******THIS IS A NEW APPENDIX. – PLEASE READ CAREFULLY.******

Iowa CPSP

Connected Portable Signal Protocol

Version 3.0

Iowa DOT

- Prepared by:

- ISRF

- Date: May 11, 2023

Table of Contents

Introduction

Problem

Timely and accurate info1mation on work zones is increasingly impottant, not only to road authorities, but to other stakeholders involved with managing road construction and maintenance activities.

The challenges in providing this info1mation are:

Collecting and repotting timely information is time consuming for staff and competes with other project administration duties.

Road construction and maintenance activities are not always repo1ted to operations staff resulting in inaccurate or no dissemination of information to traveler info1mation systems and the traveling public.

Records on the statt time, end time, and location of lane closures are not sufficiently detailed for improved post ·work zone analysis of the transportation management plan (TMP) and perfo1mance measurement.

Given the anticipated deployment of connected vehicles, driver notifications of work zone-related lane closures via in-vehicle displays offer opportunities for increased safety, but also increases the need for accurate information about active lane closures.

Solution

The location and operation of Connected Portable Traffic Signals (CPTS) can be automatically repotted using available technology. CPTS, both trailer mounted and pedestal designs, are routinely used to control two-way traffic in a single lane when lane closures are needed on two lane highways. By automatically obtaining their location and operation status, real-time and historical data can be improved without requiring significant time of agency staff.

Scope

This document is intended for use by CPTS manufacturers, monitoring-software authors, DOT traffic engineers, and CPTS field technicians. It is assumed that the reader has some familiarity with software tetms, TCP/IP communication, connected traffic signal terminology, and procedures used for remote monitoring of field devices.

General Needs

All Iowa connected portable traffic signals must meet the requirements found in the Manual on Uniform Traffic Control Devices (MUTCD), 2009, Section 4D.32 and Standard Road Plans TC-215, TC-216, TC-217, and TC-271. Connected traffic signals currently used on Iowa constrnction projects must also meet Iowa DOT Specification Article 2528.03.F and NEMA TSS specifications.

Document History

As is conunon in our indust1y, te1ms and standards evolve. The original version of this document described Iowa's "CTTS Tempora1y Signal Monitoring Protocol". That protocol had two different sub-protocols ("options"). Both of those protocols are now considered obsolete and are replaced by the protocol described by this document.

This is a reworked document intended to be more in line with new terminology and the national WZDx (Work Zone Data Exchange) standard.

Document Overview

This document describes a communication protocol for near real time monitoring of connected po1table traffic signals on Iowa DOT constrnction and maintenance projects on two-lane highways.

This Connected Po1table Signal Protocol (CPSP) uses a subset of the national WZDx standard. It's designed to supp01t collecting CPTS data from an intermedia1y consolidation server in a format that is easy for custom data collection software to parse and store.

The WZDx standard reference website is large, covers many kinds of devices, and takes a while to understand. This document gives you a framework to understand how WZDx works and then guides you to specific pa1ts of that website, focusing on just the po1tions of that standard needed for use with connected po1table traffic signals.

Disclaimer

Example CPSP documents provided herein are as accurate and up to date as possible at the time of writing this document. However, WZDx is an evolving standard. We recommend checking the referenced websites and doing your own verification. In any inconsistency between WZDx and this document, this document defers to the WZDx standard.

Definitions

For brevity, for the remainder of this document:

Agency:

Government or educational entity receiving connected traffic signal data for analysis.

Client:

Agency software (central-monitoring or TCP-te1minal software) which submits queries to one or more CPSP servers and collects the responses.

Connected Signal Set:

One or more CPTSs operating as a coordinated group.

Controller:

CPTS field controller.

Consolidation Server:

Server polled for CPSP info for one-or-more connected signals.

CPTS:

Connected Portable Traffic Signal. This is the current te1m for what was fo1merly called CTTS (Connected Temporary Traffic Signal) or TIS (Tempora1y Traffic Signal).

CPSP:

Connected Portable Signal Protocol. The protocol described by this document.

CPSP Document:

Connected signal data inJSON fo1mat returned in response to a CPSP query.

CPSP Query:

HTTP request for a CPSP document.

CPSP Server:

consolidation se1ver that provides CPSP info1mation.

GeoJSON:

A format based onJSON used for encoding a variety of geographic data.

PrettyPrinted JSON:

A JSON document with line breaks and indentation.

Primary & Secondary Traffic Signals:

Prima1y controller CPTS and secondary CPTS. (Replaces "master/slave" te1minology.)

Property:

JSON name/value pair inside a CPSP document.

