the full length of Surveyed Lane S02 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A6 L1**
3) Drive the full length of Surveyed Lane S03 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A6 L2**
4) Drive the full length of Surveyed Lane S04 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A8 L1**
5) Drive the full length of Surveyed Lane S05 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A2 L1**
6) Drive the full length of Surveyed Lane S06 and stop at the stop bar.	1) Step #1 results for furthest lane node distance and distance to stop bar for: **A2 L2**

Test 120 – INT-SI-08: W. 10 Mile Rd. & Orchard Lake Rd. Warning Distance Verification

Test Case #: INT-SI-8	
Test Description:	W. 10 Mile Rd. and Orchard Lake Rd. warning distance verification
Test Setup / Equipment:	GPS antenna offset from bumper needs to be properly configured. Functional DVI Notifier GPS corrections have had time to take effect and settle out Audio and Icon calibrated to come on at the same time (refer to INT-SI-12) Additional reaction time values all set to zero The ability to inspect live the selected approach and lane information. Refer to Figure 105 in this document.

Test Steps	**Expected Results for Corresponding Test Step**
1) Drive the posted speed limit down Approach #2 with the signal in such a state as to cause a warning as early as possible	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
2) Repeat step #1 for Approach #4	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
3) Repeat step #1 for Approach #6	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
4) Repeat step #1 for Approach #8	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.

Test 121 – INT-SI-09: W. 12 Mile Rd. & Farmington Rd. Warning Distance Verification

Test Case #:	INT-SI-9	
Test Description:	W. 12 Mile Rd. and Farmington Rd. warning distance verification	
Test Setup / Equipment:	GPS antenna offset from bumper needs to be properly configured. Functional DVI Notifier GPS corrections have had time to take effect and settle out Audio and Icon calibrated to come on at the same time (refer to INT-SI-12) Additional reaction time values all set to zero The ability to inspect live the selected approach and lane information. Refer to Figure 106 in this document.	

Test Steps	**Expected Results for Corresponding Test Step**
1) Drive the posted speed limit down Approach #1 with the signal in such a state as to cause a warning as early as possible	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
2) Repeat step #1 for Approach #2	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
3) Repeat step #1 for Approach #3	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
4) Repeat step #1 for Approach #4	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
5) Repeat step #1 for Approach #5	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
6) Repeat step #1 for Approach #6	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
7) Repeat step #1 for Approach #7	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
8) Repeat step #1 for Approach #8	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.

Test 122 – INT-SI-10: W. 11 Mile Rd. & Drake Rd. Warning Distance Verification

Test Case #: INT-SI-10	
Test Description:	W. 11 Mile Rd. and Drake Rd. warning distance verification
Test Setup / Equipment:	GPS antenna offset from bumper needs to be properly configured. Functional DVI Notifier Audio and Icon calibrated to come on at the same time (refer to INT-SI-12) Additional reaction time values all set to zero The ability to inspect live the selected approach and lane information. Refer to Figure 107 in this document.

Test Steps	**Expected Results for Corresponding Test Step**
1) Drive the posted speed limit down Approach #2 with the signal in such a state as to cause a warning as early as possible	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
2) Repeat step #1 for Approach #4	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
3) Repeat step #1 for Approach #6	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.
4) Repeat step #1 for Approach #8	1) DVI Icon and audio come on within corresponding +/- 200ms plus DVI calibration ms range of distance value in WA table.

Test 123 – INT-SI-11: Lane vs. Road Level Accuracy Verification

Test Case #: INT-SI-11	
Test Description:	Verify Lane vs. Road level accuracy requirement is supported
Test Setup / Equipment:	Intersection setup with the ability to disable DGPS message transmission from the RSE The ability to inspect live the selected approach and lane information.

Test Steps	**Expected Results for Corresponding Test Step**
1) At Orchard Lake Rd. and W. 10 Mile Rd (Road level accuracy required) ensure no correction messages are transmitted.	1) The intersection is identified 2) Lane / Approach matching is performed 3) Lane / Approach matching is successful
2) Re-enable correction messages at Orchard Lake Rd. and W. 10 Mile Rd.	1) The intersection is identified 2) Lane / Approach matching is performed 3) Lane / Approach matching is successful
3) Farmington Rd. and W. 12 Mile Rd (Lane level accuracy required) ensure no correction messages are transmitted.	1) The intersection is NOT identified 2) Lane / Approach matching is not performed 3) No warnings will be issued
4) Re-enable correction messages at Farmington Rd. and W. 12 Mile Rd.	1) The intersection is identified 2) Lane / Approach matching is performed 3) Lane / Approach matching is successful

Test 124 – INT-SI-12: DVI Icon / Audio Activation Timing

Test Case #: INT-SI-12	
Test Description:	DVI Icon / Audio activation timing
Test Setup / Equipment:	Support for DVI Icon and Audio The ability to determine activation timing for icon and audio

Test Steps	**Expected Results for Corresponding Test Step**
1) Drive toward any of the intersections such that a warning is provided	1) The DVI Icon and Audio come on at the same time.
2) If there is a difference in timing between the Icon and Audio make appropriate adjustments to the configuration data to account for this.	1) Document the timing difference
3) Repeat Steps 1 and 2 until there is no noticeable latency between the icon and audio	

