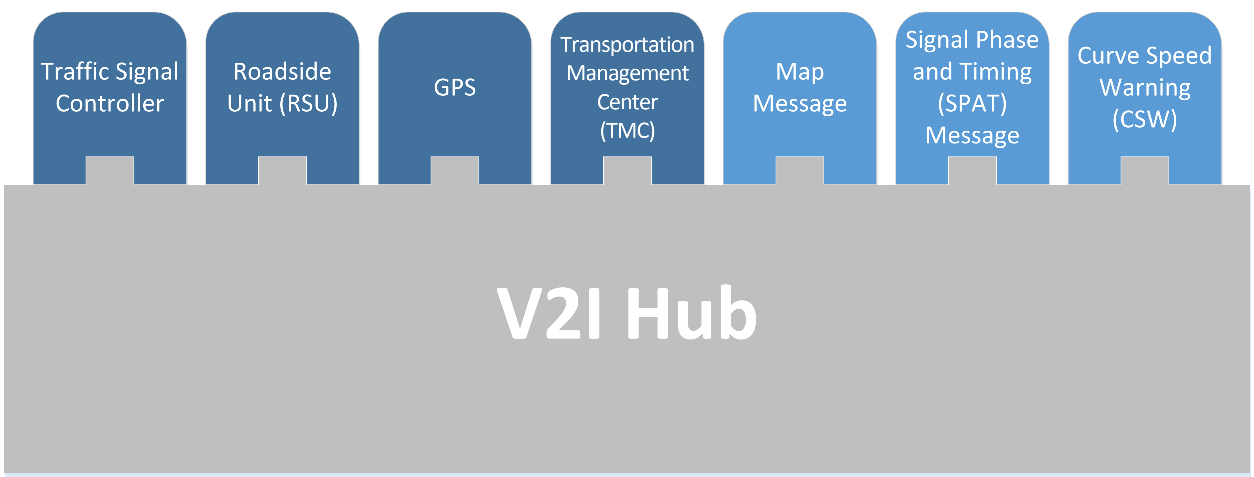
Integrated Vehicle-to-Infrastructure Prototype (IVP)

V2I Hub Interface Control Document (ICD)

www.its.dot.gov/index.htm
Draft Report — May 2016
FHWA-JPO-16-TBD



U.S. Department of Transportation

Produced by Battelle Memorial Institute
U.S. Department of Transportation
Office of the Assistant Secretary for Research and Technology
Federal Highway Administration, Office of Operations Research and Development

The cover image is from Battelle.

Notice

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA-JPO-14-TBD		

4. Title and Subtitle Integrated Vehicle-to-Infrastructure Prototype (IVP) V2I Hub Interface Control Document (ICD)	**5. Report Date** May 2016 **6. Performing Organization Code**
7. Author(s) Battelle	**8. Performing Organization Report No.** 100036384-3.2
9. Performing Organization Name and Address Battelle Memorial Institute 505 King Ave. Columbus, OH 43235	**10. Work Unit No. (TRAIS)** **11. Contract or Grant No.** DTFH61-12-D-00040-T-13008
12. Sponsoring Agency Name and Address Federal Highway Administration Office of Operations Research and Development Turner-Fairbank Highway Research Center 6300 Georgetown Pike McLean, VA 22101	**13. Type of Report and Period Covered** **14. Sponsoring Agency Code**

15. Supplementary Notes

Deborah Curtis, FHWA GTM

16. Abstract

The Integrated Vehicle-to-Infrastructure Prototype (IVP) is part of USDOT's Vehicle-to-Infrastructure (V2I) Program and was developed to support jurisdictions in deploying V2I technology by reducing integration efforts and issues.

- V2I Hub is a software platform that enables connected vehicles to talk to existing traffic management hardware and systems, such as traffic signal controllers, Transportation Management Centers, pedestrian and vehicle sensors, road weather sensors, and dynamic message signs.
- V2I Hub simplifies integration by translating communication between different standards and protocols.
- Using a modular design, software plugins enable efficient connections to new hardware, custom connections to specialized systems, and Connected Vehicle Safety Apps to run on roadside equipment.

This Interface Control Document for the Vehicle-to-Infrastructure (V2I) Hub system documents the external interfaces exhibited by the software and hardware components of the platform.

17. Key Words	18. Distribution Statement
V2I Hub, Connected Vehicles, Vehicle-to-Infrastructure, Integrated Vehicle-to-Infrastructure Prototype, IVP, Signal Phase and Timing, SPaT, V2I Hub Interface Control Document, ICD	Final

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
None	None	54	

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

Table of Contents

List of Tables

List of Figures

Chapter 1 Scope

The Integrated Vehicle-to-Infrastructure Prototype (IVP), called V2I Hub, is part of USDOT's Vehicle-to-Infrastructure (V2I) Program and was developed to support jurisdictions in deploying V2I technology by reducing integration issues and enabling use of their existing transportation management hardware and systems. V2I Hub is a software platform that utilizes plugins to translate messages between different devices and run connected vehicle safety applications on roadside equipment. This document uses V2I Hub and IVP interchangeably.

Scope

This document captures the external interfaces necessary to support the Vehicle-to-Infrastructure (V2) Hub Platform and related messages between the platform and a variety of infrastructure devices, including traffic signal controllers, back office systems, mobile devices, etc., organized around the V2I applications supported by the platform. This document is focused on the required information and format of this information that is being gathered from and provided to the devices connected to the V2I Hub Platform. This document contains references to existing standards, proposed modifications to existing standards, and new interface requirements. Each of the interfaces defined in this document will be categorized into one of these types to assist with future proposed revisions to existing standards. The internal interfaces are not documented in this document. For those wishing to integrate their own plugin with a V2I Hub system, please refer to the *V2I Hub Plugin Creation Manual* (Battelle, July 2016).

Terminology

The meanings of the auxiliary verbs used in this document are defined as follows:

- Shall Compliance with a requirement, specification or a test is mandatory
- Should Compliance with a requirement, specification, or a test is recommended
- May Expresses a permissible way to achieve compliance

Chapter 2 Applicable Documents

The following documents form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those cited in the contract.

Government Documents

System Requirement Description, 5.9GHz DSRC Roadside Equipment Device Specification, version 3.0, 10-March-2012

System Design Document for the INFLO Prototype, FHWA-JPO-14-169

Non-Government Publications

Society of Automotive Engineers (SAE)

J2735:2009-11	Object Dedicated Short Range Communications (DSRC) Message Set Dictionary

National Transportation Communications for ITS Protocol (NTCIP)

1202:2005 V02.19	Object Definitions for Actuated Traffic Signal Controllers (ASC)
1203 v02	NTCIP Object Definitions for Dynamic Message Signs (DMS)
1204 v03	Object Definitions for Environmental Sensor Stations (ESS)
1211:2008 V01.38	Object Definitions for Signal Control and Prioritization (SCP)

Radio Technical Commission for Maritime Services (RTCM)

10402.3	Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service
10403.2	Differential GNSS (Global Navigation Satellite Systems)

National Marine Electronics Association

NMEA 0183	Interface Standard for electrical signal requirements, data transmission protocol and time, and specific sentence formats for a 4800-baud serial data bus.

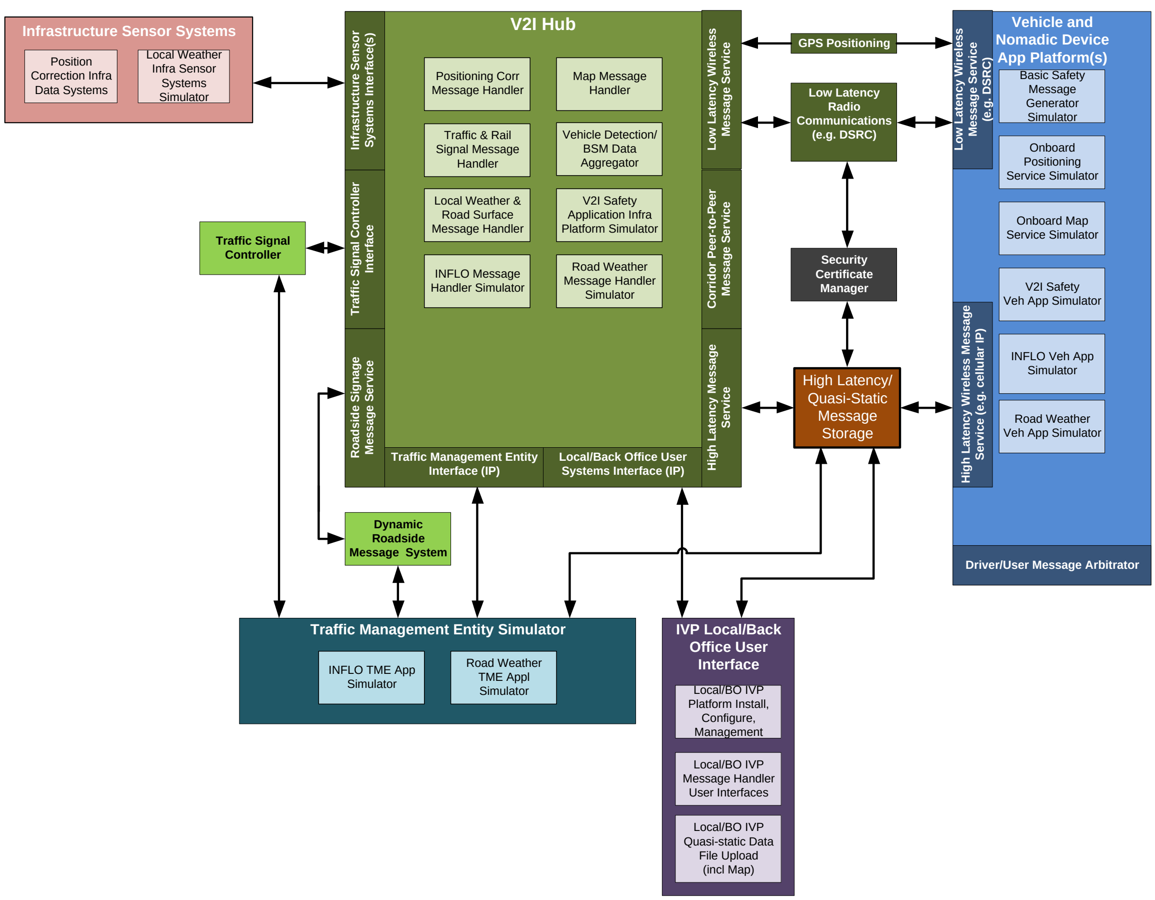
Order of Precedence

In the event of a conflict between the text of this document and the references cited herein, the inconsistencies should be brought to the attention of the project manager. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

Chapter 3 Interface Requirements

The system-of-interest (SOI) for the V2I Hub Platform project centers on defining interfaces to support the bi-directional exchange of information between a variety of 'mobile' devices (e.g., in-vehicle, aftermarket safety devices and nomadic devices) and a roadside system interconnected with a Traffic Signal Controller (TSC) and other information sources necessary to support a variety of V2I applications.