Provider:

CPTS manufacturer or other entity that controls a consolidation server.

TMC:

Traffic Management Center.

Stringified JSON:

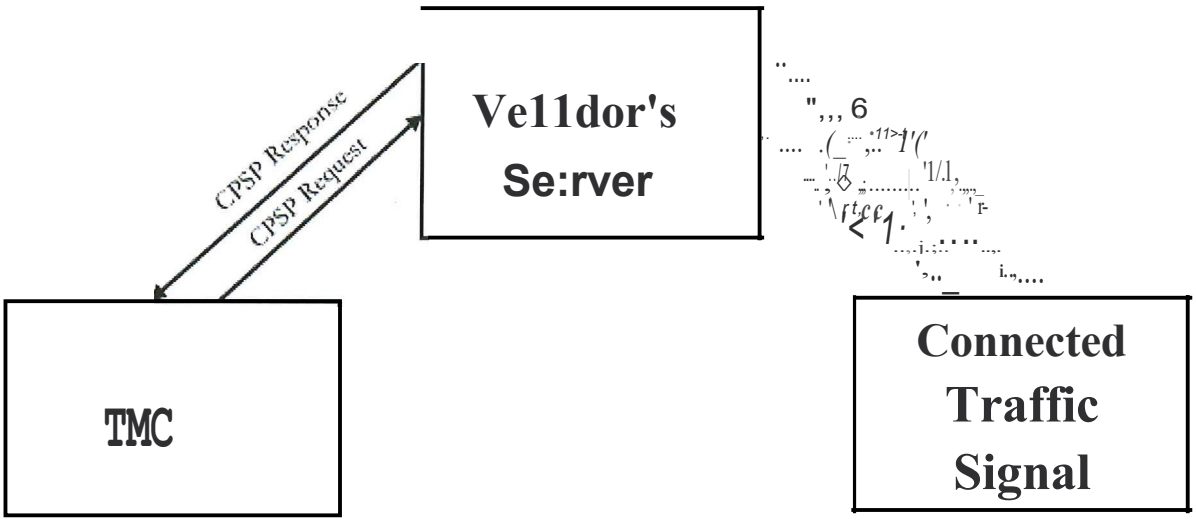
A JSON document with no line breaks or indentation.

WZDx:

Work Zone Data Exchange. A national standard for sharing data about work zone conditions and devices. The fo1mat for a WZDx data feed is based on GeoJSON.

CPSP - Protocol Description

Retrieving data from an inte1media1y se1ver.



(Solid arrow lines show CPSP protocol exchange. Dotted lines are manufacturer specified protocol.)

JSON Overview

JSON is a standard lightweight data-interchange fo1mat used to represent stmctured data in text form.

For purposes of this document, we will use the JSON description available at:
http://www.json.org/

Software libraries for manipulatingJSON documents exist in many programming languages. A pattial (but lengthy) list of available libraries is available at the json.org URL referenced above.

AllJSON documents start with a left curly brace, end with a right curly brace, and contain a series of comma-separated JSON prope1ties.

Each property has a name. All property names begin with a letter and are then followed by a combination ofletters, digits, and underscores. Property names are always surrounded (on both ends of the string) by double quotes (" = ASCII character 34) and are followed by a colon.

Each property name (with quotes and a trailing colon) is followed by a value. A value is one of the following:

- a number (integer or floating-point number),
- a text string surrounded by double quotes,
- a boolean (true or false),
- an array (a comma separated array of values surrounded by square braces),
- an object (a comma separated collection of properties surrounded by curly braces),
- or a null.

All string values are quoted using double-quote characters. Special characters (some control characters, any backslashes, and any double quotes within a string must be replaced with the appropriate escape sequence. All non-string values are not quoted. (See the json.org URL for details.)

The following is a short example JSON document:

```
"aString": "foo",
"bNurnber": 78,
"cNumber": 62. 878,
"dBoolean": true,
"eArray": (1, 2, 3, "bar"],
"£Object": {
    "flSomething": 23,
    "f2SomethingElse": true
},
"gNull": null
```

(This is not a CPSP document. It just shows how JSON represents some kinds of data.)

GeoJSON Overview

GeoJSON is JSON with the addition of standardized ways to represent information about geographic features (points, lines, polygons, etc.) along with non-spatial attributes of those features.

The prima1y document describing GeoJSON is rfc7946. A copy can be found here:

https://datatracker.ietf.org/doc/html/rfc7946

More information about how GeoJSON represents GPS coordinates is available at that web address and in the **"Appendix B - Special Property Values»** section of this document.