Test 125 – INT-SI-13: DVI Icon / Audio / Brake Pulse Activation Timing

Test Case #: INT-SI-13	
Test Description:	DVI Icon / Audio / Brake Pulse activation timing
Test Setup / Equipment:	Support for DVI Icon, Audio, and Brake Pulse The ability to determine activation timing for icon, audio, and brake pulse
Test Steps	**Expected Results for Corresponding Test Step**
1) Drive toward any of the intersections such that a warning is provided	1) The DVI Icon, Audio, and Brake Pulse come on at the same time.
2) If there is a difference in timing between any of the Icon, Audio, or Brake Pulse make appropriate adjustments to the configuration data to account for this.	1) Document the timing difference
3) Repeat Steps 1 and 2 until there is no noticeable latency between the icon, audio, and brake pulse	

Test 126 – INT-SI-14: No Signalized IID when no SPaT

Test Case #: INT-SI-14	
Test Description:	Verify intersection is not identified when there is no SPaT at a signalized intersection
Test Setup / Equipment:	Intersection setup with the ability to disable SPaT transmission
Test Steps	**Expected Results for Corresponding Test Step**
1) At one of the signalized intersections disable SPaT transmission	1) The intersection is not identified
2) Re-enable SPaT transmission at the intersection	1) The intersection is identified 2) Lane / Approach matching is performed 3) Warnings are issued

A.33.1.13 Performance Test Cases (PRF)

The following test cases test various aspects of system performance. For these tests there is no pass / fail criteria. These tests are intended for data gathering to allow for performance analysis of various aspects of the system.

A.33.1.13.1 Test Cases

Test 127 – INT-PRF-01: DVI Icon Latency

Test Case #: INT-PRF-01	
Test Description:	DVI Icon latency
Test Setup / Equipment:	Support for DVI Icon The ability to determine activation timing for icon

Test Case #:	INT-PRF-01	
Test Description:	DVI Icon latency	

Test Steps	**Expected Results for Corresponding Test Step**
1) Drive toward any of the intersections such that a warning is provided	1) Collect the appropriate data to determine the latency between the warning flag going high and the icon coming on. 2) Collect the appropriate data to determine the latency between the DVI Icon CAN message and the DVI Icon coming on
2) Repeat Step #1 until have a sufficient amount of data for analysis	

~~Test 128 – INT-PRF-02: DVI Audio Latency~~

~~**Test Case #:**~~	~~INT-PRF-02~~	
~~**Test Description:**~~	~~DVI Audio latency~~	
~~**Test Setup / Equipment:**~~	~~Support for DVI Audio~~ ~~The ability to determine activation timing for audio~~	

~~**Test Steps**~~	~~**Expected Results for Corresponding Test Step**~~
~~1) Drive toward any of the intersections such that a warning is provided~~	~~1) Collect the appropriate data to determine the latency between the warning flag going high and the audio coming on.~~
~~2) Repeat Step #1 until have a sufficient amount of data for analysis~~	

Note: This test is not easy to confirm with the equipment available to the team. It will be verified subjectively as part of the Pilot and Objective testing activities.

Test 129 – INT-PRF-03: DVI Brake Pulse Latency

Test Case #: INT-PRF-03	
Test Description:	DVI Brake Pulse latency
Test Setup / Equipment:	Support for DVI Brake Pulse The ability to determine activation timing for brake pulse
Test Steps	**Expected Results for Corresponding Test Step**
1) Drive toward any of the intersections such that a warning is provided	1) Collect the appropriate data to determine the latency between the warning flag going high and the brake pulse being applied. 2) Collect the appropriate data to determine the latency between the Brake Pulse Flexible CAN trigger message and the brake pulse activation
2) Repeat Step #1 until have a sufficient amount of data for analysis	

Test 130 – INT-PRF-04: W. 10 Mile Rd. & Orchard Lake Rd. Message Reception Range

Test Case #: INT-PRF-04	
Test Description:	W. 10 Mile Rd. and Orchard Lake Rd. DSRC message reception range
Test Setup / Equipment:	The ability to measure the distance at which messages are being received
Test Steps	**Expected Results for Corresponding Test Step**
1) Drive the posted speed limit down Approach #2	1) Determine reliable GID reception range 2) Determine reliable DGPS reception range 3) Determine reliable SPaT reception range
2) Repeat step #1 for Approach #4	- Gather same Step #1 data
3) Repeat step #1 for Approach #6	- Gather same Step #1 data
4) Repeat step #1 for Approach #8	- Gather same Step #1 data
5) Repeat Step #1 - #4 until have a sufficient amount of data for analysis	

Test 131 – INT-PRF-05: W. 12 Mile Rd. & Farmington Rd. Message Reception Range

Test Case #: INT-PRF-05	
Test Description:	W. 12 Mile Rd. and Farmington Rd. DSRC message reception range
Test Setup / Equipment:	The ability to measure the distance at which messages are being received
Test Steps	**Expected Results for Corresponding Test Step**
1) Drive the posted speed limit down Approach #2	1) Determine reliable GID reception range 2) Determine reliable DGPS reception range 3) Determine reliable SPaT reception range
2) Repeat step #1 for Approach #4	- Gather same Step #1 data
3) Repeat step #1 for Approach #6	- Gather same Step #1 data
4) Repeat step #1 for Approach #8	- Gather same Step #1 data
5) Repeat Step #1 - #4 until have a sufficient amount of data for analysis	