A functional diagram of the SOI is shown in Figure 3-1. The SOI is represented within the box labeled "V2I Hub" with the external interfaces identified by arrowed connectors to other entities. The SOI box also shows the functions and interfaces which will be required in order for the SOI to perform its intended operations.

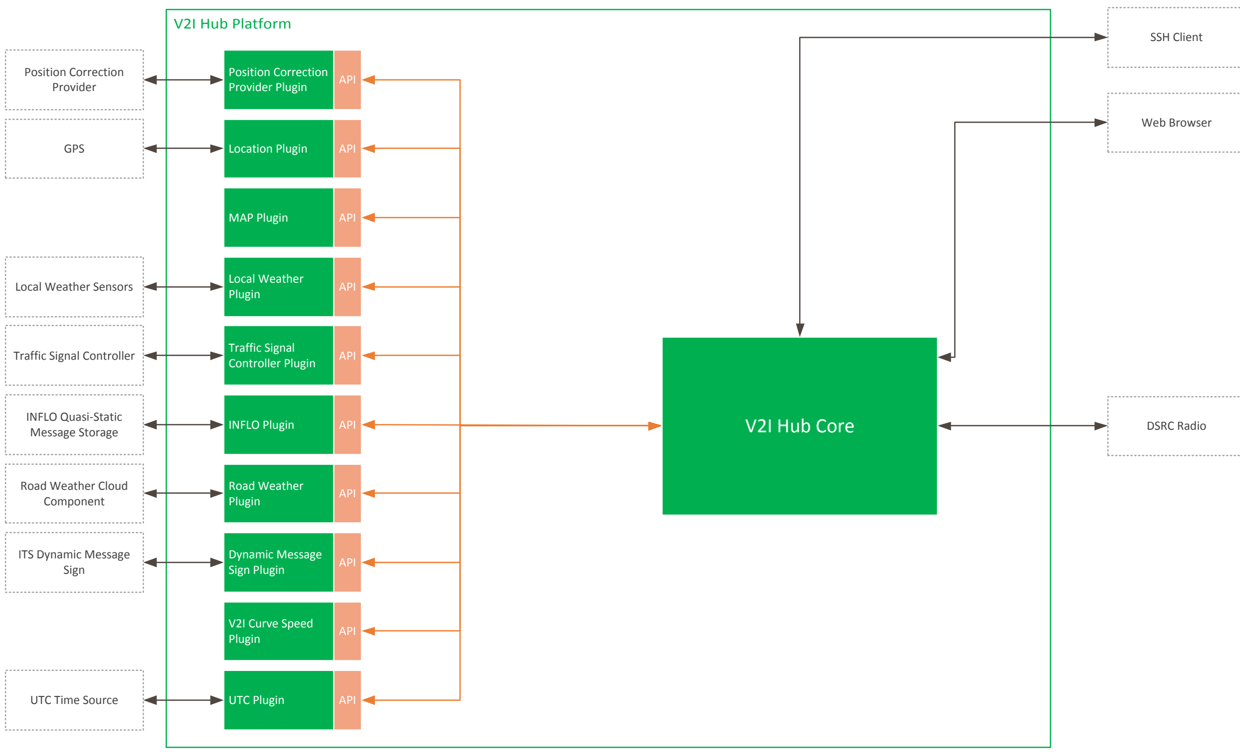


Source: Battelle

Figure 3-1. V2I Hub System Block Diagram

In addition to the bi-directional interfaces between the SOI and mobile devices, interfaces are necessary to support information flows to the SOI from a common time standard, position correction information source and other sources as required by the V2I applications that reside on the platform.

Figure 3-2 represents a view of the physical architecture currently being implemented for the SOI. The small green boxes represent the different software modules (referred to as plugins) that will be used to contain the specific interface code and required functionality for each external interface. The V2I Hub Core (or main computing module and message handler) interfaces to each of the plugins using an Application Programming Interface (API) bundled with each plugin.



Source: Battelle

Figure 3-2. V2I Hub Interface Data Flows

Table 3-1 followed by subsections describing each interface in detail.

Table 3-1. List of Interfaces

Interface Identifier	System Element 1	System Element 2	Direction
UTC	V2I Hub Platform	UTC Time Source	Input
GPS	V2I Hub Platform	Global Positioning System source	Bidirectional
Position Correction	V2I Hub Platform	RTCM Reference Station	Bidirectional
Traffic Signal Controller	V2I Hub Platform	Traffic Signal Controller	Bidirectional
INFLO Quasi-Static Message Storage	V2I Hub Platform	INFLO Quasi-Static Message Storage	Bidirectional
ITS Dynamic Message Sign	V2I Hub Platform	Dynamic Message Sign	Output
Local Weather	V2I Hub Platform	Local Weather Sensors	Input
Road Weather Cloud Service	V2I Hub Platform	Cloud-based Road Weather	Bidirectional
DSRC Radio	V2I Hub Platform	DSRC Radio	Bidirectional
Web Browser	V2I Hub Platform	System User Client Computer	Bidirectional
SSH Client	V2I Hub Platform	System User Client Computer	Bidirectional

Source: Battelle

UTC Time Source Interface

This section describes the communications interface between the V2I Hub Platform and a UTC time source, as implemented in the UTC Plugin. This interface is required only when reliable internet connectivity is available to communicate with a network time provider (NTP), and no GPS receiver is included in the V2I Hub Platform deployment. The UTC time provides a common time reference for both the roadside unit (RSU) and any on-board equipment (OBE) that is necessary to encode DSRC messages so the transmission latencies can be canceled out. This provides a common time signal and may be provided by the signal controller for synchronization with the TSC operations.

The Open System Interconnection (OSI) protocol stack describing the interconnection between the V2I Hub Platform and the UTC time reference is shown in Figure 3-3.

Application	Network Time Protocol (NTP)
Session	Socket
Transport	TCP
Network	IPv4
Data Link	802.3 Ethernet
Physical	10BASE-T

Source: Battelle

Figure 3-3. UTC Protocol Stack

In the event that the V2I Hub Platform is not deployed with reliable internet connectivity, the time source is derived from the GPS receiver interface, as described in the section below.

GPS Receiver Interface

This section describes the communication interface between the V2I Hub Platform and the Global Positioning System (GPS) receiver, as implemented in the Location Plugin. This interface is required only if the deployed DSRC radio does not include an integral GPS receiver.

The interconnection between the V2I Hub Platform and the GPS receiver conforms to the National Marine Electronics Association (NMEA) 0183 Standard for data communications for position data output from the GPS receiver to the V2I Hub Platform, and the Radio Technical Commission for Maritime Services (RTCM) 10402.3 and 10403.2 for differential correction information flowing from the V2I Hub Platform to the GPS receiver. The Open System Interconnection (OSI) protocol stack describing the interconnection is shown in Figure 3-4.

Application	NMEA 0183, RTCM10402.3, RTCM10403.2
Session	PPP
Physical	RS-232

Source: Battelle

Figure 3-4. V2I Hub Platform and GPS receiver Protocol Stack

Position Correction Interface

This section describes the communication interface between the V2I Hub Platform and a position correction provider, specifically, a Radio Technical Commission for Maritime Services (RTCM) GPS reference station, as implemented by the RTCM Plugin. The communication between the V2I Hub Platform and the RTCM reference station shall comply with the RTCM 10402.3 and RTCM 10403.2 standards.

The Open System Interconnection (OSI) protocol stack describing the interconnection between the V2I Hub Platform and the RTCM reference station is shown in Figure 3-5.

Application	RTCM10402.3, RTCM10403.2
Session	Socket
Transport	TCP
Network	IPv4
Data Link	802.3 Ethernet
Physical	10BASE-T

Source: Battelle

Figure 3-5. V2I Hub Platform and RTCM Protocol Stack

Traffic Signal Controller Interface

This section describes the communication interface between the V2I Hub Platform and the Traffic Signal Controller (TSC), as implemented in the SPaT (Signal Phase and Timing) Plugin. Generally, the communication between the V2I Hub Platform and the TSC will conform to the applicable NTCIP standards. However, in order to meet the requirements of the V2I Hub Platform, additional Simple Network Management Protocol (SNMP) objects have been included to the NTCIP 1202 Management Information Base (MIB) as described in the following paragraphs.

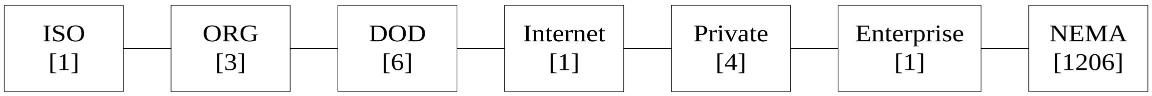
The Open System Interconnection (OSI) protocol stack describing the interconnection between the V2I Hub Platform and the TSC is shown in Figure 3-6.

