5.9 GHz Dedicated Short Range Communication Vehicle-based Road and Weather Condition Application

Final Report

Version 1.0

August 2015

Prepared For:

Connected Vehicle Pooled Fund Study

By:

Synesis Partners LLC

TABLE OF CONTENTS

EXECUTIVE SUMMARY .... **III**

1 INTRODUCTION **1**

1.1 Purpose **1**

1.2 Background and Scope **1**

1.3 Definitions, Acronyms, and Abbreviations **3**

1.4 References **3**

1.5 Overview **3**

2 APPLICATION PLANNING AND SPECIFICATION **5**

2.1 Concept of Operations **5**

2.2 Messaging Standards **7**

3 SYSTEM IMPLEMENTATION **10**

3.1 Vehicle On-board Components **10**
  3.1.1 OBE Hardware 11
  3.1.2 IceSight 13
  3.1.3 OBE Software 14
  3.1.4 OBE Interfaces 16

3.2 Roadside Components **28**
  3.2.1 RSE Hardware 28
  3.2.2 RSE Software 29
  3.2.3 RSE Interfaces 30

3.3 Back Office Services **31**

4 DEPLOYMENT **33**

4.1 Testing **33**

4.2 Backhaul Network and IPv6 .... **35**
  4.2.1 CV Application Needs and the IPv6 Network Challenge 35
  4.2.2 IPv6 Network Deployment Planning and Project Experience 37

4.3 In-Vehicle and Roadside Installation **40**

5 ANALYSIS AND RECOMMENDATIONS **42**

5.1 Messaging Standards 42

5.2 OBU and RSU Software Portability 43

5.3 IPv6 Network Support for CV Applications 43

APPENDIX A - DEFINITIONS **46**

APPENDIX B - ADDITIONAL APPLICATION SOFTWARE DESCRIPTION **49**

APPENDIX C - ICESIGHT INTERFACE **52**

TABLE OF FIGURES

FIGURE 1 - SYSTEM CONCEPT INTERACTIONS (SOURCE: SYNESIS PARTNERS LLC) 6
FIGURE 2 - CONCEPTUAL SYSTEM DEPLOYMENT (SOURCE: SYNESIS PARTNERS LLC) 10
FIGURE 3 - VEHICLE ON-BOARD COMPONENTS (SOURCE: SYNESIS PARTNERS LLC) 11
FIGURE 4 - HEAVY VEHICLE OBE, ANTENNA AND CABLES (SOURCE: SYNESIS PARTNERS LLC) 12
FIGURE 5 - LIGHT VEHICLE OBE, ANTENNA AND CABLES (SOURCE: SYNESIS PARTNERS LLC) 13
FIGURE 6 - HIGH SIERRA ICESIGHT 2020S (SOURCE: SYNESIS PARTNERS LLC) 14
FIGURE 7 - OBE APPLICATION SOFTWARE MODULES (SOURCE: SYNESIS PARTNERS LLC) 15
FIGURE 8 - EXAMPLE OBU CONFIGURATION FILE (SOURCE: SYNESIS PARTNERS LLC) 17
FIGURE 9 - RSE SCHEMATIC (SOURCE: SYNESIS PARTNERS LLC) 28
FIGURE 10 - RSE, ETHERNET SWITCH, AND BRACKET/SURGE SUPPRESSOR (SOURCE: SYNESIS PARTNERS LLC) 29
FIGURE 11 - DATA FILE LIST ON SAVARI MONITORING SERVER APPLICATION (SOURCE: SAVARI INC.; SYNESIS PARTNERS LLC) 32
FIGURE 12 - TYPICAL RSE DEPLOYMENT VALIDATION RESULT (SOURCE: SYNESIS PARTNERS LLC) 35
FIGURE 13 - CONCEPTUAL IPV6 NETWORK FOR CV APPLICATIONS (SOURCE: SYNESIS PARTNERS LLC) 39
FIGURE 14 - STARTUP SEQUENCE DIAGRAM (SOURCE: SYNESIS PARTNERS LLC) 49
FIGURE 15 - COLLECT SEQUENCE DIAGRAM (SOURCE: SYNESIS PARTNERS LLC) 50
FIGURE 16 - UPLOAD SEQUENCE DIAGRAM (SOURCE: SYNESIS PARTNERS LLC) 51

TABLE OF TABLES

TABLE 1 - APPLICABLE MESSAGING STANDARDS (SOURCE: SYNESIS PARTNERS LLC) 7
TABLE 2 - MESSAGING REQUIREMENTS OPTIONS (SOURCE: SYNESIS PARTNERS LLC) 9
TABLE 3 - OBE CONFIGURATION FILE PARAMETER SPECIFICATION (SOURCE: SYNESIS PARTNERS LLC) 19
TABLE 4 - EXAMPLE HEARTBEAT MESSAGE RECORDS (SOURCE: SYNESIS PARTNERS LLC) 31
TABLE 5 - ICESIGHT DATA RECORD DEFINITIONS (SOURCE: HIGH SIERRA ELECTRONICS) 52
TABLE 6 - ICESIGHT DATA PARAMETER DEFINITIONS (SOURCE: HIGH SIERRA ELECTRONICS) 52

REVISION HISTORY

Version	Description
1.0	Initial version.

EXECUTIVE SUMMARY

The objective of this Connected Vehicle Pooled Fund Study (CVPFS) project is to develop and test the acquisition of road and weather condition information on public agency vehicles and transmit it to roadside equipment over 5.9 GHz Dedicated Short-Range Communications (DSRC). The project consisted of four tasks that included activities needed for this and any other similar CV application development initiatives:

> **Task 1 – Requirements Development** described the data elements and data sets desired for the road weather applications; determined what weather-related data are actually available on each of the vehicles; and identified what additional sensors and equipment would be needed to provide the desired data sets. The output of this task is documented in a *Messaging Requirements* specification.
>
> **Task 2 – Concept of Operations** developed a *Concept of Operations* (ConOps) for collecting and processing the data on the DSRC on-board equipment (OBE) and sending the data to the roadside equipment (RSE). The ConOps includes use cases, a description of the system architecture, and high-level system requirements.
>
> **Task 3 – Application Development** specified the DSRC equipment, developed the OBE, RSE and data transmission components, and determined any adaptations needed to support integration with existing New York State DOT DSRC deployments on the Long Island Expressway.
>
> **Task 4 – Application Installation and Testing** procured equipment, selected deployment sites, assembled and tested the system hardware and software in preparation for New York State DOT deployment. Testing parameters are documented in a *Test Plan*, and the deployment process is described in an *Installation Guide*.

Hardware procured for the project included six DSRC OBEs from Cohda Wireless, three DSRC RSEs from Savari, and two High Sierra IceSight mobile road weather sensor units to provide additional on-board data gathering. The software developed in the project for the OBE can be configured to collect data from the vehicle's Controller Area Network (CAN) bus, the aftermarket IceSight device, and Dickey John road treatment equipment (if present), in addition to the Global Positioning System (GPS). Data are transmitted to the DSRC RSE using IPv6 messaging, and are stored as files on the RSE. Data can be retrieved from the RSE by agency network administrators and systems over a backhaul connection.

Several development and deployment challenges were raised and overcome. In particular, agencies deploying the DSRC-based road weather system and similar CV applications will want to assure that DSRC components are fully standards-compliant and meet the application functional requirements, and that sufficient IPv6 knowledge and skills are available to support deployment and operations.

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to provide a summary report of project findings and experience for the 5.9 GHz Dedicated Short Range Communication (DSRC) Vehicle Based Road and Weather Condition Application developed for the Connected Vehicle (CV) Pooled Fund Study (PFS). The report summarizes project activities and work products created throughout the project, including: the Concept of Operations; Messaging Requirements; the system software, hardware, and communications interfaces for the vehicle on-board and roadside equipment; an Installation Guide; backhaul communications networking experience; and a Test Plan for a future deployment.

1.2 Background and Scope

Significant effort has been and continues to be expended in the Federal Highway Administration's (FHWA) Road Weather Management Program and in various federal and state connected vehicle programs to identify opportunities to acquire data from vehicles acting as mobile sensor platforms. Federal, state and local transportation agencies have also been working with automakers and communications technology providers to develop and standardize information exchange between vehicles and the transportation infrastructure, enabling a variety of applications that could improve transportation safety, mobility and environmental performance. This 5.9 GHz DSRC Vehicle-based Road and Weather Condition Application project is a synergistic result of those converging opportunities.

Accurate, timely and route-specific weather information allows traffic and maintenance managers to better operate and maintain roads under adverse conditions. The research system developed by this project enables collection of vehicle-based probe and observation data from mobile sensors on transportation agency vehicles and transmission of the data over DSRC to roadside units from where it can be accessed by agency systems such as the New York State DOT INFORM. In this way, information from mobile platforms will eventually enable traffic managers and maintenance personnel to implement operational strategies that optimize the performance of the transportation system by mitigating the effects of weather on the roadways.

Potential use case scenarios for the system are well known from previous connected vehicle and road weather research. All connected vehicle applications make use of and depend on probe data, but six high-priority connected vehicle road weather applications were specifically identified in the *Concept of Operations for Road Weather Connected Vehicle Applications*[1]. Many of these applications/use cases recognize agency vehicles, including snow plow and maintenance trucks, as key sources of connected vehicle road-weather data, particularly since they are logical candidates for the installation of specialized sensors that will generate data sets that will be unavailable from vehicles in the general public fleet. Other use cases/applications are focused on delivering data to agency vehicles, especially for winter maintenance decision support and for maintenance management systems. The six road weather applications are:

- Enhanced Maintenance Decision Support System
- Information for Maintenance and Fleet Management Systems
- Variable Speed Limits for Weather-Responsive Traffic Management
- Motorist Advisories and Warnings
- Information for Freight Carriers
- Information and Routing Support for Emergency Responders

Within the greater connected vehicle context, the scope of this project is to develop, test, and prepare to deploy in-vehicle and roadside components with 5.9 GHz DSRC capabilities for road and weather condition data in maintenance and highway emergency local patrol (HELP) vehicles. The system is capable of obtaining vehicle data from SAE J1939 and J1979 diagnostic buses and various peripheral devices on maintenance vehicles; transmitting this data from 5.9 GHz DSRC on-board equipment (OBE) to compliant roadside equipment (RSE)[2]; and providing the data on the roadside equipment to agency systems when requested. It is envisioned that this application could be deployed on agency maintenance vehicles of the members of the CV PFS along connected vehicle test beds.

---

[1] U.S. Department of Transportation, Federal Highway Administration, "Concept of Operations for Road Weather Connected Vehicle Applications," prepared by Booz Allen Hamilton, Report No. FHWA-JPO-13-047, February 2013.

[2] The DSRC community vernacular refers to the on-board radio unit as an "OBE" and the roadside radio unit as an "RSE". This is somewhat confusing since there are other on-board equipment and roadside equipment components other than the DSRC radios. "OBE" and "RSE" will be used in this report to refer specifically to the DSRC units, and "on-board equipment" and "roadside equipment" will be used to refer to the equipment more generally deployed in those locations.

1.3 Definitions, Acronyms, and Abbreviations

This document may contain terms, acronyms, and abbreviations that are unfamiliar to the reader. A description of these terms, acronyms, and abbreviations is provided in Appendix A.

1.4 References

The following documents contain additional information pertaining to this project and the requirements for the system,

5.9 GHz Dedicated Short Range Communication Vehicle Based Road and Weather Condition Application Concept of Operations, May 2013, Synesis Partners LLC.

5.9 GHz Dedicated Short Range Communication Vehicle Based Road and Weather Condition Application Messaging Requirements, May 2013, Synesis Partners LLC.

5.9 GHz Dedicated Short Range Communication Vehicle Based Road and Weather Condition Application Test Plan, December 2013, Synesis Partners LLC.

5.9 GHz Dedicated Short Range Communication Vehicle Based Road and Weather Condition Application Installation Guide, August 2015, Synesis Partners LLC.

The Institute of Electrical and Electronics Engineers, Inc., 1990, *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Std 610.12-1990.

The Institute of Electrical and Electronics Engineers, Inc., 1998, *IEEE Standard for Software Test Documentation.* IEEE Std 829-1998, ISBN 0-7381-1443-X SH94687.

1.5 Overview

The remaining sections of the document describe each of the three major system interfaces and their potential use in connected vehicle.

Section 2 – Application Planning and Specification summarizes the basis for and content of the system Concept of Operations and the Messaging Requirements. These documents were generated early in the project and have been published as independent work products.

Section 3 – System Implementation describes the system OBE and RSE implementations in terms of its hardware, software, and interfaces.

Section 4 – Deployment summarizes the system in-vehicle and roadside installation (for which an Installation Manual is provided as a separate work product), describes the supporting backhaul network, and summarizes the system Test Plan (also available as a separate work product).

Section 5 – Analysis and Recommendations discusses key project findings and suggests topics for further investigation.