Test 132 – INT-PRF-06: DSRC CCH Message Latency

Test Case #: INT-PRF-06	
Test Description:	DSRC CCH message latency
Test Setup / Equipment:	GID, DGPS, SPaT messages configured to transmit on the CCH Metric object functionality enabled for GID, DGPS, SPaT DSRC messages
Test Steps	**Expected Results for Corresponding Test Step**
1) Configure RSE to transmit GID, DGPS, and SPaT on the CCH	1) Verify all three messages are being transmitted and received on the CCH
2) Get within reliable continuous message reception range of a signalized intersection	1) Determine the latency in GID transmission / reception 2) Determine the latency in DGPS transmission / reception 3) Determine the latency in SPaT transmission / reception
3) Repeat Step #2 until have a sufficient amount of data for analysis	
4) Configure RSE back to default configuration with SPaT on CCH, GID on SCH, and DGPS on SCH	1) Verify SPaT is being received on CCH 2) Verify GID and DGPS are being received on SCH

Test 133 – INT-PRF-07: DSRC SCH Message Latency

Test Case #: INT-PRF-07	
Test Description:	DSRC SCH message latency
Test Setup / Equipment:	Default configuration with GID and DGPS messages transmitting on the SCH Metric object functionality enabled for GID, DGPS DSRC messages
Test Steps	**Expected Results for Corresponding Test Step**
1) Get within reliable continuous message reception range of a signalized intersection	1) Determine the latency in GID transmission / reception 2) Determine the latency in DGPS transmission / reception
2) Repeat Step #1 until have a sufficient amount of data for analysis	

Test 134 – INT-PRF-08: Rate of Change of Distance Methods

Test Case #: INT-PRF-08	
Test Description:	Test the performance of the multiple Rate of Change of distance intersection identification calculations
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Configure WSU to use CAN vehicle speed and GPS heading rate of change method	
2) Drive the loop of intersections	1) Verify the correct intersection is identified 2) Gather data related to intersection identification
3) Configure WSU to use GPS location data rate of change method	
4) Drive the loop of intersections	1) Verify the correct intersection is identified 2) Gather data related to intersection identification
5) Configure WSU to use GPS location data with filter rate of change method	
6) Drive the loop of intersections	1) Verify the correct intersection is identified 2) Gather data related to intersection identification
7) Repeat Step #1 – Step #6 until have a sufficient amount of data for analysis	1) Analyze the data for each rate of change method to determine which if any is best

Test 135 – INT-PRF-09: Off-GID Detection Methods

Test Case #: INT-PRF-09	
Test Description:	Test the performance of the multiple Off GID detection methods
Test Setup / Equipment:	
Test Steps	**Expected Results for Corresponding Test Step**
1) Configure WSU to use distance from centerline Off-GID detection method	
2) Drive the vehicle such that it should be off the GID	1) Gather data related to off-GID detection
3) Configure WSU to use distance from lane edge Off-GID detection method	
4) Drive the vehicle such that it should be off the GID	1) Gather data related to off-GID detection
5) Repeat Step #1 – Step #4 until have a sufficient amount of data for analysis	1) Analyze the data for each off-GID detection method to determine which if either is best

Test 136 – INT-PRF-10: Vehicle in Range RSE Comes On Line

Test Case #: INT-PRF-10	
Test Description:	Verify that if the RSE comes on line and starts transmitting DSRC messages while the vehicle is in reliable message reception range
Test Setup / Equipment:	The ability to power down and up the RSE at the intersection or disable / enable safety applications Knowledge of the reliable reception range for the intersection
Test Steps	**Expected Results for Corresponding Test Step**
1) Power down the RSE or disable transmission of all three safety applications	1) Verify no OTA messages are being received
2) Delete the GID for this intersection from WSU storage and re-start the WSU	
3) Drive the vehicle such that it gets in reliable DSRC message reception range	1) No intersection should be identified
4) Power up the RSE / Enable the safety applications such that the vehicle should start receiving messages prior to getting to the stop bar	1) The intersection is identified 2) Lane Matching is working 3) Current Signal Phase and Timing is displayed

Test 137 – INT-PRF-11: Vehicle in Range RSE Goes Off Line

Test Case #:	INT-PRF-11
Test Description:	Verify that if the RSE goes off line while the vehicle is in reliable message reception range there are no ill-effects with the OBE
Test Setup / Equipment:	The ability to power down and up the RSE at the intersection or disable / enable safety applications Knowledge of the reliable reception range for the intersection

Test Steps	**Expected Results for Corresponding Test Step**
1) Drive the vehicle such that it gets in reliable DSRC message reception range	1) Intersection is identified 2) Lane / Approach matching is working as expected
2) Power down the RSE or disable transmission of all three safety applications prior to the vehicle getting to the intersection box	1) No intersection should be identified 2) Lane / Approach matching is disabled
3) After the vehicle exits the intersection, power up the RSE / Enable the safety applications	1) The intersection is not identified
4) Turn the vehicle around and re-approach the intersection	1) Intersection is identified 2) Lane / Approach matching is working as expected

Test 138 – INT-PRF-12: OBE: Netway / GPS / DSRC Link Failure / Recovery

Test Case #:	INT-PRF-12
Test Description:	Verify that the OBE can handle link failure / recovery scenarios
Test Setup / Equipment:	Access to the OBE such that the individual links can be disconnected / reconnected