Application	NTCIP data objects as described in this document
Presentation	ASN.1
Session	Sockets
Transport	UDP
Network	IPv6, port 6053
Data Link	802.3 Ethernet
Physical	10BASE-T

Source: Battelle

Figure 3-6. SPaT System and TSC Protocol Stack

The interface between the V2I Hub Platform and TSC will make use of SNMP data objects. Each data element has on Object Identifier (OID) that corresponds to its physical location within the global naming tree. The root OID for the National Electrical Manufacturers Association (NEMA) data objects is shown in Figure 3-7. The OID for each data element in the TSC interface shall be prefixed by the NEMA object identifier.



Source: Battelle

Figure 3-7. Root NEMA Object Identifier

NTCIP 1202 Data Ob0jects

The interface between the V2I Hub Platform and TSC will make use of the NTCIP 1202 and extended data objects that are listed in Table 3-2. The extended data objects listed in Table 3-3 provide the TSC phase time-to-change information that is currently not supported in the NTCIP standards.

Table 3-2. NTCIP 1202 Interface SNMP Data Objects

Data Object	OID	NTCIP 1202 Paragraph	Get Individual OID	Get Group of OIDs
maxPhases	NEMA.4.2.1.1.1.0	2.2.1	Yes	
phaseTable	NEMA.4.2.1.1.2	2.2.2		
phaseEntry	NEMA.4.2.1.1.2.1	2.2.2		
phasePedestrianClear	NEMA.4.2.1.1.2.1.3.x	2.2.2.3		Yes
phaseYellowChange	NEMA.4.2.1.1.2.1.8.x	2.2.2.8		Yes
phaseRedClear	NEMA.4.2.1.1.2.1.9.x	2.2.2.9		Yes
phaseRedRevert	NEMA.4.2.1.1.2.1.10.x	2.2.2.10		Yes
maxPhaseGroups	NEMA.4.2.1.1.3.0	2.2.3	Yes	
phaseStatusGroupTable	NEMA.4.2.1.1.4	2.2.4		
phaseStatusGroupEntry	NEMA.4.2.1.1.4.1	2.2.4		
phaseStatusGroupReds	NEMA.4.2.1.1.4.1.2.x	2.2.4.2	Yes	Yes
phaseStatusGroupYellows	NEMA.4.2.1.1.4.1.3.x	2.2.4.3	Yes	Yes
phaseStatusGroupGreens	NEMA.4.2.1.1.4.1.4.x	2.2.4.4	Yes	Yes
phaseStatusGroupDontWalks	NEMA.4.2.1.1.4.1.5.x	2.2.4.5	Yes	Yes
phaseStatusGroupPedClears	NEMA.4.2.1.1.4.1.6.x	2.2.4.6	Yes	Yes
phaseStatusGroupWalks	NEMA.4.2.1.1.4.1.7.x	2.2.4.7	Yes	Yes
preemptControlTable	NEMA.4.2.1.6.3	2.7.3		
preemptControlEntry	NEMA.4.2.1.6.3.1	2.7.3		

Data Object	OID	NTCIP 1202 Paragraph	Get Individual OID	Get Group of OIDs
preemptControlState	NEMA.4.2.1.6.3.1.2.x	2.7.3.2	Yes	
maxOverlaps	NEMA.4.2.1.9.1.0	2.10.1	Yes	
overlapTable	NEMA.4.2.1.9.2	2.10.2		
overlapEntry	NEMA.4.2.1.9.2.1	2.10.2		
overlapStatusGroupReds	NEMA.4.2.1.9.4.1.2.x	2.10.4.2	Yes	Yes
overlapStatusGroupYellows	NEMA.4.2.1.9.4.1.3.x	2.10.4.3	Yes	Yes
overlapStatusGroupGreens	NEMA.4.2.1.9.4.1.4.x	2.10.4.4	Yes	Yes

Source: Battelle

The NTCIP 1202 standard does not provide data objects to convey the time remaining for each phase of the TSC. The V2I Hub requires the phase timing information. To address this issue, a set of extended SNMP data objects were added to the NTCIP 1202 MIB to provide the phase timing information that is available in the TSC. The extended set of SNMP data objects are listed in Table 3-3.

Table 3-3. Extended SNMP Data Objects

Data Object	OID
spatPhaseTimeToChangeTable	NEMA.3.47.1
spatPhaseTimeToChangeEntry	NEMA.3.47.1.1
spatTimeToChangePhaseNumber	NEMA.3.47.1.1.1
spatVehMinTimeToChange	NEMA.3.47.1.1.2
spatVehMaxTimeToChange	NEMA.3.47.1.1.3
spatPedMinTimeToChange	NEMA.3.47.1.1.4
spatPedMaxTimeToChange	NEMA.3.47.1.1.5
spatOvlpTimeToChangeTable	NEMA.3.47.2
spatOvlpTimeToChangeEntry	NEMA.3.47.2.1
spatTimeToChangeOvlpNumber	NEMA.3.47.2.1.1
spatOvlpMinTimeToChange	NEMA.3.47.2.1.2
spatOvlpMaxTimeToChange	NEMA.3.47.2.1.3
spatCompleteMsg	NEMA.3.47.3
spatColorOutputStatus	NEMA.3.47.4
spatFlashingOutputStatus	NEMA.3.47.5
spatIntersectionStatus	NEMA.3.47.6
spatDiscontinuousChangeFlag	NEMA.3.47.7
spatMessageSeqCounter	NEMA.3.47.8
spatTimestamp	N/A

Source: Battelle

The TSC shall generate a message containing the signal phase and timing (SPaT) information comprised of the SNMP data objects and broadcast it to the V2I Hub Platform. The byte-map structure of the broadcast message is shown in Table 3-4.

Table 3-4. Traffic Signal Controller Broadcast Message

Bytes	Description	
0	0xcd	
1	number of *phase/overlap blocks* below (16)	
2-14	0x01 (phase number) spatVehMinTimeToChange.1 spatVehMaxTimeToChange.1 spatPedMinTimeToChange.1 spatPedMaxTimeToChange.1 spatOvlpMinTimeToChange.1 spatOvlpMaxTimeToChange.1	(1 byte) (2 bytes) (2 bytes) (2 bytes) (2 bytes) (2 bytes) (2 bytes)
15-196	< repeat above for each phase and overlap >	
197-209	0x10 (phase#) spatVehMinTimeToChange.16 spatVehMaxTimeToChange.16 spatPedMinTimeToChange.16 spatPedMaxTimeToChange.16 spatOvlpMinTimeToChange.16 spatOvlpMaxTimeToChange.16	(1 byte) (2 bytes) (2 bytes) (2 bytes) (2 bytes) (2 bytes) (2 bytes)
210-215	phaseStatusGroupReds phaseStatusGroupYellows phaseStatusGroupGreens	(2 bytes bit-mapped for phases 1-16) (2 bytes bit-mapped for phases 1-16) (2 bytes bit-mapped for phases 1-16)
216-221	phaseStatusGroupDontWalks phaseStatusGroupPedClears phaseStatusGroupWalks	(2 bytes bit-mapped for phases 1-16) (2 bytes bit-mapped for phases 1-16) (2 bytes bit-mapped for phases 1-16)
222-227	overlapStatusGroupReds overlapStatusGroupYellows overlapStatusGroupGreens	(2 bytes bit-mapped for overlaps 1-16) (2 bytes bit-mapped for overlaps 1-16) (2 bytes bit-mapped for overlaps 1-16)
228-229	flashingOutputPhaseStatus	(2 bytes bit-mapped for phases 1-16)
230-231	flashingOutputOverlapStatus	(2 bytes bit-mapped for overlaps 1-16)
232	spatIntersectionStatus	(1 byte, see Table 3-5)
233	spatDiscontinuousChangeFlag	(1 byte, upper 5 bits are message version)
234	spatMessageSeqCounter	(1 byte, lower byte of controller up-time)
235-239	spatTimestamp	(5 bytes, hours-minute-second-millisecond)

Bytes	Description	
240-241	spatPedestrianCall	(2 bytes, bit-mapped for phases 1-16)
242-243	spatPedestrianDetect	(2 bytes, bit-mapped for phases 1-16)

Source: Battelle

The *FlashingOutputPhaseStatus* (bytes 228-229) is comprised of two bit-mapped words indicating which phases and overlaps are currently flashing. These, used in conjunction with the Green, Yellow and Red status bytes in NTCIP 2102 interface data objects, will enable determination of a flashing color on a movement. These words will only be valid during normal or programmed flash operation.

Table 3-5. Intersection Status Byte Description

Bit	Feature	Description of Bit (1 = SET)
0	Manual Control Enable Active	Set if Manual Control Enable operation is active.
1	Stop Time (all rings) Active	Set only if the controller has been commanded to stop timing on ALL RINGS.
2	Fault Flash Active	Set if, for any reason, the controller has dropped CVM due to a Failure condition. Failure conditions include MMU faults such as conflict or short yellow; preempt faults such as Interlock or Gate Down failures; communications faults such as SDLC (TS-2 type 1) problems.
3	Preempt Active	Set if any of the preempt runs are active; it will not be set if there is a call for a preempt run but has not been activated yet.
4	TSP Active	Set if any of the TSP runs are active; it will not be set if there is a call for a TSP run but has not been activated yet.
5	Coordination Active (IN STEP)	Set if the controller is currently running an IN-STEP coordination pattern.
6	Coordination-in-Transition (DWELL, ADD, SUBTRACT)	Set when the controller is trying to get a coordination pattern IN-STEP. The controller may be using one of three methods; DWELL, ADD, SUBTRACT.
7	Programmed Flash Active	Set if the controller is in flash other than fault flashes. Example of programmed flash include scheduled, preempt, remote, or auto flash.

Source: Battelle

NTCIP 1211 Data Objects

The interface between the V2I Hub and TSC will make use of the NTCIP 1211 data objects that are listed in Table 3-6**.** These data objects are used to activate priority service calls in the TSC and to get the status of any priority service calls.