WZDx Overview

WZDx (Work Zone Data Exchange) is GeoJSON with data stmctures, properties, and values defined specifically to encode many kinds of work zone information. It also describes procedures for transferring that information.

The WZDx specification is available on GitHub:

https://github.com/usdot-jpo-ode/wzdx#readme

We do **not** recommend tiying to read the whole website. This document will provide pointers to parts of that site as needed. That link to the start of the website's documentation is provided as a reference if you want to dig deeper into WZDx after you finish here.

CPSP Overview

CPSP is "\VZDx for Connected Portable Traffic Signals". It looks like and works like WZDx, but only uses the pa1ts ofWZDx needed to monitor connected traffic signals. It is intended primarily as a way for CPTS manufacturers to provide consolidated data from their traffic signals using a unified fo1mat and web interface.

Communication Process

In its simplest form, CPSP cornmunication is a 3-step HTTP process:

1. The client submits an HTTP-GET request to the server using a **URL** specified by the provider.
2. The server responds with a single CPSP document in the body of the response.
3. When the response is complete, server and client both terminate the **HTTP** connection.

For additional WZDx communication recommendations, go to the web page noted below and scroll down to the section marked "Data Security Best Practices". Please note that these are recommendations, not requirements.

https://github.com/usdot-jpo-ode/wzdx/blob/main/Creating_a_WZDx_Feed.md

Structure of a CPSP (JSON) Document

Each CPSP document consists of at least three base-level properties. A "feed_info" property which contains info1mation about the data feed. A "type" property which tells us what type of data follows (usually a "FeatureCollection"). And a "features" property which contains an array of signal sets. Each signal set object contains sub-prope1i:ies describing the location, mode, and status of the primaty signal controller for a set of one or more connected signals.

In a real CPSP document, text shown between < and > brackets in the example will be replaced with the appropriate information described and the < and> brackets will be removed.

Example:

```
"feed info": {
      <various feed info properties>,
      "data sources": [

                  <properties for first data source>


},
"type": "FeatureCollection",
"features":

            <properties for the first signal set>
      },
      {
            <properties for the second signal set>
      },
      <And so on until ... >
      {
            <properties for the last signal set>
```

In the next section, you will find an example CPSP response document containing information for one data source and one connected signal set.

Note that in addition to containing information about multiple devices, a CPSP data feed may contain infotmation from more than one data source. Each feature propeti:y in the features array contains a "data_source_id" field that identifies which data source listed in the "data_sources" array provided information about that device.

Also note that while CPSP only addresses connected traffic signals, some data sources may have a full WZDx data feed available that includes desired info1mation about traffic signals. For this reason, it is recommended that CPSP compatible clients be written to be able to recognize and extract connected portable traffic signal information from a full WZDx data feed.

Example CPSP Document

The following is an example CPSP document containing information for one connected signal set.

The data in a CPSP document may look like this or may be in raw "serialized JSON" format with no line breaks or indentation. (A long single line of text with no obvious strncture.)

The client software that polls a CPSP se1ver is responsible for recognizing CPSP documents in either fonnat and converting the document to a form the receiving agency finds useful.


```
"feed info": (
        "update_date": "2021-12-06T15: 00: 00Z",
        "publisher": "TestVendor",
        "contact_name": "Robert Vendor",
        "contact_email": "robert. vendor@testvendor.com",
        "update_freguency": 60,
        "version": "4.2",
        "license": "https: //creativecommons .org/publicdomain/ zero/1. 0/",
        "data sources": [

                "data_source_id" : "f4922e55-002c-43cf-9724-c9c9fa362710",
                "organization_name": "Test Vendor Inc.",
                "contact_name": "Timmy Tester",
                "contact_email": "timmy. testor@testvendor.com",
                "update_freguency": 60,
                "update_date": "2021-12-06T14: 54:12Z"


}'
"type" : "FeatureCollection",
"features":

        "id": "c8cbcaa6-0d2f-46bl-a033-afcl 64a86f9e",
        "type": "Feature",
        "properties": {
                "core details":
                        "device_type": "traffic-signal",
                        "data source id": "f4922e55-002c-43cf-9724-c9c9fa362710",
                        "road names": [
                                "US 35E"
                        l,
                        "device status": "ok",
                        "has automatic location": true,
                        "road_direction": "northbound",
                        "name": "Sample Traffic Signal #1",
                        "is_moving": false,
                        nmake" : "Vendor #1",
                        "model": "Model ABC",
                        "update_date": "2021-12-0 6Tl4: 54:12Z"
                }'
                "mode": "flashing-red"
        }'
        "geometry": {
                ''type'': ''Point'',
                "coordinates": [
                        -93.196120,
                        44. 797554
```