Test Steps	**Expected Results for Corresponding Test Step**
1) With the OBE properly setup and operating drive an intersection	1) Intersection is identified 2) Lane / Approach matching is working as expected
2) Disable the Netway – OBE link and drive the intersection	1) Intersection is not identified 2) Lane / Approach matching is not performed
3) Re-enable the Netway – OBE link and drive the intersection	1) Intersection is identified 2) Lane / Approach matching is working as expected
4) Repeat steps #1 - #3 for the GPS receiver – OBE link	1) Same results for steps #1 - #3
5) Repeat steps #1 - #3 for the GPS antenna – GPS receiver link	1) Same results for steps #1 - #3
6) Repeat steps #1 - #3 for the DSRC antenna – OBE link	1) Same results for steps #1 - #3

Test 139 – INT-PRF-13: RSE: Signal Controller / GPS / DSRC Link Failure / Recovery

Test Case #: INT-PRF-13	
Test Description:	Verify that the RSE can handle link failure / recovery scenarios
Test Setup / Equipment:	Access to the RSE such that the individual links can be disconnected / reconnected Lane level accuracy intersection
Test Steps	**Expected Results for Corresponding Test Step**
1) With the RSE properly setup and operating drive an intersection	1) Intersection is identified 2) Lane / Approach matching is working as expected
2) Disable the Signal Controller – RSE link and drive the intersection	1) Intersection is not identified 2) Lane / Approach matching is not performed
3) Re-enable the Signal Controller – RSE link and drive the intersection	1) Intersection is identified 2) Lane / Approach matching is working as expected
4) Repeat steps #1 - #3 for the GPS receiver – RSE link	1) Same results for steps #1 - #3
5) Repeat steps #1 - #3 for the GPS antenna – GPS receiver link	1) Same results for steps #1 - #3
6) Repeat steps #1 - #3 for the DSRC antenna – RSE link	1) Same results for steps #1 - #3

A.33.1.14 Operating Scenarios (OPS)

A.33.1.14.1 Simple Traffic Signal Approach (SIM)

Test 140 – INT-OPS-SIM-01: Green Light with No Lane Change – No Warning

Test Case #: INT-OPS-SIM-01	
Test Description:	Approach a green light and drive through the green light without changing lanes
Test Setup / Equipment:	
Expected Results	No warning is given

Test 141 – INT-OPS-SIM-02: Green Light with Lane Change – No Warning

Test Case #: INT-OPS-SIM-02	
Test Description:	Approach a green light and drive through the green light while making a lane change in the same approach
Test Setup / Equipment:	
Expected Results	No warning is given

Test 142 – INT-OPS-SIM-03: Red to Green Light with No Lane Change – No Warning

Test Case #:	INT-OPS-SIM-03
Test Description:	Without changing lanes, approach a red light such that no warning is given Prior to reaching the warning zone the light turns green Drive through on the green light without changing lanes
Test Setup / Equipment:	
Expected Results	No warning is given

Test 143 – INT-OPS-SIM-04: Red to Green Light with Lane Change – No Warning

Test Case #:	INT-OPS-SIM-04
Test Description:	Approach a red light such that no warning is given Prior to reaching the warning zone the light turns green Prior to arriving at the stop bar make a lane change Drive through on the green light
Test Setup / Equipment:	
Expected Results	No warning is given

Test 144 – INT-OPS-SIM-05: Right Hand Turn Green Light – No Warning

Test Case #:	INT-OPS-SIM-05
Test Description:	Make a right hand turn at a green light
Test Setup / Equipment:	
Expected Results	No warning is given

Test 145 – INT-OPS-SIM-06: Left Hand Turn Green Light – No Warning

Test Case #:	INT-OPS-SIM-06
Test Description:	Make a left hand turn at an unprotected left turn green light with no threatening on-coming traffic present
Test Setup / Equipment:	
Expected Results	No warning is given

Test 146 – INT-OPS-SIM-07: Rolling Right on Red – No Warning

Test Case #:	INT-OPS-SIM-07	
Test Description:	Approach an intersection at a low speed such that no warning is given Roll through a right on red light, with no threatening cross-traffic, below the minimum speed threshold	
Test Setup / Equipment:		
Expected Results	No warning is given	

Test 147 – INT-OPS-SIM-08: Stop and Precede Right on Red – No Warning

Test Case #:	INT-OPS-SIM-08	
Test Description:	Approach an intersection at a slow speed such that no warning is given Come to a stop a car length plus MinWarnDistMeters prior to the stop bar When traffic clears and while the light is still red make a right hand turn at a speed lower than the minimum speed threshold	
Test Setup / Equipment:		
Expected Results	No warning is given	

Test 148 – INT-OPS-SIM-09: Red Light with No Lane Change – Warning

Test Case #:	INT-OPS-SIM-09	
Test Description:	Approach an intersection that will turn red prior the vehicle being able to stop while the vehicle is above the minimum speed threshold	
Test Setup / Equipment:		
Expected Results	Warning is given	

Test 149 – INT-OPS-SIM-10: Red Light with Lane Change – Warning

Test Case #:	INT-OPS-SIM-10	
Test Description:	Approach an intersection that will turn red prior the vehicle being able to stop while the vehicle is above the minimum speed threshold and while making a lane change in the same approach within the warning zone.	
Test Setup / Equipment:	The warning table will need to be modified to allow for the warning to occur and still allow time to make the lane change maneuver safely while the warning is present.	
Expected Results	Warning is given	