Table 3-6. NTCIP 1211 Interface SNMP Data Objects

Data Object	OID	NTCIP 1211 Paragraph	Get Individual OID	Get Group of OIDs
prgPriorityRequest	NEMA.4.2.11.2.1.0	3.2.1	Yes	
prgPriorityUpdate	NEMA.4.2.11.2.2.0	3.2.2	Yes	
prgPriorityStatusControl	NEMA.4.2.11.2.3.0	3.2.3	Yes	
prgPriorityStatusBuffer	NEMA.4.2.11.2.4.0	3.2.4	Yes	
prgPriorityCancel	NEMA.4.2.11.2.5.0	3.2.5	Yes	
prgPriorityClear	NEMA.4.2.11.2.6.0	3.2.6	Yes	

Source: Battelle

INFLO Quasi-Static Message Storage Interface

This section describes the communication interface between the V2I Hub Platform and the INFLO Quasi-Static Message Storage, as implemented by the INFLO plugin. The V2I Hub Platform shall comply with the interface requirements as defined by the System Design Document for the INFLO Prototype.

The Open System Interconnection (OSI) protocol stack describing the interconnection between the V2I Hub Platform and the INFLO Quasi-Static Message Storage is shown in Figure 3-8.

Application	INFLO data objects as described in this document
Presentation	ASN.1
Session	Sockets
Transport	UDP
Network	IPv4, port TBD
Data Link	802.3 Ethernet
Physical	10BASE-T

Source: Battelle

Figure 3-8. SPaT System and INFLO Quasi-Static Message Storage Protocol Stack

Dynamic Message Sign Interface

This section describes the communication interface between the V2I Hub Platform and a dynamic message sign (DMS), as implemented by the Dynamic Message Sign plugin. The communication between the V2I Hub Platform and the DMS will conform to the applicable NTCIP 1203 v02 standards.

The Open System Interconnection (OSI) protocol stack describing the interconnection between the V2I Hub Platform and the Dynamic Message Sign is shown in Figure 3-9.

Application	NTCIP 1203 v02
Session	Sockets
Transport	UDP
Network	IPv6, port TBD
Data Link	802.3 Ethernet
Physical	10BASE-T

Source: Battelle

Figure 3-9. V2I Hub Platform and Dynamic Message Sign Protocol Stack

Local Weather Sensor Interface

This section describes the communication interface between the V2I Hub Platform and local weather sensors, as implemented by the Local Weather plugin. The interface for the Local Weather sensor is NTCIP 1204v03.

The Open System Interconnection (OSI) protocol stack describing the interconnection between the V2I Hub Platform and the Local Weather Sensors is shown in Figure 3-10.

Application	NTCIP 1204 v03
Session	PPP
Physical	RS-232

Source: Battelle

Figure 3-10. V2I Hub Platform and Local Weather Sensors Protocol Stack

Road Weather Cloud Service Interface

This section describes the communication interface between the V2I Hub Platform and cloud-based, road weather data, as implemented by the Road Weather plugin. The interface for the Road Weather Cloud Service is TBD.

The Open System Interconnection (OSI) protocol stack describing the interconnection between the V2I Hub Platform and the Road Weather Cloud Service is shown in Figure 3-11.

Application	TBD
Session	Socket
Transport	TCP
Network	IPv4
Data Link	802.3 Ethernet
Physical	10BASE-T

Source: Battelle

Figure 3-11. V2I Hub Platform and Road Weather Cloud Service Protocol Stack

DSRC Radio Interface

This section describes the communication interface between the V2I Hub Platform's Dedicated Short-Range Communications (DSRC) radio and an external DSRC radio, as implemented in the DSRC Message Manager Software module. Generally, the communication between the V2I Hub Platform's DSRC radio and an external DSRC radio conforms to the immediate forwarded message from the RSU 3.1 spec. The message body sent conform to the SAE J2735:2015 specification, and can also send the SPaT and GID "blob" messages were modified to support the SPaT project as described in the following paragraphs. The immediate forward message contains flags for the RSU to encrypt and/or sign the transmitted DSRC message.

The Open System Interconnection (OSI) protocol stack describing the interconnection between the V2I Hub Platform and the DSRC radio is shown in Figure 3-12.

Application	SAE J2735 and Battelle SPaT data objects
Session	N/A
Transport	IEEE 1609.1
Network	IEEE 1609.3
Data Link	WAVE MAC, 1609.4
Physical	IEEE 802.11p

Source: Battelle

Figure 3-12. V2I Hub Platform and DSRC Protocol Stack

Signal Phase and Timing Message

This section describes the message generated from the V2I Hub Platform's DSRC radio to a mobile device for the Signal Phase and Timing message. The SPaT message provides information regarding the parameters of the controlled intersection and status and timing of each lane-movement at the intersection. The lane numbers shall correspond to the specific lanes described in the GID message for the same intersection.

The ASN.1 Notation:

```
SPATMessage :: = SEQUENCE {
   msgID     DSRCmsgID              -- one byte (0x8d SPAT)
   version   ContentVersion         -- one byte
   size      INTEGER (1..1024)      -- payload size, two bytes
   blob      SPATblob               -- SPAT payload
   crc       MsgCRC OPTIONAL        -- two bytes (J2735 paragraph 7.77)
   }

ContentVersion ::= Integer (0..255)

SPATblob ::= OCTET STRING (SIZE(1..1024))
```

Source: Battelle

The Content Version is used to indicate a change in the message content. It is not an absolute version; rather it is a rolling version number that only has meaning relative to the last message that was broadcast. Applications that notice a change to the content version shall parse the message payload; otherwise the message can be ignored.

The SPaT blob contains the content of the SPaT message and provides the signal phase and timing information for one or more lane-movements at a single intersection. The blob is a binary representation of the SPAT message formatted in accordance with this document.

SPaT Message Format

The SPaT message format is a compilation of data objects in binary form that describe the signal phase and timing at the intersection. Each data object is identified by its object identifier as described in

Table 3-7.

Some data objects will always be present while others will not be included when they are not applicable. The SPaT message always terminates with the End-of-Message object identifier.

Table 3-7. SPaT Message Object Identifiers

Object Identifier	Object Type
0x01	Intersection ID
0x02	Intersection Status
0x03	Message Timestamp
0x04	Movement
0x05	Lane Set
0x06	Current State
0x07	Minimum Time Remaining
0x08	Maximum Time Remaining
0x09	Yellow State
0x0A	Yellow Time
0x0B	Pedestrian Detected
0x0C	Vehicle or Pedestrian Count
0xFF	End of Blob

Source: Battelle

Each object in the message is comprised of three fields consisting of an object identifier, size of the objects payload, and the payload content of the object as shown in Table 3-8.

Table 3-8. SPaT Message Object Format

Field	Note
Object Identifier	Unsigned 8-bit Integer
Size	Unsigned 8-bit Integer
Payload	Varies in length

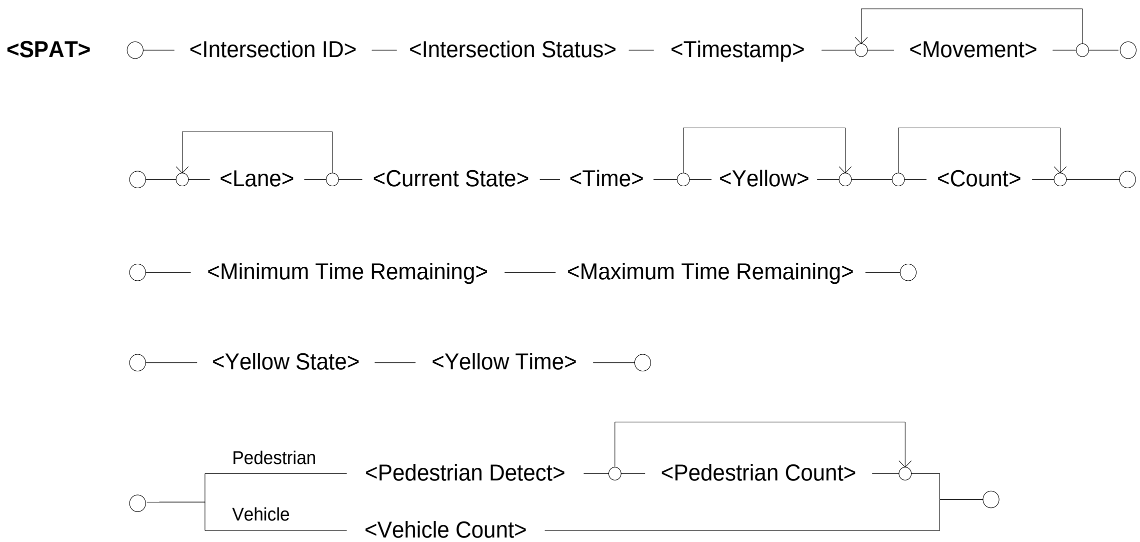
Source: Battelle

The object identifier field specifies the type of object that follows from the object list in the previous table while the object size field specifies the number of bytes contained in the objects payload field. The Movement and End-of-Blob data objects are flags and do not contain the size and payload portion of the object format.

All binary numeric data is stored in big-endian format.

SPaT Message Diagram

The structure of data objects that comprise the SPaT message are illustrated in Figure 3-13. A description of each data object is provided in the following paragraphs.



Source: Battelle

Figure 3-13. SPaT Message Diagram

Intersection ID

The Intersection ID object is used to globally and uniquely define an intersection within a country or region in a 32-bit field. Assignment rules for this value are established elsewhere and may use regional assignment schemas that vary.

Object Format:

Field	Note
Object Identifier	0x01
Size	0x04
Intersection ID	Unsigned 32-bit Integer

Source: Battelle

Example:

```
01 04 00 00 00 01  -- Intersection ID #0001
```

Source: Battelle

Intersection Status

The Intersection Status object contains Advance Signal Controller (ASC) status information that may be sent to mobile devices as part of the SPAT process.