2 APPLICATION PLANNING AND SPECIFICATION

Development of the 5.9 GHz DSRC vehicle-based data acquisition system in this project began with a Concept of Operations and an analysis of Messaging Requirements. The purpose of these tasks was to identify existing and applicable CV research and standards, and then describe the means of applying the research and standards to acquiring the data over DSRC for CV applications. The gaps between the existing needed capabilities would define the particulars for new development.

2.1 *Concept of Operations*

The Concept of Operations developed for this project[3] describes concepts for a system to support road weather operations using 5.9 GHz DSRC for mobile data gathering. It is based largely on descriptions of the current situation and justifications for change detailed in the *Concept of Operations for Road Weather Connected Vehicle Applications* mentioned in the Introduction to this report. That document describes six road weather-related application concepts and scenarios in which data from vehicles might be used to improve safety, agency operations and traveler information. The concepts and operational scenarios for this project's DSRC-based application more specifically describe a 5.9 GHz DSRC implementation for New York State DOT (NYSDOT) snow plow trucks and HELP vehicles operating along and near the Long Island Expressway from which data might be used in agency operations and traveler information systems.

As is often the case with research of this type, the scope and particulars of the demonstration application changed over the course of the project. It became clear that demonstrating particular applications of the gathered data was less useful than a more complete and flexible implementation of the ability to obtain data from various vehicle data sources, assemble the appropriate data messages for DSRC transmission, and store the data on the RSE for later retrieval. For example, it was decided during the conduct of the project to add an additional data source—an IceSight mobile weather data sensor—to the vehicle configuration and on-board data acquisition, exercising different device connection, protocols and data formats for the OBE. By contrast, the back-end network acquisition of the data from the RSE and its use in potential CV applications—by NYSDOT's INformation FOR Motorists (INFORM) system, for example—does not differ significantly from use cases for more traditional road weather information systems. As such, the overall implementation scope of the application was pared back, with a corresponding increased focus on generalizing the on-board data collection, the

---

[3] Cooperative Transportation Systems [Connected Vehicle] Pooled Fund Study, "5.9 GHz Dedicated Short Range Communication Vehicle-based Road and Weather Condition Application – Concept of Operations," prepared by Synesis Partners LLC, Version 1.0, August 2013.

data transmission over DSRC, and the staging of data on the RSE for use by others. The final implementation concept as illustrated in Figure 1 emphasizes the DSRC components and communications, and defers the application to the particular agency deployments. Data collected on the RSU can be retrieved over a network connection by distributor(s) and applications as the need arises, but there is no data "push" from the RSU.

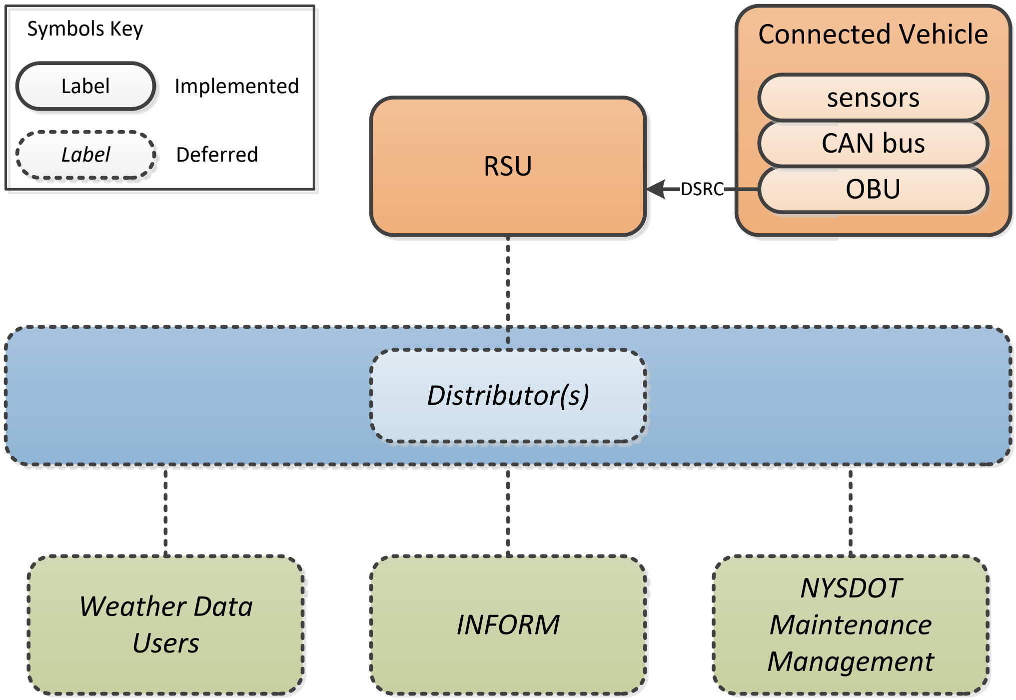


Figure 1 - System Concept Interactions (Source: Synesis Partners LLC)

This refocus and deferment is reflected in the application scenario as well as in the system concept. Whereas the more extensive scenario in the ConOps includes steps describing data flowing out to NYSDOT's INFORM system, the application scenario as implemented is focused on the DSRC interaction:

1. Vehicles equipped for connected vehicle data gathering operate on the roadways
2. Sensors on the vehicle measure and record data, in some instances reporting data to the vehicle's data bus (typically a Controller Area Network (CAN) bus)
3. The vehicle's OBE obtains weather-related data from sensor systems and the data bus, when available
4. OBE formats data snapshots into probe data message(s)
5. OBE stores probe data message(s) for transmittal
6. Vehicle comes in range of RSE; OBE receives service announcement from RSE
7. OBE broadcasts probe data message(s) to RSE
8. RSE stores probe data message(s)

Data can then be retrieved from the RSE by any system with appropriate access permissions through the data interface described later in this report.

2.2 *Messaging Standards*

The Messaging Requirements developed for this project[4] addressed application of the DSRC standards, device interfaces and data element definitions for data acquisition and communications. As was the case for the ConOps, the purpose was to identify the relevant standards and describe their application in the context of this project, rather than to develop any new standard(s) unique to the project. Lists and descriptions of applicable data standards and data elements were included in the document. Table 1 summarizes the applicable standards and their relevance.

Table 1 - Applicable Messaging Standards (Source: Synesis Partners LLC)

Standard	Title	Relevance
IEEE 802.11p	Wireless Access in Vehicular Environments	Specifies the extensions to IEEE Std 802.11 for wireless local area networks (WLANs) — the radio standard—providing wireless communications while in a vehicular environment
IEEE 1609.x	Wireless Access in Vehicular Environments (WAVE)	Set of standards describing protocols for WAVE messaging
IEEE 1609.3	Wireless Access in Vehicular Environments (WAVE) – Networking Services	The 1609 standard describing the use of Internet Protocol (IP), User Datagram Protocol (UDP), and Transmission Control Protocol (TCP) in WAVE
SAE J1939	Serial Control and Communications Heavy Duty Vehicle Network (Top Level Document)	Describes the vehicle data network, layer structure, documentation, and pre-assigned data elements for heavy vehicles
SAE J1979	E/E Diagnostic Test Modes [Ed. Note: E/E in this context is Electrical/Electronic]	Describes light vehicle On-Board Diagnostics (OBD) services, messaging and data availability
SAE J2735	Dedicated Short Range Communications (DSRC) Message Set Dictionary™	Specifies a message set, data frames and elements for applications using 5.9 GHz DSRC/WAVE

---

[4] Cooperative Transportation Systems [Connected Vehicle] Pooled Fund Study, "5.9 GHz Dedicated Short Range Communication Vehicle-based Road and Weather Condition Application – Messaging Requirements," prepared by Synesis Partners LLC, Version 2, August 2013.

Exhaustive lists of the road weather-related data elements described in these standards were provided in the appendices to the Messaging Requirements document. Analysis revealed three potential gaps in the specifications: data elements defined by the J1939 and J1979 vehicle standards missing from the J2735 DSRC messaging standard; data elements defined in J2735 but missing from J1939/1979; and elements available from third-party equipment that is not explicitly specified in J1939, J1979, or J2735. The messaging requirements stated that all available data elements would be accommodated in the outgoing OBE messages and specified the use of the DSRC Basic Safety Message (BSM), with its optional free-form “local” content field for non-standard data.

Using the BSM in this manner did not, however, prove to be feasible with the available DSRC OBEs and RSEs. As shown in Table 2, Versions of the DSRC components available during the course of the project supported only the BSM Part 1 and not the extended BSM Part 2 capabilities. (Even if they were supported, BSM Part 2 did not support all of the potential or desirable data elements.) Forwarding of BSM messages from the RSE to back office systems was supported in the Ann Arbor Safety Pilot demonstration, but not guaranteed to be available on future generations of RSEs. The availability of other potential DSRC messaging services—the Probe Vehicle Data Message (PVDM) and the A la Cart Message—was similarly questionable on current and future equipment.

The best option for fulfilling the project intent to acquire and send the available weather-related data was determined to be compiling the data on the OBE into records in a simple comma-separated variable file for transmission over the DSRC Internet Protocol (IP) service to the RSE. Although this approach did not take advantage of any of the particular DSRC messaging services, it does have the advantage of being somewhat portable—that is, independent of the availability of any particular DSRC service other than IP—between and among OBEs and RSEs.

Using the DSRC IP services and a simple comma-separated value (CSV) file has an additional advantage of providing flexibility to use the data payload for whatever vehicle data may be available and desired. The utility of the application is thereby extended beyond the original specified purpose of collecting road weather data and becomes a generalized tool for other potential CV applications. The system implementation that enables the data payload to be configured for gathering generalized vehicle probe data, including road weather-related data, is described in Section 3.

Table 2 - Messaging Requirements Options (Source: Synesis Partners LLC)

	This Project			Future Deployments		
Message/ Function	J2735 2009	Cohda OBE MK2	Savari RSE v3.x	J2735™ 2015	Cohda OBE MK4a/5	Savari RSE v4.x
BSM Part 1	●	⭘[5]	⭘[6]	●	●	●[7]
BSM Part 2	●	○	○	●	○	●
PVDM	●	○	○	●	○	●
IP [Secure Shell (SSH)]	n/a	●	●	n/a	●	●
WAVE Message Forwarding	n/a	n/a	⭘[8]	n/a	n/a	●

Table Key: ● - supports the message type/function; ○ - does not support the message type/function; ⭘ - partially supports the message type/function; n/a – message/function not applicable to this standard/component

---

[5] The Cohda Mark2 OBU natively supports the "Here I Am" implementation of the BSM Part 1 (i.e., without the optional J2735 optional elements; the BSM Part 2 is not supported.

[6] The Savari RSU 3.x provides non-standard BSM validation as implemented for the Safety Pilot Demonstration.

[7] Per the RSU 4.x specification, the Savari RSU 4.x forwards the BSM and all other WAVE message types according to its configuration settings.

[8] The Savari RSE 3.x supported a non-standard BSM forwarding for the Safety Pilot Demonstration; it did not support the standard forwarding of other WAVE messages as specified in the 4.x specification.

3 SYSTEM IMPLEMENTATION

As discussed in the Concept of Operations, the deployment concept for the system consists of on-board equipment capable of acquiring, caching, formatting and sending data from the vehicle over DSRC to roadside equipment from which other agency systems can in turn acquire the data for use in connected vehicle applications. This section of the report describes the implementation of the vehicle on-board and roadside component hardware, software, and interfaces. All implemented hardware and software has been delivered to New York State DOT.