Test 150 – INT-OPS-SIM-11: Rolling Right on Red – Warning

Test Case #:	INT-OPS-SIM-11
Test Description:	Approach an intersection and roll through a right on red light, with no threatening cross-traffic, above the minimum speed threshold
Test Setup / Equipment:	
Expected Results	Warning is given

Test 151 – INT-OPS-SIM-12: Stop and Proceed Right on Red – Warning

Test Case #:	INT-OPS-SIM-12
Test Description:	Approach an intersection at a slow speed such that no warning is given Come to a stop a car length plus MinWarnDistMeters prior to the stop bar When traffic clears and while the light is still red make a right hand turn at a speed greater than the minimum speed threshold
Test Setup / Equipment:	Configure MinSignalSpeedThreshold to the minimum or very small value Configure MaxWarnDistMeters to be greater than a car length plus MinWarnDistMeters
Expected Results	Warning is given

~~Test 152 – INT-OPS-SIM-13: Brakes Applied – No Warning~~

~~**Test Case #:**~~	~~INT-OPS-SIM-13~~
~~**Test Description:**~~	~~Approach an intersection at a speed higher than the minimum threshold such that the light will turn red prior to arriving at the stop bar~~ ~~Apply foot on brake such that the brakes active indication is True before the vehicle enters the warning zone~~
~~**Test Setup / Equipment:**~~	
~~**Expected Results**~~	~~No warning is given~~

NOTE: This test as written is no longer valid due to the WA table not taking braking into account at signalized intersections. Due to the Pilot FOT and Pre-Objective testing having tested the proper functionality, this test will not be run under Task 10.

~~Test 153 – INT-OPS-SIM-14: Brakes Applied – Warning to No Warning~~

~~**Test Case #:**~~	~~INT-OPS-SIM-14~~
~~**Test Description:**~~	~~Approach an intersection that will turn red prior the vehicle being able to stop while the vehicle is above the minimum speed threshold~~ ~~Apply brakes while warning is going off~~
~~**Test Setup / Equipment:**~~	
~~**Expected Results**~~	~~Transition from Warning to No Warning~~

NOTE: This test as written is not valid. Due to the Pilot FOT and Pre-Objective testing having tested the proper functionality this test will not be run under Task 10.

~~Test 154 – INT-OPS-SIM-15: Brakes Applied – Warning to No Warning to No Warning~~

~~**Test Case #:**~~	~~INT-OPS-SIM-15~~
~~**Test Description:**~~	~~Approach an intersection that will turn red prior to the vehicle being able to stop prior to getting to the stop bar while the vehicle is above the minimum speed threshold~~ ~~Apply brakes while warning is going off~~ ~~Release brakes and speed back up such that, while the light is still red, the vehicle is above the minimum speed threshold and will not be able to stop prior to getting to the stop bar~~
~~**Test Setup / Equipment:**~~	
~~**Expected Results**~~	~~Transition from Warning to No Warning to No Warning~~

NOTE: This test as written is not valid. Due to the Pilot FOT and Pre-Objective testing having tested the proper functionality, this test will not be run under Task 10.

Test 155 – INT-OPS-SIM-16: U-Turn Intersection Identification

Test Case #:	INT-OPS-SIM-16
Test Description:	Approach an intersection and drive through intersection box Make a U-Turn while still in reliable message reception Proceed back toward intersection and drive through intersection box
Test Setup / Equipment:	May require some configuration
Expected Results	Intersection Identification: Identified, Not Identified, Identified, Not Identified

A.33.1.14.2 Simple Stop Sign Approach Scenarios (STP)

Test 156 – INT-OPS-STP-01: Stop Sign with No Lane Change – No Warning

Test Case #:	INT-OPS-STP-01
Test Description:	Approach an intersection such that the distance to stop is greater than the distance to stop bar Keep the vehicle below the minimum speed threshold
Test Setup / Equipment:	
Expected Results	No warning is given

Test 157 – INT-OPS-STP-02: Stop Sign with Lane Change – No Warning

Test Case #:	INT-OPS-STP-02
Test Description:	Approach an intersection such that the distance to stop is greater than the distance to stop bar Keep the vehicle below the minimum speed threshold Make a lane change in the same approach
Test Setup / Equipment:	
Expected Results	No warning is given

Test 158 – INT-OPS-STP-03: Rolling Through Stop Sign – No Warning

Test Case #:	INT-OPS-STP-03
Test Description:	Approach an intersection below the minimum speed threshold Roll through stop sign, with no threatening cross-traffic, below the minimum speed threshold
Test Setup / Equipment:	
Expected Results	No warning is given

Test 159 – INT-OPS-STP-04: Stop and Precede Stop Sign – No Warning

Test Case #: INT-OPS-STP-04	
Test Description:	Approach an intersection at a slow speed such that no warning is given Come to a stop a car length plus MinWarnDistMeters prior to the stop bar When traffic clears proceed through at a speed greater than the minimum speed threshold.
Test Setup / Equipment:	Configure MinStopSignSpeedThreshold to a small the minimum or very small value Configure MaxWarnDistMeters to be greater than a car length plus MinWarnDistMeters
Expected Results	No warning is given

Test 160 – INT-OPS-STP-05: Stop Sign with No Lane Change – Warning

Test Case #: INT-OPS-STP-05	
Test Description:	Approach an intersection such that the distance to stop is greater than the distance to stop bar while the vehicle is above the minimum speed threshold
Test Setup / Equipment:	
Expected Results	Warning is given