Object Format:

Field	Note
Object Identifier	0x02
Size	0x01
Intersection Status	Unsigned 8-bit Integer

Source: Battelle

The bits of the Intersection Status object are as follows. A bit set to a logic one indicates that the described condition is present at the intersection while a bit set to zero indicates that it is not present. All bits are set to zero in normal operation.

0 (LSB)	Manual Control is enabled.
1	Stop Time is activated and all counting/timing has stopped.
2	Intersection is in the conflict flash state.
3	Preempt is active
4	Priority is active
5	Reserved
6	Reserved
7	Reserved

Example:

```
02 01 00  -- Intersection status normal
```

Source: Battelle

Message Timestamp

The Message Timestamp provides the UTC time signature that indicates when the message was originally generated. This time indication allows detection and correction of different rates and offsets between the clocks and latency of message transmission. It is presumed that the receiver shares a common sense of time with the sender using a common UTC time reference.

Object Format:

Field	Note
Object Identifier	0x03
Size	0x05
Timestamp, seconds	Unsigned 32-bit Integer
Timestamp, tenths	Unsigned 8-bit Integer

Source: Battelle

The time format is represented in two numbers, the first as seconds from epoch, midnight of January 1, 1970 and the second is in one-tenth of a second increments.

Example:

```
03 05 4E BA F6 B5 02 - Wed Nov 9 16:55:01.2 EST 2011; 1320875701.2
```

Source: Battelle

Movement

The Movement data object indicates the beginning of a movement description. Each movement describes its lane set, current state, and remaining service time. Typically, inactive (red phase) movements are not included in the SPAT message; however, they can be if an application requires them. This object is a flag and does not contain any data.

Object Format:

Field	Note
Object Identifier	0x04

Source: Battelle

Each movement must provide the data objects that describe its current signal phase and timing information before the next Movement flag is encountered in the SPAT message. The objects may appear only once and can appear in any order. A movement description may consist of the following data objects.

1. Lane Set
2. Current State; Vehicle, Pedestrian, or Special Lane
3. Minimum and Maximum Time Remaining
4. Yellow State and Time Duration
5. Pedestrian Detected (optional)
6. Vehicle or Pedestrian Count (optional)

Example:

```
04 -- Beginning of a movement description
```

Source: Battelle

Lane Set

The Lane Set object is a sequence of one or more double-octets where each octet pair represents one of the lanes in an intersection and a set of one or more movements. The lane numbers must correspond with the lane number included in the GID message for the intersection.

Object Format:

Field	Note
Object Identifier	0x05
Size	Number of bytes in the payload
Lane Set	Varies in length

Source: Battelle

Each lane is described with a double-octet where the first octet is a bit-mapped byte that provides information about which movements are being described and the second octet is the lane number for which the movements apply.

All lane movements at the intersection that share a common phase and timing shall be included in the lane set. This aggregation reduces the overall byte size of the SPaT message and will diminish the bandwidth loading of the wireless link to the mobile devices.

The Movements octet is obtained by combining (logical OR) the various bit-masks together to produce a single value. A bit set to a logic one indicates that the described movement is present while a bit set to zero indicates that it is not present.

```
0 (LSB)  Straight.
1        Left Turn.
2        Right Turn.
3        U Turn.
0-7      Unused
```

Examples:

```
05 02 07 03         -- Movement: Straight/left/right on lane 3
05 04 02 05 02 09   -- Movement: Left turn on lanes 7 and 9
05 04 08 04 03 02   -- Movement: U-Turn on lane 4; straight/left on lane 2
```

Source: Battelle

Current State

The Current State object defines the current state of a particular known movement. The content of this object is determined by the type of lane(s) that it applies too; vehicle, pedestrian, or special lane.

Object Format:

Field	Note
Object Identifier	0x06
Size	Number of bytes in the payload
Current State	Varies in length

Source: Battelle

Vehicle Lanes

For vehicle lanes, this object defines all possible lights pertaining to a particular known movement. This data object is a bit-mapped integer value whose bits represent the signal state in accordance with Table 3-9. Only the lower bits of significance are sent, allowing shorter payload byte counts when conditions represented by the higher order bit are not applicable. The signal light state value is assembled by combining (logical OR) the various bit-masks together.

Examples:

```
06 02 01 04          -- Solid red ball with green right arrow
06 01 01             -- Solid green ball
06 04 04 00 00 00    -- Red U-Turn Arrow
```

Source: Battelle

Table 3-9. Current Phase State Bit Mapping

	Green	Yellow	Red	Flashing
Ball	0x00000001	0x00000002	0x00000004	0x00000008
Left Arrow	0x00000010	0x00000020	0x00000040	0x00000080
Right Arrow	0x00000100	0x00000200	0x00000400	0x00000800
Straight Arrow	0x00001000	0x00002000	0x00004000	0x00008000
Soft Left Arrow	0x00010000	0x00020000	0x00040000	0x00080000
Soft Right Arrow	0x00100000	0x00200000	0x00400000	0x00800000
U-Turn Arrow	0x01000000	0x02000000	0x04000000	0x08000000
Dark = 0x00000000				

Source: Battelle

Pedestrian Lanes

For pedestrian lanes, this object defines the current signal state of the crosswalk indicators and conditions for a particular pedestrian movement. The lower nibble contains the state of the crosswalk indicators and the upper nibble contains a bit-mapped value describing conditions that exist at the pedestrian lane.

MSN	LSN
Conditions	Indicator State

Source: Battelle

The upper nibble is obtained by combining (logical OR) the various bit-masks together to produce a single value. A bit set to a logic one indicates that the described condition is present while a bit set to zero indicates that it is not present.

```
0-3       State of Crosswalk Indicators.
4         Pedestrian Call Active.
5         Unused.
6         Unused.
7 (MSB)   Unused.
```

The enumerated values for the lower nibble of this data object are as follows.

```
0    Unavailable or not equipped
1    Do not walk
2    Flashing, do not walk
3    Walk
```

Examples:

```
06 01 00  -- No pedestrian control
06 01 03  -- Walk is active, no pedestrian call.
06 01 11  -- Do not walk is active, pedestrian service has been called.
```

Source: Battelle

Special Lanes

For special lanes, this object defines the current signal state of special indicators for a particular known special movement, such as a train. This data object is only used for special lanes.

The enumerated values of this data object are as follows.

```
1    Empty or not in use
2    Special lane is about to be occupied
3    Special lane is occupied
4    Special lane is about to be empty
```

Example:

```
06 01 03  -- The special lane is currently occupied (i.e. a train)
```

Source: Battelle

Minimum Time Remaining

The minimum time remaining data object is used to relate the minimum guaranteed amount of time remaining before the signal phase will change to the next phase with a precision of one-tenth of a second. The remaining time is referenced to the local UTC time signature contained in the message timestamp.

Object Format:

Field	Note
Object Identifier	0x07
Size	0x02
Time Remaining	Unsigned 16-bit Integer

Source: Battelle

A range of two full minutes is supported in tenths of a second (0 to 1200). A value of zero is taken to mean no time remaining while a value greater than the allowed range shall have the following interpretations.

```
1201   Indefinite time remains (greater than two minutes)
1202   Unknown amount of time remains
```

Examples:

```
07 02 01 60    -- At least 35.2 seconds remain in the current phase
07 02 04 B1    -- More than 2 minutes remain in the current phase
```

Source: Battelle

Maximum Time Remaining

The maximum time remaining data object is used to relate the anticipated maximum time remaining before the signal phase is predicted to change to the next phase with a precision of one-tenth of a second. This time is not guaranteed and may change. The remaining time is referenced to the local UTC time signature contained in the message timestamp.

Object Format:

Field	Note
Object Identifier	0x08
Size	0x02
Time Remaining	Unsigned 16-bit Integer

Source: Battelle

The value range is identical to the Minimum Time Remaining data object. This value may be the same as, but will never be less than the minimum time value. The minimum and maximum values will be identical when the time remaining is known to be exact.

Example:

```
08 02 01 60    -- Up to 35.2 seconds may remain in the current phase
```

Source: Battelle

Yellow State

The Yellow State data object is identical to the Current State data object for motorized vehicle lanes and pedestrian lanes. Use of this data object is optional.

Object Format:

Field	Note
Object Identifier	0x09
Size	Number of bytes in the payload
Current State	Varies in length

Source: Battelle

Vehicle Lanes

For motorized vehicle lanes, this object indicates next (yellow) signal state of all possible lights pertaining to a particular known movement after the current signal state expires. This data object is only used when the current signal state is in the green phase, not yellow or red.

See the Current State data object definition for format and content information.

Examples:

```
09 02 02 04          -- Solid red ball with yellow right arrow is next
09 01 02             -- Solid yellow ball is next
09 04 02 00 00 00    -- Yellow U-Turn arrow is next
```

Source: Battelle

Pedestrian Lanes

For pedestrian lanes, this data object indicates next active state of the crosswalk indicators for a particular known pedestrian movement.

See the Current State data object definition for format and content information.

Example:

```
09 01 02  -- Flashing, do not walk is next
```

Source: Battelle

Yellow Time

The Yellow Time data object is used to relate the duration of yellow signal phase with a precision of one-tenth of a second. The yellow phase time parameter is fixed in the phase timing configuration so the yellow time is considered exact. Use of this data object is required when the Yellow State or Yellow Pedestrian State data objects is used.

Object Format:

Field	Note
Object Identifier	0x0A
Size	0x02
Time Remaining	Unsigned 16-bit Integer

Source: Battelle

See the Time Remaining data object definition for format and content information.

Example:

```
0A 02 00 28  -- Yellow duration is 4.0 seconds
```

Source: Battelle

Pedestrian Detect

The pedestrian detected data object indicates the possible presence of one or more pedestrians or other objects in the movements walk area. This data object is optional.

Object Format:

Field	Note
Object Identifier	0x0B
Size	0x01
Detect Flags	Unsigned 8-bit Integer

Source: Battelle

This data object can contain the pedestrian detection information for pedestrian lanes equipped to detect pedestrian movements. The detection state of the pedestrian detector at the intersection is placed in the lower nibble of the byte as shown below.