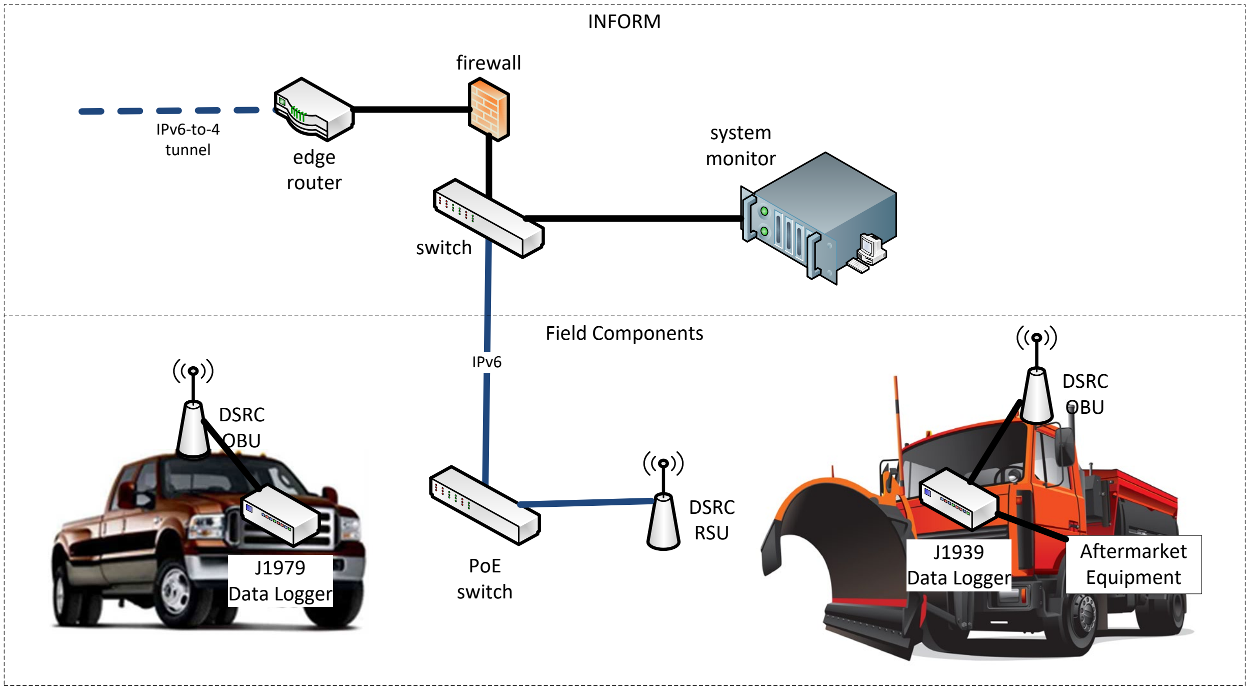


Figure 2 - Conceptual System Deployment (Source: Synesis Partners LLC)

3.1 Vehicle On-board Components

The vehicle on-board components are illustrated in Figure 3 for the light commercial vehicle (in this case, a Ford F250 or F350) and the heavy vehicle (a Mack truck). Connections to the DSRC OBE provide power, access to communication antennas, and data connections to on-board sensors. Sensor connections are made to the vehicle data buses (J1979 OBD-II on commercial light vehicles and J1939 on heavy vehicles); to the Dickey John plow and treatment equipment; and to the IceSight road weather sensors. The on-board equipment consists of the DSRC OBE, with the application software; the OBE's associated DSRC/Global Positioning System (GPS) antenna; the ChargeGuard vehicle power interface; a cable to connect the OBE to the vehicle data bus; a serial DE9 cable to connect the OBE to a Dickey John road treatment system (if present); and an

Ethernet serial cable to connect the OBE to an aftermarket IceSight sensor unit (if present).

Functionally, the OBE reads the desired data as specified in the application configuration file[9] from the CAN bus and other devices on the connected vehicle. The software formats those data into a CSV file that includes header information defining the tabular data. If the OBE is not within range of an RSE, as determined by the absence of a WAVE Service Announcement, the data are stored for transmission at a later time. When an OBE detects the presence of a WAVE Service Announcement, vehicle-derived data are transmitted in last-in-first-out order so that the most recent data are sent to the RSE first. Data files stored on the OBE are deleted upon successful transmission to an RSE. If storage on the OBE becomes scarce because of lack of contact with an RSE, the oldest data files are deleted to make room for newer data.

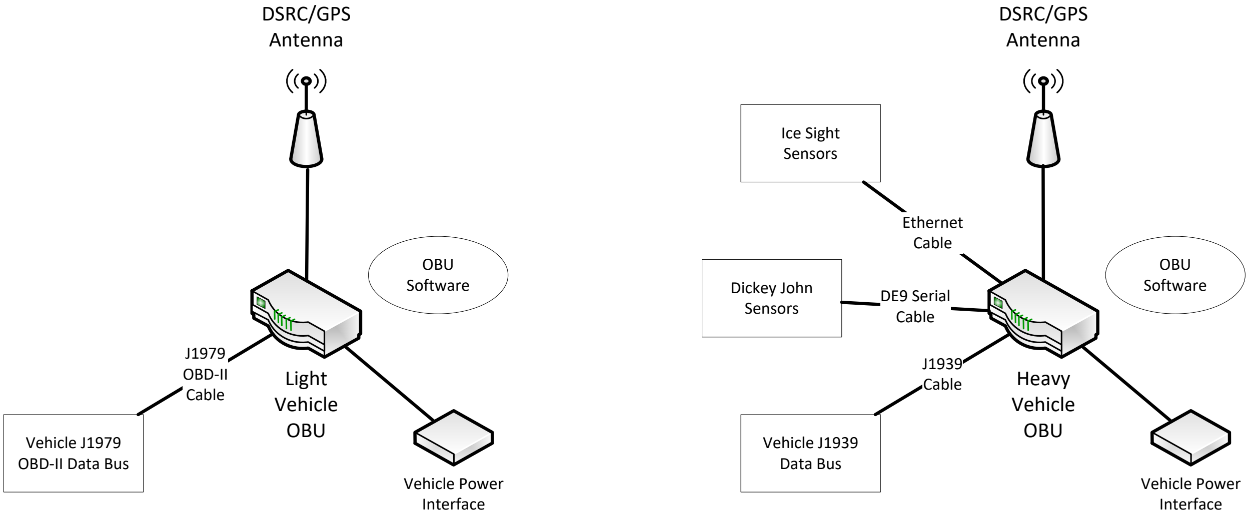


Figure 3 - Vehicle On-board Components (Source: Synesis Partners LLC)

3.1.1 OBE Hardware

The OBE hardware collectively consists of the components that together integrate DSRC radios to vehicle data sources. The core of OBE is the OBU itself. This device is an embedded computer with the processor, memory, storage, and application software in an aluminum enclosure that also contains interface hardware for DSRC radio, GPS, CAN bus, Ethernet, and serial data. DSRC and GPS radios are connected to a combined external antenna with quick-connect automotive standard terminations. Serial data sources are connected using common DE9 serial cables and Ethernet network connections use readily available RJ45 CAT5 (or higher) cabling.

---

[9] The desired data and sampling frequency are configurable by the system manager using the specification described in Section 3.1.4 of this document.

The OBU is connected to the vehicle CAN bus with cables assembled as described in the installation guide. There are two types of CAN bus cables: one for light vehicles that are terminated with an OBD-II style low profile connector, and the other with a heavy vehicle J1979 connector. The OBU side of the CAN bus cable is similar to the serial data connection, except that it is a DE9 male termination.

Power is wired directly to a ChargeGuard power management module through the exposed power supply wires from the CAN bus cable. The OBU input power is terminated via a 4-pin Molex connector directly wired to the power management module output.

Figure 4 - Heavy Vehicle OBE, Antenna and Cables (Source: Synesis Partners LLC)

Figure 5 - Light Vehicle OBE, Antenna and Cables (Source: Synesis Partners LLC)

3.1.2 IceSight

The High Sierra Electronics IceSight (Model 2020S) mobile road weather sensor device (Figure 6) procured for this project provides data to supplement those available from a vehicle's on-board sensors. The device is intended by NYSDOT for deployment on one of their heavy snow plow trucks and will provide road surface temperature, air temperature, relative humidity, surface state (dry, damp, etc.) and surface grip data. The IceSight device is connected to the OBU with an Ethernet cable.

Figure 6 - High Sierra IceSight 2020S (Source: Synesis Partners LLC)

3.1.3 OBE Software

The OBE software includes modules provided by the manufacturer and those developed specifically for this project. As shown in Figure 7, these software modules are the operating system; the WAVE Basic Service Set (WBSS) that supports the DSRC messaging; the Startup script that identifies and initiates the system modules; the Upload module that manages interactions with the RSE through the WBSS; and the Road Weather (RdWx) application that manages the data processing for the connected data sources.

The Cohda RSU procured for this project use an embedded GNU/Linux operating system for the firmware based on the version 2.6 kernel that includes BusyBox (combined set of standard command-line utilities for embedded systems) and Dropbear (small-memory footprint Secure Sockets Layer (SSL) utility that supports Secure Shell (SSH) server, client, and Secure Copy (SCP) functions).

The WBSS is supplied by the OBU vendor as part of the native OBU software package. It is also started and runs in the background when the device is powered on. Once running, the WBSS listens for RSE radio signals advertising the IP version 6 (IPv6) WAVE service and creates an IP network connection when an RSU is in range, and also initiates the upload component.

The OBU is powered on by the ChargeGuard power monitor a few minutes after the host vehicle is started. The OBU operating system executes the RdWx startup script that first checks for a newer version of the RdWx being present and installs it if present. It then initiates the RdWx application followed by the WBSS application.

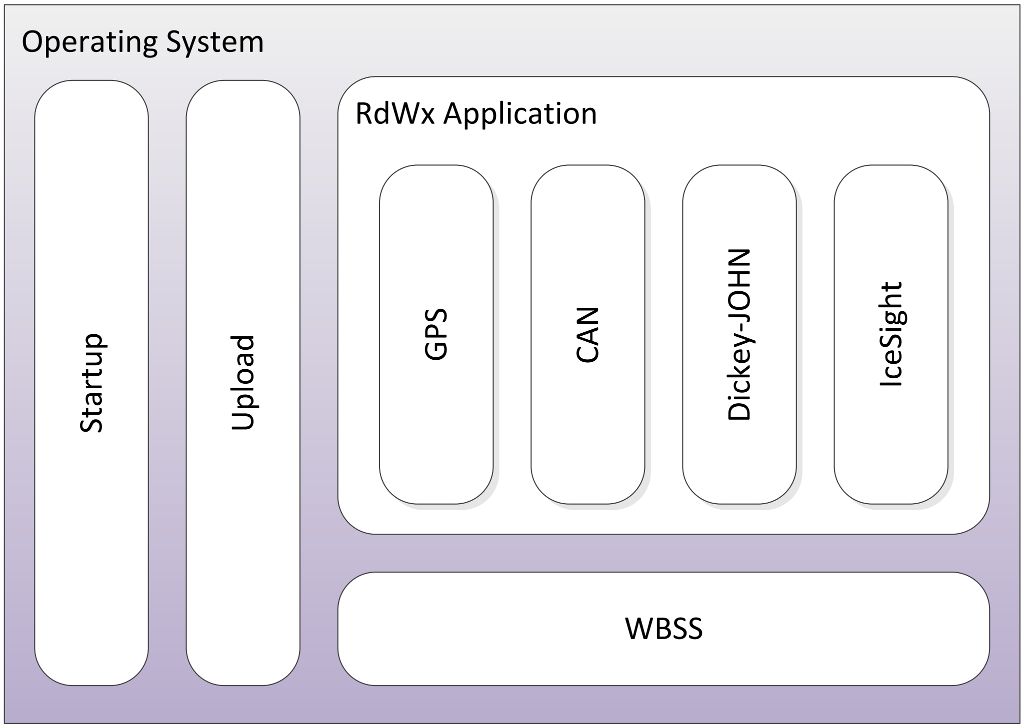


Figure 7 - OBE Application Software Modules (Source: Synesis Partners LLC)

The Upload component’s primary purpose is to connect to the RSU and upload collected data files, with the most recent data being sent first. The upload component also checks for application updates and downloads them when available. This check occurs a maximum of once per day so as to not interfere with data transmission but still enabling remote software updating to occur at reasonable intervals.

The Road Weather application itself consists of four independent modules that each independently manages data processing for the connected data sources: GPS, CAN, Dickey John, and IceSight. The startup script initiates the main RdWx application when the OBU is first powered on, first checking for and applying updated software. The main RdWx application continues to run in the background after startup is complete, collecting data from its configured data sources, aggregating the data into snapshots, managing the storage space on the OBU, and formatting the data into CSV files for upload.

Additional information on the OBE application data processing is provided in Appendix B.

3.1.4 OBE Interfaces

This section describes the OBE interface for the IP messaging application in terms of its configuration settings on the OBE. The OBU configuration settings are contained in a single configuration file consisting of six sections. The six sections each specify how the OBE should process data from particular connected sources:

- [RdWx] configures data selection, sampling frequency, and storage options for the main Road Weather application
- [GPS] sets the names associated with global positioning data
- [Dickie John] configures the reading of Dickie John Control Point® data
- [IceSight] configures weather measurements reported by the mobile pavement condition sensor
- [J1979] configures the weather-related data to be captured from the light vehicle CAN bus (using the OBD-II parameter identifiers [PIDs]) according to the SAE J1979 standard
- [J1939] configures the weather-related data to be captured from the heavy vehicle CAN bus according to the SAE J1939 standard

The order of the sections within the file is not constrained. The example configuration in Figure 8 suggests the order [RdWx], [GPS], [J1979], [J1939], [Dickey John], and [IceSight], reflecting a presumed likelihood that a particular device is available. The application and GPS are always present; otherwise no data would be captured. Only one of the CAN bus devices will be present at any given time. The SAE J1979 standard applies when the system is deployed on a light vehicle, and SAE J1939 when the system is deployed on a heavy vehicle. The Dickey John and IceSight equipment are completely optional. They provide additional data beyond those available from the CAN bus, but do not have to be present for the application to function properly.

Each of the configuration sections includes a parameter that indicates whether that component is active in this configuration. "Active" can be either "0" for an inactive component or "1" for an active component. Its purpose is to switch off individual components for testing purposes. While completely removing a section accomplishes the same goal—the component module won't be initialized—the "active" parameter setting enables retaining a standard configuration file while testing individual components. The RdWx and GPS components default to active, and the others default to inactive since they represent optional sensors as in the case of DICKEY-john or IceSight, or are deployment dependent as with J1939 for a heavy vehicle and J1979 for a light vehicle.