Test 161 – INT-OPS-STP-06: Stop Sign with Lane Change – Warning

Test Case #: INT-OPS-STP-06	
Test Description:	Approach an intersection such that the distance to stop is greater than the distance to stop bar while the vehicle is above the minimum speed threshold and while making a lane change in the same approach
Test Setup / Equipment:	
Expected Results	Warning is given

Test 162 – INT-OPS-STP-07: Rolling Stop Sign – Warning

Test Case #: INT-OPS-STP-07	
Test Description:	Approach an intersection at a slow speed and roll through stop sign, with no threatening cross-traffic, above the minimum speed threshold
Test Setup / Equipment:	
Expected Results	Warning is given

A.33.1.14.3 Dedicated Turn Lane Scenarios (DTL)

Test 163 – INT-OPS-DTL-01: Left on Flashing Red – No Warning

Test Case #:	INT-OPS-DTL-01
Test Description:	Approach a flashing red light at a speed below the minimum speed threshold With no on-coming traffic and without stopping make a left hand turn
Test Setup / Equipment:	
Expected Results	No warning is given

Test 164 – INT-OPS-DTL-02: Left on Flashing Red after Stop – No Warning

Test Case #:	INT-OPS-DTL-02
Test Description:	Approach a flashing red light at a slow speed such that no warning is given Come to a stop a car length plus MinWarnDistMeters prior to the stop bar When traffic clears make a left hand turn at a speed greater than the minimum speed threshold
Test Setup / Equipment:	Configure MinSignalSpeedThreshold to the minimum or very small value Configure MaxWarnDistMeters to be greater than a car length plus MinWarnDistMeters
Expected Results	No warning is given

Test 165 – INT-OPS-DTL-03: Left on Green – No Warning

Test Case #:	INT-OPS-DTL-03
Test Description:	Make a left hand turn at a protected left turn green light
Test Setup / Equipment:	
Expected Results	No warning is given

Test 166 – INT-OPS-DTL-04: Lane Change Red to Green – Warning to No Warning

Test Case #: INT-OPS-DTL-04	
Test Description:	Set the Minimum warning time to zero Approach an intersection on a straight-through approach such that a warning is going off Make a lane change from the straight through red warning approach into a green left hand turn approach
Test Setup / Equipment:	Minimum warning time set to zero
Expected Results	Transition from Warning to No Warning

Test 167 – INT-OPS-DTL-05: Lane Change Red to Green – Warning to Warning

Test Case #: INT-OPS-DTL-05	
Test Description:	Set the Minimum warning time to the maximum value Approach an intersection on a straight-through approach such that a warning is going off Make a lane change from the straight through red warning approach into a green left hand turn approach
Test Setup / Equipment:	Minimum warning time set to the maximum value
Expected Results	Transition from Warning to No Warning

Test 168 – INT-OPS-DTL-06: Lane Change Green to Flashing Red – Warning

Test Case #: INT-OPS-DTL-06	
Test Description:	Make a lane change from a straight through green approach into a flashing red dedicated left hand turn approach above the minimum speed threshold.
Test Setup / Equipment:	
Expected Results	Warning is given