MSN	LSN
Reserved	Primary Detector

Source: Battelle

The enumerated values of this data object are as follows.

```
0   Unavailable
1   No pedestrians detected
2   One or more possible pedestrians detected
```

Examples:

```
0B 01 00  -- Pedestrian detection unavailable
0B 01 01  -- Pedestrian detection available, none detected
0B 01 02  -- One or more pedestrians detected
```

Source: Battelle

Vehicle or Pedestrian Count

The vehicle or pedestrian count data object provides an estimated count of vehicles or pedestrians within a predefined time period. For motorized vehicle lanes this data object provides the estimated vehicle count while for pedestrian lanes it provides the estimated pedestrian count. This data object is optional.

Object Format:

Field	Note
Object Identifier	0x0C
Size	0x01
Count	Unsigned 8-bit Integer

Source: Battelle

Example:

```
0C 01 0B  -- An estimate count of 11 vehicles or pedestrians
```

Source: Battelle

Geographical Intersection Description Message

This section describes the message generated by the V2I Hub Platform's DSRC radio to mobile devices for the Geographical Intersection Description (GID) message. The GID message provides the physical attributes associated with the intersection, as well as the permitted lane and traffic movements associated with the intersection at the time of day. The extent of the map depends on factors such as topology, signal reception, and other intersections in the area.

The ASN.1 Notation:

```
GIDMessage :: = SEQUENCE {
   msgID     DSRCmsgID              -- one byte (0x87 GID)
   version   ContentVersion         -- one byte
   size      INTEGER (1..1024)      -- payload size, two bytes
   blob      GIDblob                -- GID payload
   crc       MsgCRC OPTIONAL        -- two bytes (J2735 paragraph 7.77)
   }

ContentVersion ::= Integer (0..255)

GIDblob ::= OCTET STRING (SIZE(1..1024))
```

Source: Battelle

The Content Version is used to indicate a change in the message content. It is not an absolute version; rather it is a rolling version number that only has meaning relative to the last message that was broadcast. Applications that notice a change to the content version shall parse the message payload; otherwise the message can be ignored.

The GID blob contains the content of the GID message for a single intersection. The GID blob is a binary representation of the GID message formatted in accordance with this document.

GID Message Format

The GID message format is a compilation of data objects in binary form that provides the geometric description of the intersection. Each data object is identified by its object identifier as described in Table 3-10. Some data objects will always be present while others will not be included when they are not applicable. The GID message always terminates with the End-of-Message object identifier.

Table 3-10. GID Message Object Identifiers

Object Identifier	Object Type
0x01	Message Attributes
0x02	Intersection ID
0x03	Reference Point
0x04	Approach/Egress/Barrier
0x05	Lane
0x06	Lane Attributes
0x07	Barrier Attributes
0x08	Width
0x09	Node List
0x0A	Connection
0x0B	Reference Lane
0xFF	End of Message

Source: Battelle

Each object in the message is comprised of three fields consisting of an object identifier, size of the objects payload, and the payload content of the object as shown in Table 3-11.

Table 3-11. GID Message Object Format

Field	Note
Object Identifier	Unsigned 8-bit Integer
Size	Unsigned 8-bit Integer
Payload	Varies in length

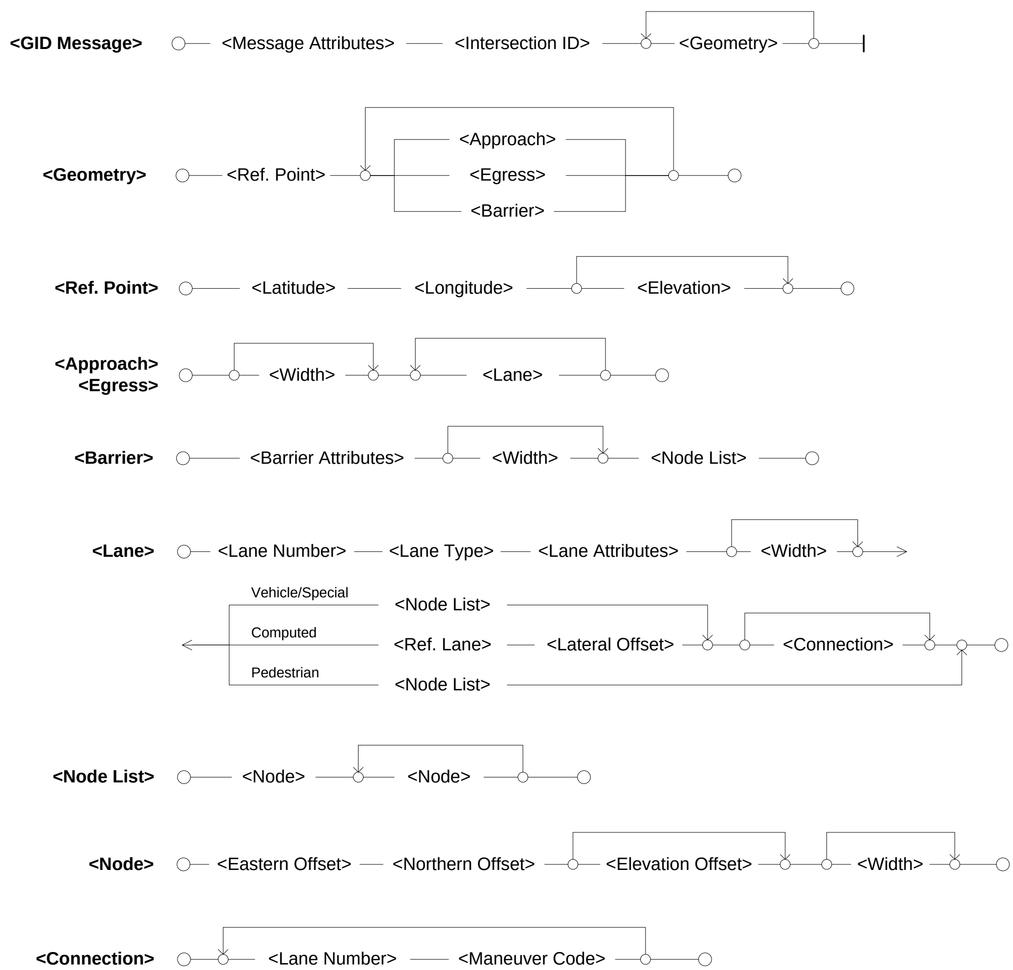
Source: Battelle

The object identifier field specifies the type of object that follows from the object list in the Table 3-10 while the object size field specifies the number of bytes contained in the objects payload field. The Approach and End-of-Message data objects are flags and do not contain the size and payload portion of the object format.

All binary numeric data is stored in big-endian format.

GID Message Diagram

The structure of data objects that comprise the GID message are illustrated in Figure 3-14. A description of each data object is provided in the following paragraphs.



Source: Battelle

Figure 3-14. GID Message Diagram

Message Attributes

The Message Attributes data object is a bit-mapped byte that provides information about the format and content of the rest of the GID message.

Object Format:

Field	Note
Object Identifier	0x01
Size	0x01
Attributes	Unsigned 8-bit Integer

Source: Battelle

The bits of the message attributes object are as follows. A bit set to a logic one indicates that the described condition is present while a bit set to zero indicates that it is not present.

```
0 (LSB)  Elevation data is included.
1        Node offset resolution; 0-centimeter, 1-decimeter.
2        Message contains the intersections geometric data.
3        Message contains navigational movement data.
4        Reserved
5        Reserved
6        Reserved
7        Reserved
```

Example:

```
01 01 0F  --  Elevation, Geometric, and Navigational data included.
              Node offset resolution is in decimeters.
```

Source: Battelle

Intersection ID

The Intersection ID object is used to globally and uniquely define an intersection within a country or region in a 32-bit field. Assignment rules for this value are established elsewhere and may use regional assignment schemas that vary.

Object Format:

Field	Note
Object Identifier	0x02
Size	0x04
Intersection ID	Unsigned 32-bit Integer

Source: Battelle

Example:

```
02 04 00 00 00 01  -- Intersection ID #0001
```

Source: Battelle

Reference Point

The Reference Point object provides a definitive and precise location of the intersection in the world geodetic system (WGS-84) coordinate system from which short offsets may then be used to define lane node positions using the flat earth projection.

Object Format:

Field	Note
Object Identifier	0x03
Size	Number of bytes in the payload
Latitude	Signed 32-bit Integer
Longitude	Signed 32-bit Integer
Elevation	Signed 16-bit Integer, optional

Source: Battelle

This data object provides the latitude and longitude of the center of the intersection in decimal degrees to one-tenth of a micro-degree where the LSB represents 10-7 degrees. When included, the elevation data is provided in decimeters above or below the reference ellipsoid. The elevation value is an asymmetric range of positive and negative values.

The elevation data is only included if indicated in the Message Attributes data object.

Elevation	Encoding	Decimal Values
Positive Range	0x0000 to 0xEFFF	0 to 61439 decimeters
Negative Range	0xF001 to 0xFFFF	-1 to -4095 decimeters

Source: Battelle

Examples:

```
03 08 00 00 00 00 00 00 00 00          -- Lat and Long coordinates
03 0A 00 00 00 00 00 00 00 00 00 00    -- Lat, Long, and Elevation data
```

Source: Battelle

Approach/Egress/Barrier

The Approach/Egress/Barrier data object indicates the beginning of a structure used to relate one or more motor vehicles lanes for an intersection approach or egress description with any associated pedestrian and special purpose lanes or to describe one or more barriers.

Object Format:

Field	Note
Object Identifier	0x04
Size	0x01
Direction	Unsigned 8-bit Integer

Source: Battelle

The enumerated values of Direction data are as follows.

```
1   Approach
2   Egress
3   Barrier
```

The lanes described in an approach may appear in any order; however, computed lanes must appear after the description of their reference lane. All the lane descriptions must be provided before the next Approach/Egress/Barrier flag is encountered in the GID message.