Figure 8 - Example OBU Configuration File (Source: Synesis Partners LLC)

```
[RdWx]
active=1
params=gps.time,gps.lat,gps.lon,gps.alt,gps.sat,can.airtemp,dj.all,is.airtemp1
delay=1000
records=600
prefix=
work=/dev/shm
save=/mnt/ubi/dbg/data
maxspace=30

[GPS]
active=1
params=gps.ts,gps.lat,gps.long,gps.alt,gps.fix,gps.sat,gps.spd
device=/dev/ttygps

[DICKEY-john]
active=0
params=dj.all
device=/dev/ttymxc1

[IceSight]
active=0
params=is.airtemp1,is.airtemp2,is.rh
address=192.168.1.180
port=1776

[J1979]
active=0
params=obd.airtemp,obd.airpressure
device=can0
# CAN id, filter, start bit, bit count, conversion
obd.airtemp=2024,034146,24,8,-40
obd.airpressure=2024,034133,24,8,

[J1939]
active=1
params=can.airtemp,can.airpressure
device=can1
# PGN, filter, start bit, bit count, conversion
can.airtemp=65269,,32,16,/32 -273
can.airpressure=65269,,1,8,/2
```

The component activation logic is straightforward: if the section is present, load the software; if the “active” parameter is “0” then stop; otherwise, continue and try to connect to the specified data device. Each device has a built-in timeout period so the software won't get stuck trying to collect data from a device that isn't configured properly or isn't connected.

The configuration file data specifications are shown in Table 3. The low-level configuration options for each section, such as device driver name or network address, have reasonable default values. The majority of the configuration effort is determining what data to capture and what label to give it. The "params" option is a comma-separated list of labels used to identify each piece of data being collected from a data capture device.

Some of the application devices report similar weather information. For example, air temperature is a common weather observation and is reported by the CAN bus as well as multiple times from an IceSight. Consequently, the data labels should be unique across the entire configuration file to distinguish among the data types and values. If two or more labels are the same, then the "last man wins"; if the IceSight section is configured to use "airtemp" as a label and the CAN bus also uses "airtemp" as a label, then the most recently polled device will determine the value associated with that label.

The RdWx "params" is the list of parameters identified in the other (for example, [GPS] and [J1979]) sections to be to saved and transmitted in the data file. The data labels used in the RdWx "params" option must match the labels identified in the other sections. The order of "params" in the [RdWx] section defines the order of parameters in the header and each record of the data file. While "airtemp, airtemp, airtemp" is a valid configuration, it will report the same value three times and isn't an especially useful feature.

For these reasons, it is recommended that data labels be prefixed with the device section. For example, three air temperatures values from three IceSight sensors could be labeled "icesight.airtemp1, icesight.airtemp2, icesight.airtemp3" and another air temperature from the CAN bus could be labeled "CAN.airtemp". This pattern distinguishes between data provided by particular devices and better describes the source of the data in the transmitted data file.

Table 3 - OBE Configuration File Parameter Specification (Source: Synesis Partners LLC)

Configuration Parameter	Description	Valid Values/ Default Value	Notes/Example
[RdWx]			
active	Enables or disables the component configured by the section. In the case of RdWx, it is an easy way for a user to disable the entire application without making OBU operating system changes.	0 for inactive, or 1 for active. Default value is 1.	active=1
params	The comma-separated list of labels to be included in the application output files. Labels must correspond to the labels specified in other modules.	Any text label that matches labels defined by the data capture components. Default value is nothing, so output files will be empty.	params=gps.time,gps.lat,gps.lon,gps.alt,gps.sat,j1979.airtemp,dj.all,icesight.airtemp1
delay	The number of milliseconds to wait between recording data in the output.	A reasonable number of milliseconds. Default is 1000, which is 1 second.	delay=10000 will record one set of data every ten seconds delay=100 will record 10 sets of data every second
records	The number of data records to include in each output file.	Default is 600.	This is related to the delay. If delay is set to 1000, then 600 records will record ten minutes (600 seconds) of data in each file. A very low number will result in many files being recorded, but few transmitted because the overhead of connecting wirelessly is much greater than the time to

			exchange the actual data file. A very large number creates few files but likely won’t successfully transmit because the exchange takes longer than a vehicle is within range of a RSU.
prefix	Optional alphanumeric string prepended to the data file name to simplify identification.	Default value is blank; maximum length is 16 characters.	This can be any readable text that represents useful information to the operating agency. It could be a vehicle identification number, for example. The data file name without the prefix will be the OBU media access control (MAC) address with a UTC date and 24-hour time, for example, “0A22C10B751D-20150630-1425.csv”. With a “prefix=TRUCK64”, the same file name would be “TRUCK64-0A22C10B751D-20150630-1425.csv”.
work	OBU local storage location for work-in-progress.	Default value is /dev/shm.	work=/dev/shm specifies a shared memory location where data can be rapidly accumulated and then written all at one time to the save directory for later transmission.
save	OBU local storage location for completed data files to be stored and from where to be transmitted.	Default value is /mnt/ubi/dbg/data	save=/mnt/ubi/dbg/data specifies onboard flash memory storage. A micro SD card can also be inserted into the OBU and then this configuration item can be changed to point there for potentially greater storage capacity.
maxspace	The total storage space,	Default value is 30 for 30	The default delay and record values will

	measures in megabytes, that the locally saved data files may occupy before the application begins deleting the oldest files	megabytes.	generate six files per hour with an estimated size of 100 kilobytes. A 30 megabyte local storage limit allows for 50 continuous hours of vehicle and OBU operation before needing to eliminate old files to free storage space. This should be sufficient time for the OBU to encounter an RSU to offload data files.
[GPS]			
active	Enables or disables the GPS.	0 for inactive, or 1 for active. Default value is 1.	active=1
params	The comma separated list of labels that uniquely identifies the data captured by this component.	There is no default value. The GPS component produces these seven data in order: timestamp, latitude, longitude, altitude, fix, satellite count, and calculated speed.	params=gps.ts,gps.lat,gps.long,gps.alt,gps.fix,gps.sat,gps.spd The timestamp is a 24-hour clock and measures to the nearest millisecond in the format YYYY-MM-DD HH:mm:ss.sss; e.g., 2015-06-30 11:58:23.014. Geo-coordinates are written in decimal degrees to six decimal places, fix is either 0 for no fix or 1 for good fix, satellite is the number of satellites used to calculate the position, and ground speed is in miles per hour computed from the changing position.
device	The logical name of the GPS operating system device.	Default value is /dev/ttygps	device=/dev/ttygps corresponds to the serial port that reads GPS coordinates in National Marine Electronics Association (NMEA) strings.

[Dickey-John]			
active	Enables or disables the DICKEY-john data capture component.	0 for inactive, or 1 for active. Default value is 0.	active=0
params	The comma separated list of labels that uniquely identifies the data captured by this component.	Default value is dj.all.	params=dj.all. The Dickey-John data capture component reads everything currently on the serial port connected to a DJ Control Point and exposes it as a single output/value.
device	The logical name of the operating system device from which data are gathered.	Default value is /dev/ttymxc1	device=/dev/ttymxc1 corresponds to the external OBU serial port.
[IceSight]			
active	Enables or disables the IceSight external sensor data capture.	0 for inactive, or 1 for active. Default value is 0.	active=0
params	The comma separated list of labels that uniquely identifies the data captured by the IceSight sensor.	There is no default value. The IceSight sensor records eighteen parameters: y voltage, x voltage, ratio of the voltages (y/x), air temperature 2, surface temperature, displayed condition code, Measured condition code, mnemonic for displayed condition, mnemonic for measured condition, displayed friction code number, measured friction code number,	params=is.airtemp1,is.airtemp2,is.rh Refer to Appendix C for the complete description of each parameter and its interpretation in the case of a lookup value.

		displayed friction code value, measured friction code value, dirty lens value, grip value, relative humidity, air temperature 3, air temperature 1.	
address	The IP address to listen on for capturing data from a connected IceSight sensor.	Default value is 192.168.1.180	The IceSight sends its data via TCP on address 192.168.1.180. and the OBU is configured to be on the same network 192.168.1.x.
port	The network port to listen on used in conjunction with the address configuration item.	Default value is 1776	Port 1776 is the logical port where the IceSight sensor sends its data.
[J1979]			
active	Enables or disables the light vehicle CAN bus data capture component.	0 for inactive, or 1 for active. Default value is 0.	active=0
params	The comma separated list of labels that uniquely identifies the CAN bus data captured by this component.	Default value is blank.	The text labels in this list correspond to the labels defined by the CAN bus data capture configuration items contained within the same J1979 section. If a label is in this list, then there is a CAN bus configuration of the same name. For example, “params=can.airtemp,can.airpressure” should have a corresponding “can.airtemp=” and “can.airpressure=” configuration items. Refer to the <name>

			description for additional detail.
device	The logical name of the operating system device from which data are gathered.	Default value is can0	device=can0 corresponds to the high-speed CAN bus port that includes a termination resistor and is configured for the light vehicle bus speed of 500 kbps.
<name>	A changeable configuration item that maps the "params" labels to the CAN bus data capture configuration.	There is no default value as these items only exist if they are included by the params configuration item.	This is a complex configuration item with its own embedded formatting. Refer to Figure 8 for an example. There should be one of these for each of the labels included in the params list. The CAN configuration is a comma-separated list in the order of CAN id, filter, start bit, bit count, and conversion. CAN data consist of an 11- or 29-bit identifier and 8 bytes of data. CAN id is the decimal representation of the desired CAN identifier. Filter is a variable length hexadecimal representation of a number that the CAN data must match before being accepted. There are 64 bits in the 8-byte value and the start bit references where the desired data starts. The bit count is the number of bits to extract from the data starting from the start bit position. The conversion configuration is a space-separated list of mathematical operations. Once the desired bits have been read from the data portion of the CAN message, the operations are evaluated on the resulting number from left to right. Each operation

			consists of a mathematical operator symbol immediately followed by an integer number. Decimal numbers are not allowed. Fractional conversions are achieved by multiplying and dividing, i.e. *5 /9. Refer to a published OBD-II parameter identifiers for more detail.
[J1939]			
active	Enables or disables the light vehicle CAN bus data capture component.	0 for inactive, or 1 for active. Default value is 0.	active=0
params	The comma separated list of labels that uniquely identifies the CAN bus data captured by this component.	Default value is blank.	The text labels in this list correspond to the labels defined by the CAN bus data capture configuration items contained within the same J1939 section. If a label is in this list, then there is a CAN bus configuration of the same name. For example, “params=obd.airtemp,obd.airpressure” should have a corresponding “obd.airtemp=” and “obd.airpressure=” configuration items. Refer to the <name> description for additional detail.
device	The logical name of the operating system device from which data are gathered.	Default value is can1	device=can1 corresponds to the CAN bus port that is unterminated and configured for the heavy vehicle bus speed of 250 kbps.

<name>	A changeable configuration item that maps the "params" labels to the CAN bus data capture configuration.	There is no default value as these items only exist if they are included by the params configuration item.	This is a complex configuration item with its own embedded formatting. Refer to Figure 8 for an example. There should be one of these for each of the labels included in the params list. The CAN configuration is a comma-separated list in the order of CAN id, filter, start bit, bit count, and conversion. Heavy vehicle CAN data consist of a 29-bit identifier and 8 bytes of data. CAN id is the decimal representation of the desired PGN. Filter is a variable length hexadecimal representation of a number that the CAN data must match before being accepted. There are 64 bits in the 8-byte value and the start bit references where the desired data starts, together with the PGN, is known as the SPN. The bit count is the number of bits to extract from the data starting from the start bit position. The conversion configuration is a space-separated list of mathematical operations. Once the desired bits have been read from the data portion of the CAN message, the operations are evaluated on the resulting number from left to right. Each operation consists of a mathematical operator symbol immediately followed by an integer number. Decimal numbers are not allowed. Fractional conversions are achieved by multiplying and dividing, i.e.

			*5 /9. Refer to the published OBD-II parameter identifiers for more detail.

3.2 Roadside Components

The roadside equipment procured, configured, tested, and deployed in this project is shown in Figure 9and consists of the DSRC RSU, GPS antenna, DSRC antennae, antennae surge protectors, RSU mounting bracket, 48VDC power supply, power-over-Ethernet (PoE) switch, and PoE Ethernet cable surge protector. Functionally, the RSE advertises IPv6 service over the DSRC broadcast, routes IPv6 traffic, receives data files from OBEs, sends data files when requested by backhaul network servers, and sends updated software on-demand to OBEs over DSRC.