Appendix A - WZDx References

Overall stmcture and mles for a CPSP (WZDx) data feed are described in:

https://github.com/usdot-jpo-ode/wzdx/blob/main/Creating a WZDx Feed.md

Special attention should be paid to the following sections of that web page:

- Feed Format and File Type
- Feed Content: DeviceFeed Object
- Business Rules
- Update Guide**
- Data Security Best Practices
- JSON Schemas

** At the time of writing this document, WZDx v4.2 is the latest version, and all examples included are based on that standard. The "Update Guide" section of the above web page should provide clues as to what the current version is and what may have changed since v4.2.

CPSP property stmctures and values are described on these web pages:

https://github.com/usdot-jpo-ode/wzdx/blob/main/spec-content/objects/DeviceFeed.md

https://github.com/usdot-jpo-ode/wzdx/blob/main/spec-content/objects/Feedinfo.md

https://github.com/usdot-jpo-ode/wzdx/blob/main/spec-content/objects/FeedDataSource.md

https://github.com/usdot-jpo-ode/wzdx/blob/main/spec-content/objects/FieldDeviceFeature.md

https://github.com/usdot-jpo-ode/wzdx/blob/main/spec-content/objects/TrafficSignal.md

https://github.com/usdot-jpo-ode/wzdx/blob/main/spec-content/objects/FieldDeviceCoreDetails.md

https://github.com/usdot-jpo-ode/wzdx/blob/main/spec-content/enumerated-types/TrafficSignalMode.md

Note: Depending on the size of your computer screen, you may need to use a horizontal slider bar that appears near the bottom of those pages (just above the "Used By" section, only appears if needed), to see all five of the Properties columns (Name, Type, Description, Confo1mance, and Notes).

Appendix B - Special Property Values

The following provides tips for specific technical values included in CPSP documents.

Timestamp Values

All CPSP timestamps are string values containing date and time. They are fo1matted using the basic ISO 8601 standard. To avoid issues with configuring connected signals for different time-zones and dealing with daylight savings time switchovers, known CPSP timestamps are always fo1matted using the Zulu (UTC) time zone. A null value is used to indicate any timestamp that is unknown.

Example:

```
"2012-04-23Tl8:25:43.500Z"
```

Additional info about formatting timestamp values can be obtained at

http://en.wikipedia.org/wiki/ISO 8601

and

http://datatracker.ietf.org/doc/html/rfc3339#section-5.6

Identifier Values

Data Source IDs (FeedDataSource "data_source_id") and Traffic Signal IDs (FieldDeviceFeature "id") are unique strings used to identify a data publisher and to identify each field device.

For simplicity, CPSP recommends using UUID (Universally Unique Identifier) strings for both types of identifiers to avoid needing to manually assign a project or owner specific name to each. Since this is only a recommendation, no CPSP implementation should require that either type of identifier confo1ms to UUID fo1matting standards (length or content).

Example UUID string:

```
"6dl7db66-de 16-11ed-bSea-0242acl20002"
```

Additional info about UUIDs can be obtained at:

https://www.ietf.org/rfc/rfc4122.txt

GPS Coordinate Values

Known GPS coordinates in a CPSP document are represented in a "geometry" object containing a "type" prope1ty of "Point" and a "coordinates" property containing an array of two signed decimal GPS coordinates in longitude, latitude order.

Note that this order of values is different from many other uses of GPS coordinates. A small writeup on the subject can be found here:

https://macwright.com/lonlat/

Longitude values range from -180.0 to 180.0. Positive longitudes are east of the Prime meridian. Negative longitudes are west of the Prime Meridian. Latitude values range from -90.0 to 90.0. Positive latitude values are above the equator. Negative latitude values are below the equator.

The following is an example of a typical geomet1y propetty:

```
"geometry":    {
        "type":  "Point",
        "coordinates":    [
                -93.196120,
                44.797554
```

Unknown GPS coordinates are represented by replacing the geometty object with null:

```
"geometry":  null
```

There are other ways sometimes used to represent unknown GPS coordinate in a GeoJSON document, but many of them are not universally accepted by all parsers. We recommend using the notation above as it's explicitly mentioned in the document that defines GeoJSON.

https://datatracker.ietf.org/doc/html/rfc7946#autoid__Q

Additional info about decimal GPS coordinates can be obtained at:

http://en.wik ipedia.org/wiki/Decimal_degrees