A.34 Individual Test Case Status

Test Cases		Status	
Test Case ID	Brief Description	Pass (P) Fail (F) Removed (RM)	Fixed (Y/N)
CMP-VEH-01	$600-$605 Batch CAN Processing	P	
CMP-VEH-02	Incomplete $600-$605 CAN Message Set	P	
CMP-VEH-03	Incomplete CAN Message Set $605 Processing	P	
CMP-VEH-04	CAN Timeout Support	P	
CMP-VEH-05	CAN Timeout Configuration Support	F	Y
CMP-VEH-06	CAN Reception Rate	P	
CMP-VEH-07	Message Reception Rate Variability	P	
CMP-VEH-08	Message $650 Support	P	
CMP-VEH-09	Heartbeat Support	P	
CMP-VEH-10	$606 Heartbeat Error Processing	F	N
CMP-VEH-11	Logging Enable / Disable	P	
CMP-RAD-RH-01	Verify WBSS Join	P	
CMP-RAD-RH-02	Verify WBSS Detach	P	
CMP-RAD-RH-03	Verify Periodic Radio Stats Requests	P	
CMP-RAD-RH-04	Verify Radio Stats Polling Rate Configuration	P	
CMP-RAD-RSC-01	Verify PSID Configuration	P	
CMP-RAD-RSC-02	Unrecognized PSID Handling	P	
CMP-RAD-HDR-01	Invalid Message Type (TOM Header)	P	
CMP-RAD-HDR-02	Unsupported TOM Framework Version	P	
CMP-RAD-HDR-03	Incorrect / Zero Message Length	P	
CMP-RAD-HDR-04	Incorrect / Too Small Message Length	P	
CMP-RAD-HDR-05	Incorrect / Too Large Message Length		
CMP-RAD-HDR-06	Incorrect CRC-16	P	
CMP-RAD-FTR-01	Invalid Message Termination Flag	P	
CMP-RAD-NULL-01	Empty Message	P	
CMP-RAD-NULL-02	Empty Layer	P	
CMP-RAD-NML-01	Normal GID on CCH	P	
CMP-RAD-NML-02	Normal GID on SCH	P	
CMP-RAD-NML-03	Normal SPAT	P	
CMP-RAD-NML-04	Normal GPSC on CCH	P	
CMP-RAD-NML-05	Normal GPSC on SCH	P	
CMP-RAD-NML-06	Maximum DSRC Message Receipt Support	P	
CMP-RAD-LAY-01	Unsupported Layer	P	
CMP-RAD-LAY-02	Dangling Layer	P	
CMP-RAD-RHTX-01	TSVWG Transmission	RM	
CMP-RAD-RHTX-02	RCMD Transmission	RM	
CMP-RAD-HB-01	Heartbeat transmission to Error Handler	RM	
CMP-RAD-HB-02	Heartbeat Error	RM	
CMP-GPSH-01	GPS Handler Registration	P	
CMP-GPSH-02	GPS Healthy Status Check (WSA)	P	
CMP-GPSH-03	NMEA Data Processing	F	Y
CMP-GPSH-04	GPS Data Timer Expiration	F	Y
CMP-GPSH-05	DGPS Metric Object Processing	P	
CMP-GPSH-06	DGPS Unhealthy Check	F	Y
CMP-GPSH-07	RTCM Checksum Check	P	
CMP-GPSH-08	Error Reporting to Error Handler	RM	
CMP-GPSH-09	GPS Handler Heartbeat Support	RM	
CMP-GID-01	GID Object Parsing	P	
CMP-GID-02	Un-Supported GID Object Parsing	P	
CMP-GID-03	Un-supported GID Format Version Processing	P	
CMP-GID-04	Multiple GID Storage	F	Y
CMP-GID-05	GID Storage Configuration	F	Y
CMP-GID-06	GID Content Version Processing	P	
CMP-GID-07	Expired GID Deletion	P	
CMP-GID-08	GID Load Time Updates		
CMP-GID-09	GID Expiration Period Configuration		
CMP-GID-10	New GID Received, Oldest GID Deleted		
CMP-GID-11	Area GID Processing	P	
CMP-GID-12	GID Metric Object Processing	F	Y
CMP-GID-13	GID Handler Heartbeat Processing	RM	
CMP-GID-14	GID Heartbeat Error	RM	
CMP-GID-15	Intersection Reference Point Updates	P	
CMP-GID-16	Intersection Reference Point List via Stored GIDs	P	
CMP-SPAT-01	Verify DSRC Link and Basic SPaT Message	P	
CMP-SPAT-02	Verify Correct Operation of WSU SPaT Parser (static	P	
CMP-SPAT-03	Verify Correction Operation of WSU SPaT Parser (live	P	
CMP-SPAT-04	Verify SPaT is Parsed and Available to Applications in	P	
CMP-DASH-01	CAN Message $701 Registration	P	
CMP-DASH-02	CAN $606 OBE-DAS Heartbeat Transmission	P	
CMP-DASH-03	CAN $701 DAS-OBE Heartbeat Processing	F	Y
CMP-DASH-04	OBE-DAS Data Frame Composition	P	
CMP-DASH-05	OBE-DAS $751 Event Triggered Transmission	P	
CMP-DASH-06	OBE-DAS Periodic Data Transmission	P	
CMP-DASH-07	Data Logging	P	
CMP-DASH-08	Heartbeat Transmission to Error Handler	RM	
CMP-IID-01	Directional Intersection Approach Test	P	
CMP-IID-02	Distance from Intersection Identification Verification	P	
CMP-IID-03	Zero Speed Velocity Behavior	P	
CMP-IID-04	Non-pre-loaded GID Selection	P	
CMP-IID-05	Pre-loaded GID Selection	P	
CMP-LID-01	Quick Lane Shifts	P	
CMP-LID-02	Multiple Lane Shifts	P	
CMP-LID-03	Early Lane Shift	P	
CMP-LID-04	GPS Antenna Offset Verification	P	