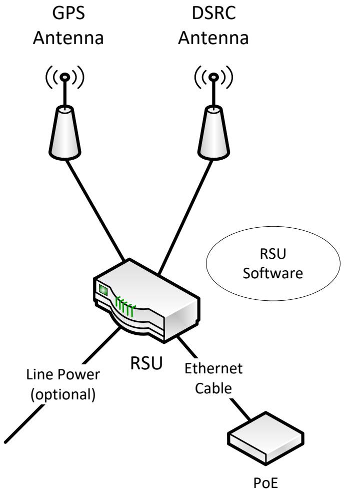


Figure 9 - RSE Schematic (Source: Synesis Partners LLC)

3.2.1 RSE Hardware

The roadside equipment includes the DSRC radio roadside unit and the infrastructure support components providing structural and power. The DSRC-enabling components consist of the DSRC RSU with its GPS and DSRC antennae. The DSRC RSU consists of a weather-proof enclosure that houses the processing, memory, storage, DSRC radio, and power electronics, with protruding connectors for antennae, power, and the local wired network. There are four external DSRC antennae mounts to support various antennae configurations—two paired antennae on the top or bottom of the enclosure are most common, with unused antenna connectors capped. There is one GPS antenna connector at the top of the enclosure. RSUs can be powered either through the Ethernet cable connected to a PoE switch or wired directly to line voltage. The Ethernet port is also used to connect the RSU to the agency network or local network equipment for stand-alone operation. Infrastructure support equipment consists of the mounting brackets, electrical surge suppressors (both antennae and Ethernet), and power supply with PoE network switch.

Figure 10 - RSE, Ethernet Switch, and Bracket/Surge Suppressor (Source: Synesis Partners LLC)

3.2.2 RSE Software

At its core, the RSU is a single-board computer interfaced to dual DSRC radios with network software based on an open-source Linux operating system. The Linux OS is configured to provide network hardware (wired Ethernet and DSRC radios) with IPv6 addresses, default routing information, and firewall rule enforcement. The OS also includes widely-available secure shell (SSH) and secure copy (SCP) applications that enable remote interaction and file transfer with each RSU.

Most of the software functions used by the road weather application are provided by the built-in Linux capabilities. The RSU IPv6 application enables OBUs within range of an RSU to use IPv6 network features. The IPv6 application broadcasts the radio-side IPv6 address and network. Nearby OBUs receive the IPv6 network information and dynamically set their network address and routing information. The IPv6 application also monitors radio data and repackages the DSRC data for non-DSRC (back-office) network transport.

From the perspective of the OBU, the RSU advertises the availability of the IPv6 service, routes network traffic, receives collected data files, and supplies updated road weather application software. The IPv6 application enables the network connection, but file transfer between the OBU and RSU is handled by the SSH and SCP applications provided by the operating system on both devices.

From the perspective of a RSE managing organization, each RSU is a router that can be contacted through the wired network to modify configuration parameters, retrieve accumulated data files and move them to other servers, and send new road weather application software for distribution to roaming OBUs.

3.2.3 RSE Interfaces

The RSE interfaces provide access to the data received by the RSE from vehicles and monitor the state of operations ("health") of the RSE. Health monitoring on the RSEs is provided by three applications built over the IPv6 services that send information to the back office: file upload, heartbeat, and alarm.

The road weather application software includes two complementary scripts that reside on each RSU and are configured to manage the collected data files.

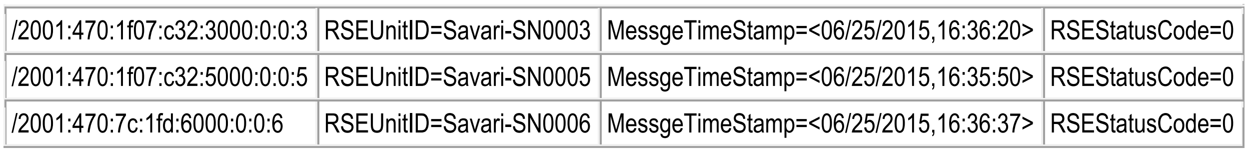
The first script is enabled by default and scheduled to run once daily. It accepts the source data directory and threshold file count as inputs. The executing script reads the data file storage directory and, when there are more files than the specified threshold, removes excess files in oldest first order to prevent overrunning the limited storage space. The default threshold file count limit is 300, which is about 30 MB and imitates similar OBE local storage limits.

The second related script pushes collected data files to another network destination and cleans them up once successfully transmitted, but is not enabled by default. To enable this function, the script accepts the destination network address and source data directory as input. The operating system scheduler must then be configured to run the script at regular intervals—every five or ten minutes is reasonable timing. The receiving server also needs the RdWx user added with the appropriate public key from the RSU RdWx user.

Each RSU records a system log file and a network traffic log file which are regularly offloaded to a defined network address through the file upload application. This allows the detailed log files to be recorded continuously and analyzed as needed while minimizing RSU storage space. The RSU 4.x firmware disables the log files by default so an operating agency can opt-in to collect them rather than cause RSU storage problems when the agency is unaware that the RSU storage needs to be freed regularly. As such, the log file gathering should be enabled if detailed monitoring is desired.

The heartbeat application transmits a simple device status message that consists of the RSU identifier, time stamp, and a coded status number that indicates potential problems such as low storage. The heartbeat message is sent once per minute using UDP. Although UDP does not guarantee delivery, its messages are easy to process and rarely get dropped, making it an excellent choice for heartbeat monitoring. There are a variety of open-source monitoring applications for the back office that can be configured to listen for these messages and send email alerts to appropriate personnel when problems arise and need correction. For this project, a simple web server was configured to listen for the heartbeat messages and put them on a web page to be viewed regularly. This was sufficient for the two-unit RSE deployment. Table 4 is an example of the received data from project deployed RSU.

Table 4 - Example Heartbeat Message Records (Source: Synesis Partners LLC)



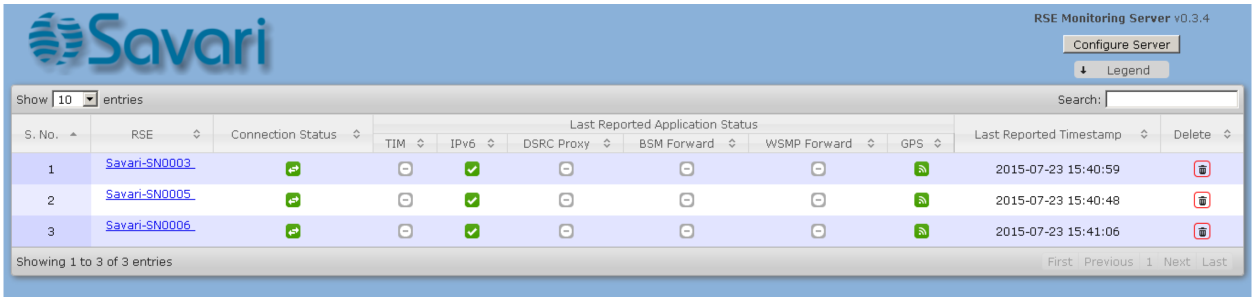
/2001:470:1f07:c32:3000:0:0:3	RSEUnitID=Savari-SN0003	MessgeTimeStamp=<06/25/2015,16:36:20>	RSEStatusCode=0
/2001:470:1f07:c32:5000:0:0:5	RSEUnitID=Savari-SN0005	MessgeTimeStamp=<06/25/2015,16:35:50>	RSEStatusCode=0
/2001:470:7c:1fd:6000:0:0:6	RSEUnitID=Savari-SN0006	MessgeTimeStamp=<06/25/2015,16:36:37>	RSEStatusCode=0

The alarm application is similar to the heartbeat application in that it sends short messages via UDP. The alarm message includes the status of installed applications, and if they are configured to run, disabled, or report a failure condition. The Savari RSU manufacturer provides an alarm monitoring application based on Java with configuration documentation.

3.3 *Back Office Services*

The back office services play a passive role in the system as deployed in this project. Back office services receive the data and the health monitoring heartbeat and alarm messages pushed from the RSEs. Server locations to which the messages are delivered are configured on the RSEs.

A script or application similar to the RSE push script could be created to gather the distributed data files by pulling them from the RSEs. This application would be configured with the list of RSEs to contact and also should clean up the remote files when successfully received. In this case, the username and public key used by this application would need to be applied to the originating RSEs.



Show 10 entries

Search:

S. No.	RSE	Connection Status	Last Reported Application Status						Last Reported Timestamp	Delete
			TIM	IPv6	DSRC Proxy	BSM Forward	WSMP Forward	GPS		
1	Savari-SN0003	⇄	⊖	✓	⊖	⊖	⊖	📶	2015-07-23 15:40:59	🗑
2	Savari-SN0005	⇄	⊖	✓	⊖	⊖	⊖	📶	2015-07-23 15:40:48	🗑
3	Savari-SN0006	⇄	⊖	✓	⊖	⊖	⊖	📶	2015-07-23 15:41:06	🗑

Showing 1 to 3 of 3 entries

Figure 11 - Data File List on Savari Monitoring Server Application (Source: Savari Inc.; Synesis Partners LLC)

4 DEPLOYMENT

Deployment of the DSRC-based system for collecting road weather data for this project consisted of three related sets of activities: installing the roadside and vehicle on-board equipment, configuring the supporting IPv6 backhaul network connections, and testing the connections, data transport, and interfaces. As discussed earlier, however, the actual scope of deployment was somewhat modified from the original plan to place the focus more squarely on the DSRC capability rather than on any application-specific back office data exchange. This section describes the deployment insofar as it was completed to demonstrate the system capabilities, with the installation guidance and test plans as will be needed when an agency might complete the deployment.

4.1 Testing

The objective of system testing is to demonstrate that the system performs its intended functions as described in its concept of operations and specified by the user needs and requirements. Testing will assess both nonconformance with the stated requirements and any unexpected or undesired side effects of system operation. Just as the functional requirements specify behaviors of particular system components, testing will need to be performed on particular components as well as the fully integrated system. The behaviors of even moderately complex systems are nearly impossible to adequately test when fully assembled.

The system Test Plan[10] describes the overall approach to testing the application. It is based on the user need and requirements and is complemented by test scripts that reflect the specific testing needs based on implemented system features. The Test Plan first identifies system elements—i.e., the components and processes to be tested—and the testing methods. It then describes the test cases and test scripts at a high-level, relates the test cases to requirements, and relates the test cases to project tasks during which particular tests are to be performed. The Test Plan also identifies activities needed to configure test environments, if any. Eight test cases are described in the Test Plan. Of these, five have been completed as unit and integration tests during the system development; three will need to be completed as part of the eventual field deployments.

Test Case 1 confirms that the OBE acquires weather-related data according to the OBE application configuration settings from the various connected vehicle data sources. This testing has been competed for the GPS, the light vehicle CAN bus, and the IceSight unit. Installation testing will be needed for heavy vehicle and Dickey John data connections. When executing the test procedures for this test case, the completed OBE data files are

---

[10] *5.9 GHz Dedicated Short Range Communication Vehicle Based Road and Weather Condition Application Test Plan*, December 2013, Synesis Partners LLC.

inspected to assure that the anticipated values are present. Values are then compared to the vendor sensor specifications, where available, to verify that weather-related data values are within the range of a sensor and make sense given the local environmental circumstances. An air temperature of 140 F would, for example, indicate a problem.

Test Case 2 evaluates the transport mechanisms of the vehicle weather-related data over DSRC. Data files generated on an OBE and transmitted to an RSE are retrieved from RSE. Analyzing the RSE log file against the data files confirms that DSRC transmission was successful since that is the only means for the OBE to deliver the data to the RSE. This testing was performed during development and in the initial RSE and OBE configuration and test operations.

Test Case 3 assures that the data collected on vehicles is sent to RSEs and retains the ability to associate individual data records with the source OBE according to the specification encoded in the OBE's configuration file. This testing was performed during development.

Test Case 4 uses the inspection testing method to verify that OBE-provided data are available on and can be retrieved from the RSE. Files on the RSE are pushed to a remote server as configured on the RSE; the contents of the files are inspected to ensure that they are uncorrupted and complete. This testing was performed during development.

Test Case 5 verifies that monitoring software correctly identifies system problems and notifies administrators to fix the problems. Malfunctions are simulated by moving processed data files and heartbeat logs from their expected locations. System monitoring results are then compared to the malfunction characteristics to verify that the error condition was correctly identified. This testing was performed after development of the Heartbeat Monitor software.

Test Case 6 verifies the field installation and operation of OBEs. Its activities are similar to Test Cases 1 and 2 in that they verify OBE operation, but the testing environment is moved to the field. The Test Case 6 testing environment includes light (HELP) vehicles and heavy (snow plow) vehicles with Dickey John equipment and IceSight units. This OBE field installation test case also evaluates independent power control hardware (the Charge Guard unit) that protects the OBEs from noisy vehicle power supplies and prevents OBEs from discharging vehicle batteries when vehicles are not operating. This testing should be performed in conclusion of the OBE field installation and is described in the OBE installation guidance.

Test Case 7 verifies the field installation and operation of RSEs. Just as Test Case 6 shares common activities with previous test cases, Test Case 7 shares some aspects of Test Case 3. The RSE field testing environment consists of the RSEs deployed to the LIE sites, relying on power supplied from PoE switches installed in cabinets, and using the

NYSDOT INFORM network. This testing should be performed in conclusion of the OBE and RSE field installation and is described in the RSE installation guidance.