Examples:

```
04 01 01  -- Beginning of the intersection approach lanes
04 01 02  -- Beginning of the intersection egress lanes
04 01 03  -- Beginning of the intersection barriers
```

Source: Battelle

Lane

The Lane data object indicates the beginning of a lane description and provides the lane number and lane type information.

Object Format:

Field	Note
Object Identifier	0x05
Size	0x02
Lane Number	Unsigned 8-bit Integer
Lane Type	Unsigned 8-bit Integer

Source: Battelle

The data objects that describe each lane must be provided before the next Lane data object is encountered in the GID message. The lane description may be comprised of the following set of data objects. The data objects that are included in the lane descriptions are determined by the geometric and navigational flags in the Message Attributes data object. The data objects can be in any order.

Geometric Data	Navigational Data
• Width	• Connections
• Node List	
• Reference Lane	

Source: Battelle

The enumerated values of Lane Type data object are as follows.

```
1    Motorized Vehicle Lane
2    Computed Lane
3    Pedestrian Lane
4    Special Purpose Lane
```

Examples:

```
05 02 01 01    -- Lane #1, Motorized Vehicles
05 02 04 03    -- Lane #4, Pedestrian
```

Source: Battelle

Lane Attributes

The Lane Attributes data object is a bit-mapped value that describes a combination set of characteristics specific to a lane type.

Object Format:

Field	Note
Object Identifier	0x06
Size	0x02
Lane Attributes	Unsigned 16-bit Integer

Source: Battelle

The Attributes value is obtained by combining (logical OR) the various bit-masks together to produce a single value. A bit set to a logic one indicates that the described attribute is present while a bit set to zero indicates that it is not present.

Motorized Vehicle or Computed Lanes

The attributes for the motorize vehicle and computed lane types relate the allowed navigational maneuvers and other restrictions as follows.

```
0 (LSB)  Lane provides a two-way travel
1        Straight maneuver permitted
2        Left turn maneuver permitted
3        Right turn maneuver permitted
4        Yield
5        No U-turn
6        No turn on red
7        No stopping
8        HOV lane
9        Bus only lane
10       Bus and taxi only lane
11       Shared two-way left turn lane
12       Bike lane
13-15    Reserved
```

Special Purpose Lanes

The attributes for special purpose lanes relate the allowed navigational maneuvers and other restrictions. These attributes deal with lanes describing trains and transit vehicles that are part of an intersection as follows.

```
0 (LSB)  Lane provides a two-way travel
1        Lane is a railroad track
2        Transit vehicle only lane
3        HOV lane
4        Bus only lane
5        Vehicles entering
6        Vehicles leaving
7-15     Reserved
```

Pedestrian Lanes

The attributes for a pedestrian lane relate the type of crosswalk, bicycle-crossing, or other non-motorized lane as follows.

```
0 (LSB)  Lane provides a two-way travel
1        Pedestrian crosswalk
2        Bicycle crossing
3        Railroad track is present
4-15     Reserved
```

Examples:

```
06 02 00 6A    -- Vehicle: Straight and right turns, no turn on red
06 02 00 13    -- Pedestrian: Crosswalk and bicycle crossing, two-way
```

Source: Battelle

Barrier Attributes

The Barrier Attributes data object is a bit-mapped value that describes a combination set of characteristics specific to a barrier.

Object Format:

Field	Note
Object Identifier	0x07
Size	0x02
Barrier Attributes	Unsigned 16-bit Integer

Source: Battelle

The Attributes value is obtained by combining (logical OR) the various bit-masks together to produce a single value. A bit set to a logic one indicates that the described attribute is present while a bit set to zero indicates that it is not present.

The attributes for barriers relate the type of barriers being described. A barrier is any object which normal vehicle traffic cannot transverse as follows.

```
0 (LSB)  Median
1        White line
2        Stripped lines
3        Double-stripped lines
4        Traffic cones
5        Construction barrier
6        Traffic channels
7        No curbs
8        Low curbs
9        High curbs
10       HOV, do not cross
11       HOV, entry allowed
12       HOV, exit allowed
13-15    Reserved
```

Example:

```
07 02 08 00    -- Barrier: HOV lane, entry allowed
```

Source: Battelle

Width

The Width data object conveys the width of a lane or barrier in centimeters. The width can be expressed at various levels of granularity; such as an approach, lane, or individual node. Lower levels supersede higher levels.

Object Format:

Field	Note
Object Identifier	0x08
Size	0x02
Lane Attributes	Unsigned 16-bit Integer

Source: Battelle

Example:

```
08 02 01 90    -- Width is 4 meters (400 centimeters) wide
```

Source: Battelle

Node List

The Node List data object structure provides the sequence of offset values relative to the intersections reference point to determine the coordinates of individual nodes that together describe the lane centerline. The node offset values are given in absolute distance and can have either a one centimeter or one decimeter resolution as indicated in the Message Attributes data object, bit 3.

Object Format:

Field	Note
Object Identifier	0x09
Size	Number of bytes in the payload
Attributes	Unsigned 8-bit Integer
Nodes	Varies in length

Source: Battelle

The bits of the node list attributes are as follows. A bit set to a logic one indicates that the described condition is present while a bit set to zero indicates that it is not present.

```
0   (LSB) Width data is included in this node list.
1   Node data is packed in 12-bit format instead of standard 16-
    bit.
2-7 Reserved
```

A node is a spot on the ground. When describing a path, the first node is the one closest to the intersection for the lane or the beginning point in a roadway segment and its offsets. This is typically located on the stop-line for approach lanes and where the outbound lane begins for egress lanes. Subsequent nodes provide points further away along the lanes center line. A lane must always have at least two nodes; however, as many nodes as necessary to characterize the lane curvature should be included.

The first node offset values are relative to the intersections reference point. The offset values for subsequent nodes are relative to the previous node position. The Node object has the following structure.

Object Format:

Field	Note
Eastern Offset	Signed 16-bit Integer
Northern Offset	Signed 16-bit Integer
Elevation Offset	Signed 16-bit Integer, optional
Width	Unsigned 16-bit Integer, optional

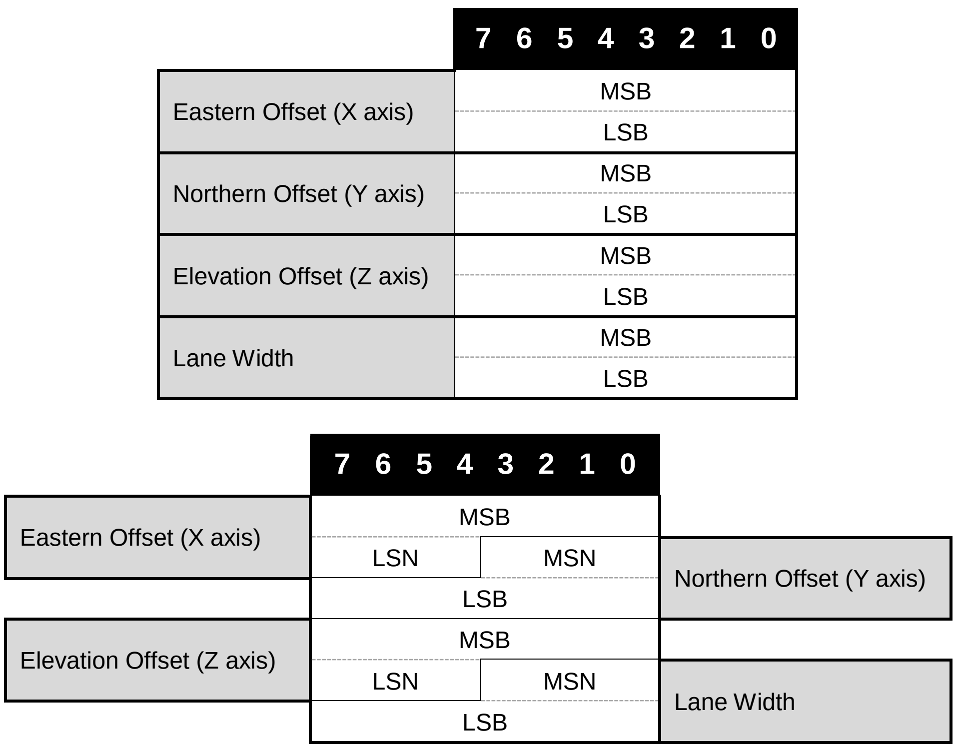
Source: Battelle

Lateral offset values are referenced using the Cartesian coordinate system on an intersection with a northern orientation; thus eastern offsets are indicated by a positive value while western offsets are negative. Likewise, northern offsets are positive while southern offsets are negative.

Node elevation offsets may also be present in the node list as indicated in the Message Attributes data object, bit 0. Positive elevation values indicate a rise in elevation while negative values indicate a drop in elevation relative to the intersections reference point.

Variable widths along the path may be specified in centimeters when width information is present in the node list as indicated in the node list attributes, bit 0.

The node list can be compacted when byte efficiency is critical. The use of the compacted data format is indicated in the node list attributes, bit 1. The compacted node list encodes all node data as 12-bit integers rather than 16-bit integers. This is useful when there are no node offsets that require the 4 most-significant-bits to be used, thus reducing the overall byte count of the node list by approximately 25% as illustrated in the figure below. The maximum offset distance that can be specified is a function of the maximum value of the node offset and the node offset resolution as indicated in the Message Attributes data object.



Source: Battelle

Examples:

```
09 05 00 XX XX YY YY          -- X, Y offsets
09 04 02 XX XY YY             -- X, Y offsets, packed
09 07 00 XX XX YY YY ZZ ZZ    -- X, Y, Z offsets
09 06 02 XX XY YY ZZ Z0       -- X, Y, Z offsets, packed
09 07 03 XX XY YY ZZ ZW WW    -- X, Y, Z offsets and width, packed
```

Source: Battelle

Connection

The Connection data object identifies a previously defined lane to which this lane connects through the use of a specified type of maneuver.