Test Cases		Status	
Test Case ID	Brief Description	Pass (P) Fail (F) Removed (RM)	Fixed (Y/N)
CMP-LID-05	Lane Segment Transition Test	P	
CMP-LID-06	Off Lane but On GID Threshold	P	
CMP-WARN-01	SPaT Requirement for Warning	P	
CMP-WARN-02	Approach ID Requirement for Warning	P	
CMP-WARN-03	WA Sleep Frequency	P	
CMP-WARN-04	Equipped Logic at Stop Sign Intersection	P	
CMP-WARN-05	Equipped Logic at Signalized Intersection	P	
CMP-WARN-06	Warning Logic Approach ID Input Verification	RM	
CMP-WARN-07	Warning Logic SPAT Input Verification	RM	
CMP-WARN-08	Is Braking Logic Verification	P	
CMP-WARN-09	Is Braking Transition	P	
CMP-WARN-10	Time to Red Logic Verification	RM	
CMP-WARN-11	Time to Stop Bar Verification	RM	
CMP-WARN-12	Time Comparison Verification	RM	
CMP-WARN-13	Distance Comparison Verification	RM	
CMP-WARN-14	Braking Exception Logic Verification	P	
CMP-WARN-15	Minimum Warning Distance Threshold	P	
CMP-WARN-16	SPAT Freewheeling Logic	F	Y
CMP-DVI-01	State Machine Input/Output Verification I	P	
CMP-DVI-02	State Machine Input/Output Verification II	P	
CMP-DVI-03	State Machine Input/Output Verification III	P	
CMP-DVI-04	State Machine Input/Output Verification IV	P	
CMP-DVI-05	Flexible Trigger Warning Message Verification	P	
CMP-DVI-06	Pre-Warning State Transitions	P	
CMP-PWR-01	Power Cycle Disk Resiliency	P	
CMP-PWR-02	Correct Power Mode Choice	P	
CMP-WDG-01	Application Heartbeat Message Transmission	P	
CMP-WDG-02	Missed Heartbeats WSU Reset	P	
INT-SI-01	GID, GPSC, SPAT Tx / Rx	P	
INT-SI-02	GID, GPSC, SPAT Tx Rate	P	
INT-SI-03	W. 10 Mile Rd. & Orchard Lake Rd. Delta Position	P	
INT-SI-04	W. 12 Mile Rd. & Farmington Rd. Delta Position	P	
INT-SI-05	W. 10 Mile Rd. & Orchard Lake Rd.	P	
INT-SI-06	W. 12 Mile Rd. & Farmington Rd. Approach/Lane/Stop	P	
INT-SI-07	W. 11 Mile Rd. & Drake Rd. Approach/Lane/Stop Bar	P	
INT-SI-08	W. 10 Mile Rd. & Orchard Lake Rd. Warning Distance	P	
INT-SI-09	W. 12 Mile Rd. & Farmington Rd. Warning Distance	P	
INT-SI-10	W. 11 Mile Rd. & Drake Rd. Warning Distance	P	
INT-SI-11	Lane vs Road Level Accuracy Verification	F	Y
INT-SI-12	DVI Icon / Audio Activation Timing	P	
INT-SI-13	DVI Icon / Audio / Brake Pulse Activation Timing	P	
INT-SI-14	No Signalized IID when no SPaT	P	
INT-PRF-01	DVI Icon Latency	P	
INT-PRF-02	DVI Audio Latency	RM	
INT-PRF-03	DVI Brake Pulse Latency	P	
INT-PRF-04	W. 10 Mile Rd. & Orchard Lake Rd. Message	P	
INT-PRF-05	W. 12 Mile Rd. & Farmington Rd. Message Reception	P	
INT-PRF-06	DSRC CCH Message Latency	F	Y
INT-PRF-07	DSRC SCH Message Latency	F	Y
INT-PRF-08	Rate of Change of Distance Methods	P	
INT-PRF-09	Off-GID Detection Methods	P	
INT-PRF-10	Vehicle In Range RSE Comes On Line	P	
INT-PRF-11	Vehicle In Range RSE Goes Off Line	P	
INT-PRF-12	OBE: Netway / GPS / DSRC Link Failure / Recovery	P	
INT-PRF-13	RSE: Signal Controller / GPS / DSRC Link Failure /	P	
INT-OPS-SIM-01	Green Light with No Lane Change - No Warning	P	
INT-OPS-SIM-02	Green Light with Lane Change - No Warning	P	
INT-OPS-SIM-03	Red to Green Light with No Lane Change - No	P	
INT-OPS-SIM-04	Red to Green Light with Lane Change - No Warning	P	
INT-OPS-SIM-05	Right Hand Turn Green Light - No Warning	P	
INT-OPS-SIM-06	Left Hand Turn Green Light - No Warning	P	
INT-OPS-SIM-07	Rolling Right on Red - No Warning	P	
INT-OPS-SIM-08	Stop and Proceed Right on Red - No Warning	P	
INT-OPS-SIM-09	Red Light with No Lane Change - Warning	P	
INT-OPS-SIM-10	Red Light with Lane Change - Warning	P	
INT-OPS-SIM-11	Rolling Right on Red - Warning	P	
INT-OPS-SIM-12	Stop and Proceed Right on Red - Warning	F	Y
INT-OPS-SIM-13	Brakes Applied - No Warning	RM	
INT-OPS-SIM-14	Brakes Applied - Warning to No Warning	RM	
INT-OPS-SIM-15	Brakes Applied - Warning to No Warning to No	RM	
INT-OPS-SIM-16	U-Turn Intersection Identification	P	
INT-OPS-STP-01	Stop Sign with No Lane Change - No Warning	P	
INT-OPS-STP-02	Stop Sign with Lane Change - No Warning	P	
INT-OPS-STP-03	Rolling Through Stop Sign - No Warning	P	
INT-OPS-STP-04	Stop and Proceed Stop Sign - No Warning	P	
INT-OPS-STP-05	Stop Sign with No Lane Change - Warning	P	
INT-OPS-STP-06	Stop Sign with Lane Change - Warning	P	
INT-OPS-STP-07	Rolling Stop Sign - Warning	P	
INT-OPS-DTL-01	Left on Flashing Red - No Warning	P	
INT-OPS-DTL-02	Left on Flashing Red after Stop - No Warning	P	
INT-OPS-DTL-03	Left on Green - No Warning	P	
INT-OPS-DTL-04	Lane Change Red to Green - Warning to No Warning	P	
INT-OPS-DTL-05	Lane Change Red to Green - Warning to Warning	P	
INT-OPS-DTL-06	Lane Change Green to Flashing Red - Warning	P	

U.S. Department of Transportation
ITS Joint Program Office-HOIT
1200 New Jersey Avenue, SE
Washington, DC 20590

Toll-Free "Help Line" 866-367-7487
www.its.dot.gov

FHWA-JPO-10-068

U.S. Department of Transportation

Research and Innovative Technology Administration