Test Case 8 verifies the end-to-end system deployment. At this stage, it is important to verify that the remote power cycling capability (used to reboot RSEs when necessary) is functional, reducing the need for on-site maintenance. This test case evaluates OBEs continuously collecting sensor data from operating vehicles and successfully detecting RSEs under moving conditions. Test Case 8 also evaluates each component's ability within the system to store and forward information when some components are unavailable. This testing should be performed in conclusion of the OBE and RSE field installation and is described in the installation guidance. Successful performance of Test Case 8 will fundamentally confirm that data originating on a particular vehicle are retrievable from the RSEs, and secondarily demonstrate system coverage and capabilities. For example, Figure 12 depicts the path travelled by a test vehicle (blue line) when in range of a deployed RSE (red dot), and implicitly illustrates the coverage provided by that RSE.

Figure 12 - Typical RSE Deployment Validation Result (Source: Synesis Partners LLC)

4.2 *Backhaul Network and IPv6*

The purpose of this section is to describe the implementation of the IPv6 network for this 5.9 GHz DSRC Vehicle Based Road and Weather Condition Application and for similar projects. The section summarizes the networking needs for connected vehicle applications and the case for using IPv6 instead of the more established IPv4 standards and describes the project experience with deployment of IPv6 support for the CV application.

4.2.1 CV Application Needs and the IPv6 Network Challenge

CV application performance requirements particular to vehicle-based networking with DSRC have resulted in the network implementation decision to use IPv6 rather than the more established IPv4. The major concerns, driven largely by the brief intermittent connections needed between vehicles, are to support large numbers of connections, minimize network connection time, assure security of transactions, and protect privacy. Some CV may require connections from vehicles to infrastructure, then through an agency network to external servers, and then back to the vehicle, necessitating an IPv6 connection through the network from end to end.

A synopsis of Internet Protocol history is helpful in providing some context for the discussion. IPv4 dates to the 1970s and became the basis for the existing Internet deployment during the rapid growth of the Internet in the 1990s. The IPv4 addressing scheme uses 4-byte network addresses to yield approximately 4.2×10^{9} directly addressable network nodes—fewer than the number of humans on Earth. As such, schemes such as network address translation were used to extend the number of nodes available for addressing, with the consequence of increasingly elaborate network management and routing. The network management standard practice has become "switch where you can, route where you must"—meaning to stay whenever possible within trusted local networks. Hardware and software advances have alleviated some of the pain of growth, but have largely maxed out network performance within the inherent constraints of IPv4.

IPv6 was developed in the 1990s primarily to expand the available Internet address space, and also to simplify network routing configuration, enable multi-cast data streams (the IPv4 mechanism was not interoperable), and to support improved security algorithms. Network address lengths were increased from the four bytes in IPv4 to sixteen bytes in IPv6, yielding a network capacity of approximately 3.4×10^{38} addressable nodes. This increase in the address space could create significant changes in the way networks are managed and operated, with a corresponding variety of mechanisms for transitioning from IPv4 to IPv6.

With this context, it becomes clearer why IPv6 is a better fit than IPv4 for CV applications. IPv6 accommodates larger numbers of vehicles more easily because of the larger number of addresses available on the network and within any particular subnet. It furthermore simplifies addressability by enabling connections between more network nodes without having to translate addresses between subnets. IPv6 enables faster connections because each node (radio/vehicle) is individually addressable (using the nearby RSE router portion of its IPv6 address combined with the device radio hardware identifier) and does not rely on subnetwork address assignments as with IPv4.

IPv6 nonetheless has its own deployment and operations challenges. As described earlier, IPv6 network domains are bigger than ITS IPv4 networks, with associated more complex management needs. IPv6 addressability creates more opportunities for unauthorized access; IPv4 created multiple subnetworks to manage, but protected us from ourselves (through dynamic host configuration protocol (DHCP) and network address translation (NAT)). The so-called "Internet of Things" depends on enabling IPv6 devices to communicate directly with other IPv6 devices. Consequently, IPv6 requires diligent network planning and policies. Typical considerations include determining what type of network traffic is allowed from which sources and to which destinations, and establishing proactive security management and monitoring to protect devices from potentially harmful interactions (rather than depending at least in part on DHCP and NAT to obscure the network for IPv4). These considerations may present new challenges for network managers not yet trained or familiar with IPv6 deployments.

The operational challenges virtually assure that IPv4 and IPv6 networks will coexist thru a transition period. Agency networks for ITS do not have to have to be converted to IPv6 before CV technologies are deployed. Most recent networking equipment is capable of dual-stacked IPv4 and IPv6 protocol support, providing flexibility for network managers to use the most appropriate protocol for each purpose and device. The new operations mantra is "dual-stack IPv4 and IPv6 where you can, tunnel IPv6 where you must, avoid translation between the two."

4.2.2 IPv6 Network Deployment Planning and Project Experience

Setting up the IPv6 network for supporting CV applications starts with obtaining the external address for the network from an IPv6 service provider (ISP). Up to 65,000 subnetworks would typically be made available to enable IPv6 applications, and ISPs may provide more or fewer to meet the particular need. In cases of unusually large network needs, multiple independent IPv6 addresses could be requested with each one representing 65,000 subnetworks. Each DSRC RSU acts as a router for one of the assigned subnetworks with its own connections to nearby OBUs. The RSU's routing capabilities enable connected OBUs to contact remote applications and services.

Deployment of the IPv6 network does not necessarily mean a complete redeployment of all network equipment. Most networking equipment manufactured within the last ten years supports both IPv4 and IPv6 addressing, and it will be beneficial to dual-stack wherever possible. This strategy enables both current IPv4 ITS equipment and potentially new IPv6 CV equipment to connect to the same routers and switches; a DSRC RSE along a freeway, for example, could be connected to a switch originally deployed to support a message sign or vehicle detection station. The RSE inside-network-interface is itself dual-stacked.

IPv6 connections between network nodes similarly need not depend on dedicated IPv6 infrastructure. Not all ISPs currently supply native IPv6 connection, and may themselves be using tunnels through IPv4 connections to provide IPv6 service. Tunneling wraps IPv6 data packets with an IPv4 header for transmission through the IPv4 network. The header is stripped off at the end of the tunnel for routing to its eventual IPv6 destination. The process requires configuring the tunnel at its IPv4 ends, but generally does not any significant transmission delays.

An alternative means of supporting IPv6 data traffic through an IPv4 network is to translate or rewrite the IPv6 packets as IPv4 and then to retranslate the packets at their destination. Although technically feasible, translation of this type is complex to configure and support, can degrade performance, and should general be avoided.

Deploying RSEs on the IPv6 network is similar to deploying a Wi-Fi connection on an IPv4 network, and will require similar security provisions to protect the network from potentially misbehaving end-user/vehicle and network applications. Figure 13 illustrates the general concepts. As with any existing Internet connection, the IPv6 connection should be protected by an appropriately configured firewall. The firewall on the RSE enables and disables routing of messages to and from particular destinations that could include an agency's own CV services or those of third-party providers on the far side of the agency IPv6 Internet connection. The four connection paths (IPv4 in and out, IPv6 in and out) on each RSE firewall necessitate multiple configuration sets to be managed. Anti-lockout rules on the firewall are used to keep network administrators from inadvertently disabling access.

A few notes on distinctions between and use of networking components…

A **router** connects an organization's inside network to the Internet. IPv6 tunneling is typically configured here.

A **firewall** enforces an organization's network traffic policies. Its configuration settings define which addresses and protocols are allowed access to the network.

A **switch** physically connects network devices within a subnetwork. It tracks hardware addresses by physical ports for fast packet movement. RSUs in a CV network can be isolated using a virtual local area network (VLAN) or dedicated switches.

On the other end, CV application services are likely to need connections into and through an agency IPv6 network. Externally hosted services such as a Security Certificate Management System (SCMS), for example, may need to be able to send messages to vehicle OBEs connected to RSEs on the network. This will generally be managed by configuring the network edge router for specific IPv6 addresses. This will necessitate one route table entry per RSU, manually or script generated, and may in the near term present management challenges. Internally hosted services would avoid the need to configure the edge router for all RSEs. Every RSE network interface will need to be configured to accept traffic from application services whether inside or external to the agency network.

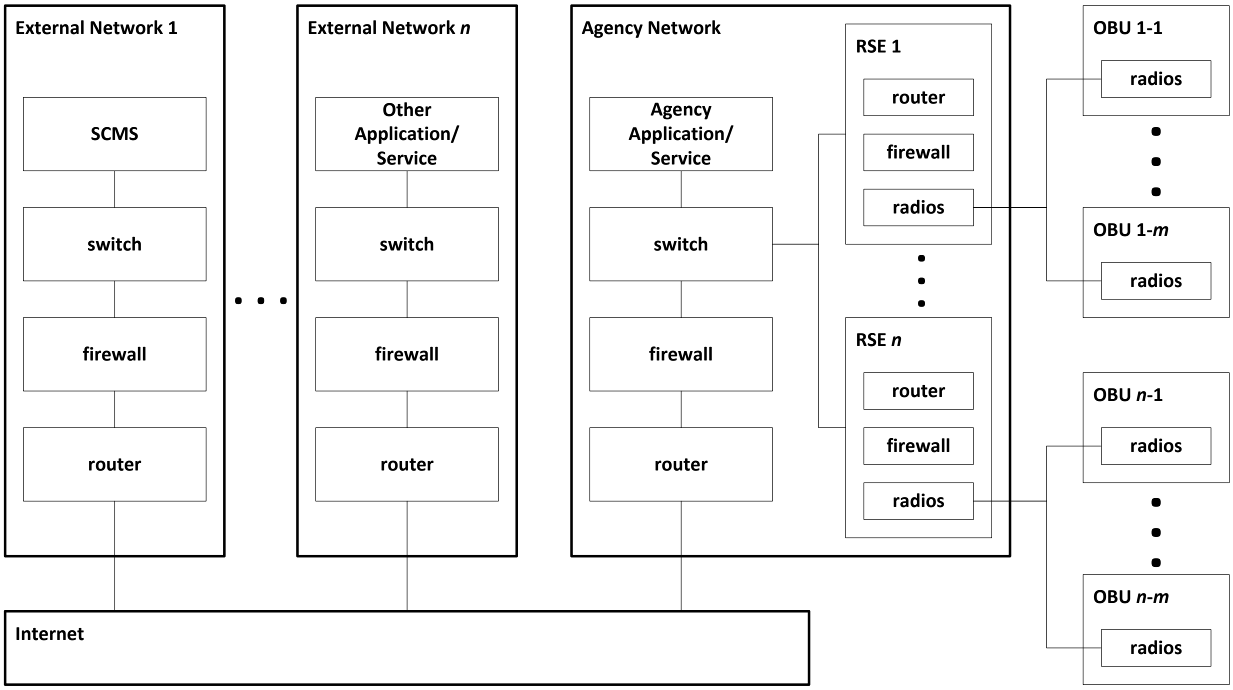


Figure 13 - Conceptual IPv6 Network for CV Applications (Source: Synesis Partners LLC)

The IPv6 network deployment and configuration experience for this project reflects and amplifies the general planning considerations. The Long Island Expressway (INFORM) network infrastructure is operated by Hinck Electric for New York State DOT. Hinck Electric personnel were very familiar with the network components, their configuration, and IPv6 concepts. This made the deployment of IPv6 needs for RSE support relatively seamless.

The provider of the public IP address to INFORM did not currently provide IPv6 addresses, so a tunneling solution was chosen. Hurricane Electric (tunnelbroker.net) provides free IPv6 over IPv4 tunneling, as well as configuration guides for a wide variety of common network routers. An IPv6 tunnel was configured to operate between the back-office servers and the INFORM network using the Hurricane Electric web portal. The necessary configuration settings for the INFORM side of the IPv6 tunnel were sent to Hinck Electric personnel for implementation on the Internet router.

The tunnel was then tested using the ping utility for the tunnel end-points since RSEs had not been deployed at the time. The initial tests revealed that the tunnel was configured correctly, but that a firewall configuration needed to be updated to allow the traffic at the back-office server side. Once the firewall modification was completed, all the tunnel communication tests passed.

In addition to the IPv6 tunnel configuration, an access control list (ACL) was also put in place on the INFORM router to only allow traffic between their network and the back-office network. Related to this basic network security precaution, the RSUs to be deployed were configured to send their data only to the back-office server IPv6 address, and their individual firewalls were set to allow only secure traffic.

Since an IPv6 tunneling solution was chosen, each RSE inside-network interface was configured to have a maximum transmission unit (MTU) size of 1280. This ensures that data packets easily traverse the IPv6 tunnel without the need for intermediate routers to split single packets into two or more packets for transmission. The smaller MTU originating from an RSE also helps reduce unexpected network behavior such as secure connections being dropped or appearing to hang.

Enterprise-class network equipment can have complex configuration characteristics, some of which may affect IPv6 tunneling. In the INFORM case, RSEs worked flawlessly with an MTU of 1500 until the ACL was applied at the Internet edge router. It was then necessary to use an MTU of 1280 with the ACL—which was modifying the original data packets in unexpected ways when verifying them against the allowed IPv6 addresses.