Object Format:

Field	Note
Object Identifier	0x0A
Size	0x02
Lane Number	Unsigned 8-bit Integer
Maneuver Code	Unsigned 8-bit Integer

Source: Battelle

The connected lane must be of the same type as the described lane. The maneuver code is a bit-mapped value as follows.

```
0 (LSB)  Straight Ahead        4    Soft Left Turn
1        Left Turn             5    Soft Right Turn
2        Right Turn            6    Merge Left
3        U-Turn                7    Merge Right
```

Example:

```
0A 02 04 02    -- This lane connects to lane #4, left turn
0A 02 09 01    -- This lane connects to lane #9, straight ahead
```

Source: Battelle

Reference Lane

The Reference Lane data object identifies the reference lane and lateral offset of the center line that a computed lane parallels. This data object is only present the lane type is a computed lane.

Object Format:

Field	Note
Object Identifier	0x0B
Size	0x03
Reference Lane	Unsigned 8-bit Integer
Lateral Offset	Signed 16-bit Integer

Source: Battelle

The referenced lane number must be previously described prior to being referenced by a computed lane. The orientation is from the perspective of the driver where a negative value is a leftward offset and a positive value is rightward. The offset is independent of any width values and uses the node offset resolution. Offset is measured in centimeters.

Example:

```
0B 03 01 01 90    -- References lane #1, 400 centimeters rightward offset
```

Source: Battelle

RTCM Message

This section describes the message generated from the V2I Hub Platform to an external DSRC radio for Global Positioning System (GPS) position correction using the Radio Technical Commission for Maritime Services (RTCM) message. The RTCM message broadcast by the V2I Hub is used by the mobile device to reduce errors in the GPS radio signals and improves position accuracy to the level required to differentiate lane position and location. The data objects that are used by the RTCM message are listed in Table 3-12.

Table 3-12. RTCM Interface Data Objects

Data Object	Object Type	SAE J2735 Paragraph
RTCM-Corrections	Message	5.11
DSRCmsgID	Element	7.30
MsgCount	Element	7.76
RTCM-Revision	Element	7.101
RTCMHeader	Frame	6.52
RTCMmsg	Frame	6.53
GPSstatus	Element	7.39
AntennaOffsetSet	Frame	6.4
RTCM-Payload	Element	7.100

Source: Battelle

The RTCM message object provides an array that contains the raw RTCM sentence in both 2.3 and 3.0 formats. The RTCM message acts as a wrapper around the information that is received from the GPS corrections station through the V2I Hub Platform and RTCM interface described in paragraph 0. The V2I Hub Platform receives the RTCM corrections information and repackages it as a SAE J2735 RTCM message for broadcast to the mobile devices.

Priority Service Request Message

This section describes the message received by the V2I Hub Platform's DSRC radio from an external DSRC radio for the purpose of requesting priority service for an authorized vehicle. The data objects that are used by the Service Request Message (SRM) are listed in Table 3-13.

Table 3-13. SRM Interface Data Objects

Data Object	Object Type	SAE J2735 Paragraph
SignalRequestMsg	Message	5.13
DSRCmsgID	Element	7.30
MsgCount	Element	7.76
SignalRequest	Frame	6.58
IntersectionID	Element	7.44
SignalReqScheme	Element	7.103
LaneNumber	Element	7.64
NTCIPVehicleclass	Element	7.83
DTime	Frame	6.20
DHour	Element	7.21
DMinute	Element	7.23
DSecond	Element	7.28
VehicleIdent	Frame	6.67
TemporaryID	Element	7.117
BSMBlob	Frame	6.9

Source: Battelle

The following represents previously proposed changes1 to the SAE J2735 SRM message schema. The changes shown in Table 3-14 represent the minimum requirements needed to generate a priority signal request message as specified an NTCIP 1211 implementation.

[1] These changes were initially proposed in the ICD for the SPaT and Related Messages for V2I Applications document.

Table 3-14. SRM Message Schema Changes

Element	Recommended Change
SignalRequest .isCancel	Mandatory
SignalRequest .inLane	Mandatory
SignalRequest .type	Mandatory
timeOfService	Mandatory
vehicleVIN	Mandatory
vehicleVIN.id	Mandatory
vehicleData	Mandatory

Source: Battelle

Priority Service Status Message

This section describes the message generated by the V2I Hub Platform's DSRC radio to an external DSRC radio for the purpose of communicating the status of any priority service requests being serviced by the TSC at an intersection served by the V2I Hub Platform. It relates the current status of the signal and all pending, active, and canceled priority service requests acknowledged by the controller. The data objects that are used by the Service Status Message (SSM) are listed in Table 3-15.

Table 3-15. SSM Interface Data Objects

Data Object	Object Type	SAE J2735 Paragraph
SignalStatustMessage	Message	5.14
DSRCmsgID	Element	7.30
MsgCount	Element	7.76
IntersectionID	Element	7.44
IntersectionStatusObject	Element	7.43
RequestStatus	Frame	See Below
VehicleIdent	Frame	6.67
NTCIPVehicleclass	Element	7.83
SignalReqScheme	Element	7.103
DTime	Frame	6.20
DHour	Element	7.21
DMinute	Element	7.23
DSecond	Element	7.28
RequestState	Element	NTCIP 1211 3.1.1.1.9

Source: Battelle

The following represent proposed changes to the SAE J2735 SSM message schema to support the SSM message described here.2 The changes shown in Table 3-16 represent minimum requirements needed to provide a status report of all priority service requests based on an NTCIP 1211 implementation. The revised SSM message content is described in the following ASN.1 notation.

```
RequestStatus ::= SEQUENCE (SIZE(1..10)) OF SEQUENCE {
   priorityRequestID            PriorityRequestID    OPTIONAL,
   priorityVehicleID            VehicleIdent,
   priorityVehicleType          NTCIPVehicleClass    OPTIONAL,
   requestedServiceTypeDesired  SignalReqScheme      OPTIONAL,
   timeOfServiceDesired         DTime                OPTIONAL,
   endOfServiceDesired          DTime                OPTIONAL,
   timeOfServiceAnticipated     DTIme                OPTIONAL,
   requestState                 NTCIP1211.priorityRequestStatusInPRS
}

PriorityRequestID ::= OCTET STRING (Size(4))
```

Source: Battelle

Table 3-16. SSM Message Schema Changes

Element	Recommended Change
priority	Deprecate
priorityCause	Deprecate
prempt	Deprecate
premptCause	Deprecate
requestStatus	Add
requestStatus.priorityRequestID	Add
requestStatus.priorityVehicleID	Add
requestStatus.priorityVehicleType	Add
requestStatus.requestedServiceTypeDesired	Add
requestStatus.timeOfServiceDesired	Add
requestStatus.endOfServiceDesired	Add
requestStatus.timeOfServiceAnticipated	Add
requestStatus.requestState	Add

Source: Battelle

[2] These changes were initially proposed in the ICD for the SPaT and Related Messages for V2I Applications document.

Basic Safety Message

This section describes the message received by the V2I Hub Platform's DSRC radio for the Basic Safety Messages (BSM) received from mobile devices approaching the V2I Hub Platform.

Incoming BSM messages conforming to the SAE J2735 standard and received from the nomadic devices are passed from the DSRC Message Manager to the V2I Hub Message Router for use by other applications. No message retention is performed by the V2I Hub core system. Aggregation or other processing of BSM messages received from nomadic devices is the responsibility of the application plugin that subscribes to them.

Web Browser

The web browser interface to the V2I Hub Platform is used by the V2I Hub Local / Back Office User to Install, Configure and Manage the V2I Hub system via an interface to a local client computer.

The Open System Interconnection (OSI) protocol stack describing the interconnection between the V2I Hub Platform and the user's local client computer is shown in Figure 3-15.

Application	Web Browser
Transport	TCP
Network	IPv4 and IPv6
Data Link	802.3 Ethernet
Physical	10BASE-T

Source: Battelle

Figure 3-15. V2I Hub Platform Web Browser Protocol Stack

SSH Client

The SSH Client interface to the V2I Hub Platform is used by the V2I Hub Local / Back Office User to Install, Configure and Manage the V2I Hub system via an interface to a local client computer.

The Open System Interconnection (OSI) protocol stack describing the interconnection between the V2I Hub Platform and the user's local client computer is shown in Figure 3-16.

Application	Web Browser
Transport	TCP
Network	IPv4 and IPv6
Data Link	802.3 Ethernet
Physical	10BASE-T

Source: Battelle

Figure 3-16. V2I Hub Platform SSH Protocol Stack

Appendix A: List of Acronyms

ASC	Actuated Signal Controller
BSM	Basic Safety Message
DII	Driver Infrastructure Interface
DSRC	Dedicated Short-Range Communication
GID	Geographical Intersection Description
GPS	Global Positioning System
ICWS	Intersection Conflict Warning System
IVP	Integrated V2I Prototype
LSB	Least-Significant Bit
MIB	Management Information Base
NEMA	National Electrical Manufacturers Association
NTCIP	National Transportation Communications for ITS Protocol
OBE	On-Board Equipment
OID	Object Identifier
OSI	Open Systems Interconnection
RSE	Road-Side Equipment
RTCM	Radio Technical Commission for Maritime Services
SAE	Society of Automotive Engineers
SCP	Signal Control and Prioritization
SNMP	Simple Network Management Protocol
SOI	System Of Interest
SPaT	Signal Phase and Timing
SRM	Service Request Message
SSM	Service Status Message
TSC	Traffic Signal Controller
UTC	Coordinated Universal Time
WGS	World Geodetic System

WSA	Wave Service Advertisement

U.S. Department of Transportation
ITS Joint Program Office-HOIT
1200 New Jersey Avenue, SE
Washington, DC 20590

Toll-Free "Help Line" 866-367-7487
www.its.dot.gov

FHWA-JPO-16-TBD

U.S. Department of Transportation