The INFORM network switches were capable of dual-stacking to simultaneously support IPv4 and IPv6 hosts. This greatly facilitated deploying the RSEs. Each RSE was assigned both an IPv4 address and an IPv6 address. The IPv6 addresses were only reachable between the back-office and INFORM networks, and the IPv4 addresses were only accessible from within the INFORM network.

Setting both IPv4 and IPv6 addresses is an important consideration for the RSE. Externally connected equipment, such as signal controllers, may only support IPv4, and having an IPv4 address on the RSE facilitates retrieving data from that external equipment. Having an IPv4 address can also be beneficial during RSE firmware upgrades. Firmware updates frequently modify network settings, and the IPv4 address provided another mechanism to connect with the RSE should there be unexpected results.

4.3 In-Vehicle and Roadside Installation

Installation of the in-vehicle and roadside components of the road weather application is described in the *Installation Guide*[11]. The *Guide* provides a description and list of the system parts; identifies the immediate tools and resources needed for installation; describes the step-by-step on-board equipment installation; describes the step-by-step roadside equipment installation; and provides system installation testing instructions. Appendices to the *Guide* provide instructions for installation tasks that might be needed for deployment of units beyond those procured and provided for the NYSDOT

---

[11] Connected Vehicle Pooled Fund Study, "5.9 GHz Dedicated Short Range Communication Vehicle-based Road and Weather Condition Application – Installation Guide," prepared by Synesis Partners LLC, Version 1, August 2015.

deployment: construction of OBU-to-OBD-II cables for light vehicle installation; construction of OBU-to-J1939 cables for heavy vehicle installation; the ChargeGuard configuration; and OBU software installation.

5 ANALYSIS AND RECOMMENDATIONS

This project has successfully developed and demonstrated a capability for aggregating weather-related data from a variety of original and aftermarket vehicle on-board sensors, sending the data over a DSRC connection from the vehicle to the roadside, and making the data available to other agency systems. This section describes some of the challenges and opportunities encountered in the project, and offers recommendations for future consideration.

5.1 *Messaging Standards*

DSRC messaging standards offer three options for getting probe data from vehicles to the roadside, from where they can be provided to other applications and data services: probe vehicle data messages, Part II of the basic safety messages (BSMs), and IP datagrams. Although any of these options could presumably be used to meet the data messaging objectives of the application, each option has its challenges.

There is generally a lack of J2735 probe vehicle data message support on current generation RSEs, including the Savari RSU 3.2 procured for this project. The most recent April 2015 version of the J2735 DSRC Message Set Dictionary identifies but does not provide any specific message formatting for the probe vehicle data message.

BSMs are broadcast continuously by the OBE and would require an OBE store-and-forward application in order to provide continuous operational coverage. If that capability were developed, accessing BSMs received by an RSE would require a store-and-forward capability as well. The store component could be accommodated by the log file, but requires processing of the log file to extract only the BSM probe data content. Alternatively, a BSM forwarding application on the RSE could be used to specifically store and forward only the BSMs. BSM forwarding on the Savari 3.x RSE was specific to Safety Pilot and not necessarily supported on other RSUs.

IP messaging is supported by the RSE 3.x and 4.x standards, but requires an OBU application as developed in this project to originate messages, rather than using native DSRC J2735 messages. This is a practical solution to the project objectives and constraints, but somewhat misses the point of having the DSRC standards for probe data.

Recommendation: Agencies wanting to deploy DSRC-based applications should monitor ongoing standards development and deployment to assure that viable probe message capabilities are included. Probe message capability and support (in one of the three forms described above) can be included in DSRC equipment procurements, even if applications for the data are not fully developed.

5.2 OBU and RSU Software Portability

Although the road weather application developed for this project has been deployed and tested on specific manufacturer versions of OBEs and RSEs, it may become desirable to deploy the application on devices from other manufacturers as well. The Portable Operating System Interface (POSIX) is a group of standards maintained by the IEEE Computer Society that specifies consistent application programming interfaces between operating systems in order to maintain compatibility. It appears that both OBE and RSE vendors have consistently used variants of open source GNU/Linux operating systems as the basis for their device software and GNU/Linux is POSIX-certified.

The road weather application itself is implemented through a combination of operating system scripts and compiled standard C++ instructions. Because the application is simple, and both it and the operating system adhere to POSIX standards, it should be possible to recompile the source code to run on a different OBE, under the condition that the vendor of the different OBE provides the necessary cross-compiler software. Cross-compiling (using a computer with one type of central processing unit (CPU) to create software for a computer with a different CPU) is a fairly common practice. If a vendor won't supply the cross-compiling software, they should be able to cross-compile the RdWx application if requested.

The most critical questions for porting the road weather application to another OBE are whether the target OBE (1) contains a component that detects RSE IPv6 application announcements, (2) can configure the OBU IPv6 network settings, and (3) can be adapted through a script to notify the RdWx upload script to perform its function. Beyond this, every vendor that bases its device software on a POSIX-compliant GNU/Linux variant should be able to support the functions needed for the RdWx application.

Recommendation: Agencies considering deployment of the road weather application developed in this study should, at a minimum, assure that the OBEs and RSEs to be used are POSIX-compatible and that the OBE meet the three configuration conditions.

5.3 IPv6 Network Support for CV Applications

Deployments of DSRC RSEs have numbered generally in the tens of devices. This deployment is relatively small given that three version 3.2 RSUs were procured—two for deployment and one as a drop-in spare or for additional deployment. OBU numbers are similar small with six devices being procured—five for deployment and one as a spare. The limited number of devices greatly reduces the burden of configuring the devices and the network. However, upcoming pilot deployments could include hundreds of devices, and repetitive one-off manual configurations will not be practical or recommended.

RSUs are network routers. The IPv6 Internet of Things (IoT) has made the managing agencies (or their contractors) responsible for expanding network access and administration beyond the traditional roadside cabinet. The management entity must ensure that critical network systems are protected while providing unprecedented public access to and across its network.

Ideally, installers should be able to hang it up, plug it in, and walk away. If planned correctly, RSEs should be able to be deployed in a shoot-then-aim fashion—deploy it to the field and, as long as it has power and is connected to the network, test and configure it remotely.

In addition to being a network router, each RSU maintains a firewall on both the inside network and outside radio interfaces; it transmits wave service advertisements; broadcasts active messages; and receives and forwards field data. Managing a network of RSUs with this diversity of functions will necessarily be complex. To facilitate RSE management, the RSU 4.0 specification describes in detail the management information base (MIB) for Simple Network Management Protocol (SNMPv3), which enables COTS network management systems to monitor RSU hardware and configure its functions.

One item that the RSU 4.0 Specification does not define is the use of DHCPv6 with prefix delegation. Each RSU has at least one IPv6 address on the inside network interface and, depending on application needs, may also have up to two more IPv6 addresses—one for each of the radios. If the RSU vendor supports DHCPv6 with prefix delegation, it is possible to configure the agency router so that the RSU automatically acquires all the needed addresses and the agency routing tables are updated accordingly. For this road weather application study, routing information was updated manually as there were potentially only three RSU in total and the 3.2 firmware did not support DHCPv6 with prefix delegation.

If the procured RSU hardware does not support DHCPv6, one possible alternative is to deploy software that acts as a surrogate on behalf of both agency routers and RSE. Every IPv6-enabled device generates a link-local IPv6 address based on the manufacturer-assigned Ethernet Media Access Control (MAC) identifier. By definition, the agency router and RSE must be connected to the same virtual LAN (VLAN) and associated set of network switches. Therefore, surrogate software is able to request IPv6 addresses from the router, provide those addresses to RSE, and inform the router to update its routing table information.

Once the RSE has a routable IPv6 address, COTS network management software can be used to manage and configure the remaining RSE configuration items depending on managing agency application requirements. It is also possible to use COTS network management software to configure both RSE and routers through the same link-local network address mechanism without the need for DHCPv6. However, the COTS network management software should be carefully specified so that it can automatically assign IPv6 network addresses and update routing tables. Without attention to these functions, network management software tends to degenerate into simply a display for manually updating thousands of network entries.

Experience updating firmware suggests that it is best to deploy first on a spare device at the agency to avoid unworkable surprises, like the needed IPv6 address being masked by a new tunnel configuration thereby cutting off contact with the unit.

The only equipment failures during the initial field deployments have been the PoE switches and fiber transceivers—the RSE has been more robust than the COTS equipment.

Recommendation: An agency planning for deployment of an IPv6 network to support DSRC devices should have in-house and/or contract network administrators with IPv6-specific training and experience. The agency should also consider the use of SNMP and/or related network management tools to assist in the configuration, management and operations of the deployed network and devices.

APPENDIX A - DEFINITIONS

The following table provides the definitions of all terms, acronyms, and abbreviations required to properly interpret this System Test Plan.

Term	Definition
ACL	Access Control List
BSM	Basic Safety Message
CAN	Controller Area Network. An electrical specification and signaling protocol developed by Bosch to facilitate simple data communication between connected equipment control units.
ConOps	Concept of Operations
COTS	Commercial off-the-Shelf
CPU	Central Processing Unit
CSV	Comma-separated Value
CV	Connected Vehicles
DHCP	Dynamic Host Configuration Protocol
DHCPv6	DHCP for IPv6
DOT	Department of Transportation
DSRC	Dedicated Short Range Communication. A low-latency, line-of-sight wireless data transmission standard designed for interactions between vehicles and infrastructure in a dynamic transportation environment.
FHWA	Federal Highway Administration
FTP	File Transfer Protocol
GHz	Gigahertz
GPS	Global Positioning System
HCI	Hardware Configuration Item
HELP	Highway Emergency Local Patrol
HTTP	Hyper-Text Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol version 4

Term	Definition
IPv6	Internet Protocol version 6
LAN	Local Area Network
MAC	Media Access Control
MTU	Maximum Transmission Unit
NMEA	National Marine Electronics Association
NY	New York
NYSDOT	New York State Department of Transportation
OBD-II	On-board Diagnostics II. A standard for a light vehicle diagnostics communication port.
OBE	On-board equipment
OBU	On-board unit. In this context, more specifically the DSRC equipment connected directly to a vehicle data bus.
OS	Operating System
PFS	Pooled Fund Study
PGN	Parameter Group Number. A unique identifier used as a network address in the SAE J1939 data standard to group similar data parameters.
PID	Parameter identifier. A unique code used in a controller area network to request specific equipment operational and state data.
PoE	Power over Ethernet
POSIX	Portable Operating System Interface
PSID	Provider service identifier
PVDM	Probe Vehicle Data Message
RSE	Roadside equipment. DSRC-related equipment deployed near a roadway or intersection.
RSU	Roadside unit. In this context, more specifically the DSRC equipment (radio and processor) at the roadside connected to a backhaul connection.
SAE	Society of Automotive Engineers
SCMS	Security Certificate Management System
SCP	Secure Copy
SNMP	Simple Network Management Protocol

Term	Definition
SP	Synesis Partners
SPN	Suspect Parameter Number. A lower-level identifier within a PGN that describes what a particular data value represents, its update frequency, and its unit of measure.
SSH	Secure Shell
SSL	Secure Sockets Layer
STP	System Test Plan
UDP	User Datagram Protocol
UML	Unified Modeling Language
U.S. DOT	United States Department of Transportation
UTC	Coordinated Universal Time. The primary time standard by which clocks can be regulated. Daylight savings time changes are ignored.
VLAN	Virtual Local Area Network
WAVE	Wireless Access in Vehicular Environments
WBSS	WAVE Basic Service Set

APPENDIX B - ADDITIONAL APPLICATION SOFTWARE DESCRIPTION

This appendix provides additional information about the functions of the OBU application software. Unified Modeling Language (UML) sequence diagrams are provided herein as a way of amplifying the broader description in Section 3 of the main report.

Startup The Startup sequence, as shown in Figure 14, begins with the OS initiating the startup module, which first checks for updates on the OBU, updates application modules as appropriate from local storage, and then starts the main RdWx application module. The RdWx module then reads the configuration file and initiates each of the submodules. Each submodule then reads its own configuration and schedules its continuous data collection.

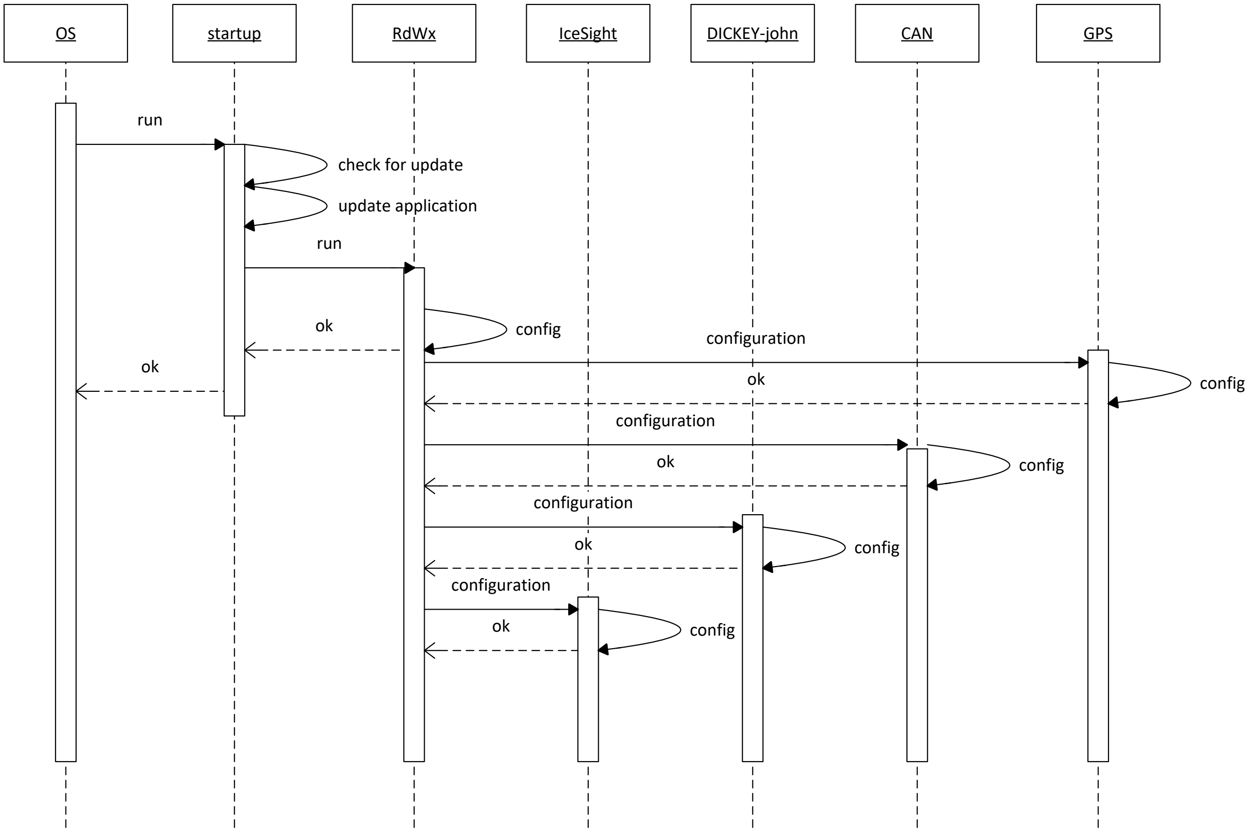


Figure 14 - Startup Sequence Diagram (Source: Synesis Partners LLC)

Collect

The Collect sequence, as shown in Figure 15, begins with the RdWx module determining that the collection interval has passed and asking for the most recent data from each of the device-specific submodules. Each submodule has been continuously monitoring its sensing devices for new data and provides the most recent data to the

RdWx module. The RdWx module then checks the available storage space on the OBU and, if necessary, deletes the oldest data file to make room for the newest data. The RdWx module then opens the current data file and saves the most recent data.

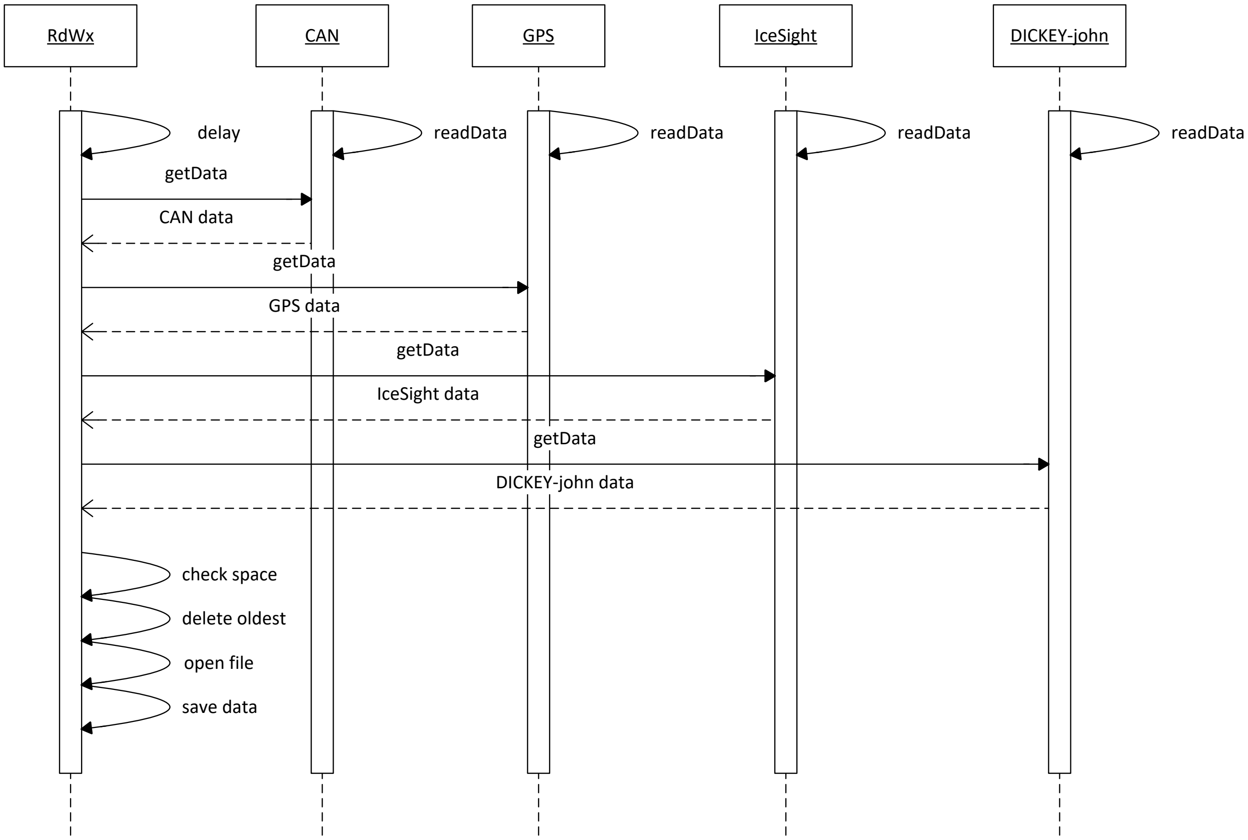


Figure 15 - Collect Sequence Diagram (Source: Synesis Partners LLC)

Upload

The Upload sequence, as shown in Figure 16, begins with the WBSS monitoring the OBU for detection of an available IPv6 network connection over DSRC. When a network connection is made, the Upload module checks the remote server over the network connection for OBU software updates and downloads any appropriate updates. The Upload module then starts uploading data files from local OBU storage, starting with the most recent file and working back through the older files. The Upload module verifies with the remote server that each file has been received. Data files received on the remote server are then deleted from local OBU storage.

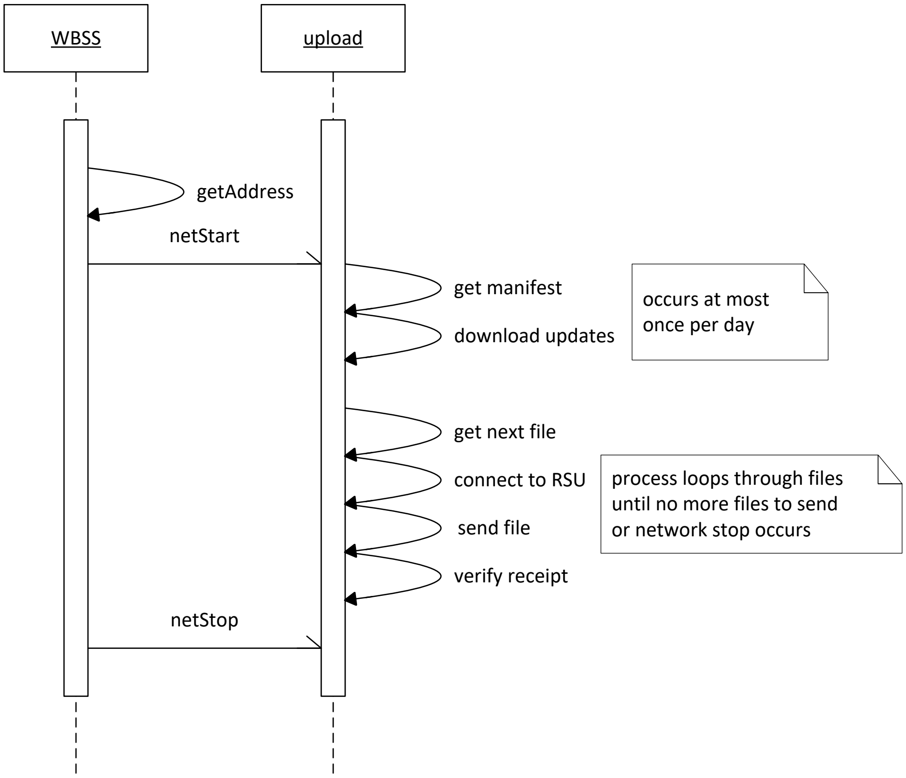


Figure 16 - Upload Sequence Diagram (Source: Synesis Partners LLC)

APPENDIX C - ICESIGHT INTERFACE

The IceSight sensor package procured in this project for deployment on a New York State DOT maintenance vehicle provides road condition data to supplement the data obtained directly from the vehicle CAN bus. The IceSight device is connected to the OBU by an Ethernet cable and provides data once per second (1 Hz). As detailed below, the data include instantaneous and interval-averaged parameter values.

Data Definitions

Each record (line) of the data stream from the IceSight contains thirteen or more values, each separated by at least one space, as demonstrated by the following example:

```
1321 1210 1.091 32.1 35.4 3 3 WET WET 7 7 0.86 0.86 0 FAIR 32.22 33.55 33.55 -102
```

The sequence and values appearing in the data record are defined in Table 5 and Table 6, respectively.

Table 5 - IceSight Data Record Definitions (Source: High Sierra Electronics)

Position	1	2	3	4	5	6	7	8	9
Example	1321	1210	1.091	32.1	35.4	3	3	WET	WET
Format	YYYY	XXXX	R.RRR	AAA.A	BBB.B	C	D	EEE	FFF

Position	10	11	12	13	14	15	16	17	18	19
Example	7	7	0.86	0.86	0	FAIR	32.22	33.55	33.55	-102
Format	G	H	I.II	J.JJ	K	LLLL	MMM.MM	NNN.NN	OOO.OO	PPP.PP

Table 6 - IceSight Data Parameter Definitions (Source: High Sierra Electronics)

Position	Example	Format	Data Type	Description
1	1321	YYYY	Floating point	Reported Y voltage (0-5,000 mV)
2	1210	XXXX	Floating point	Reported X voltage (0-5,000 mV)
3	1.091	R.RRR	Floating point	Ratio of the voltages (y/x)
4	32.1	AAA.A	Floating point	Air temperature (Celsius), secondary, (100.1 signifies an error condition)
5	35.4	BBB.B	Floating point	Surface temperature (Celsius), (100.1 signifies an error condition)
6	3	C	Integer	Displayed condition code number (0-15)[1 2]
7	3	D	Integer	Measured condition code number (0-15)[1 2]
8	WET	EEE	Text	Mnemonic for displayed condition[1 2]
9	WET	FFF	Text	Mnemonic for measured condition[1 2]
10	7	G	Integer	Displayed friction code number (0-31)[1]
11	7	H	Integer	Measured friction code number (0-31)[1]
12	0.86	I.II	Floating point	Displayed friction code value[1]
13	0.86	J.JJ	Floating point	Measured friction code value[1]
14	0	K	Integer	"dirty lens" value3 (0-10)
15	FAIR	LLLL	Text	"grip" value (GOOD, FAIR, POOR)
16	32.22	MMM.MM	Floating point	Relative humidity (%)
17	33.55	NNN.NN	Floating point	Air temperature (Celsius), tertiary, (100.1 signifies an error condition)

18	33,55	OOO.OO	Floating point	Air temperature (Celsius), primary, (100.1 signifies an error condition)
19	-102	PPP.PP	Floating point	Currently not used

[1] Measured condition, measured friction code, measured mnemonic, and measured friction are instantaneous values.
Displayed condition, displayed friction code, displayed mnemonic, and displayed friction are averaged values.
[2] Condition Codes:

0 UNK Unknown
1 DRY Dry
2 DMP Damp
3 WET Wet
4 SNO Snow
5 ICE Ice
6 SLH Slush
10 ERR Error Condition

[3] “Dirty lens” value: Note that values of 2, 6 and 8 would indicate lens cleaning is necessary.

0 Not “Soiled” in any way.
1 Within “Soiled” zone, but not for very long.
2 Within “Soiled” zone for long enough to be of concern.
3 Not Used.
4Received optical signal is low enough that lens could be “Soiled” if condition continues for a long time. Note that a 4
reading is normal in adverse weather conditions.
5 Low enough that lens could be “Soiled”, and also within of “Soiled” zone for a short time.
6 Low enough that lens could be “Soiled”, and also has been inside of “Soiled” polygon long enough to be of concern.
8 Low for long enough that lens should be considered “Soiled”.
9 Low for a long time and also within the “Soiled” zone for a short time.
10 Low for a long time and also within the “Soiled” zone for a long time.