






Ensuring Compliance and Reliability in EV Charging Station Management Systems: A Novel Testing Tool for OCPP 1.6 Messages Conformance



Dwidharma Priyasta^{1,2*}, Hadiyanto¹, Reza Septiawan², Fito Herminawan², Himawan Bayu²

¹ School of Postgraduate Studies, Diponegoro University, Semarang 50241, Indonesia

² Center for Electronics Research, National Research and Innovation Agency, Jakarta 10340, Indonesia

Corresponding Author Email: dwid002@brin.go.id

<https://doi.org/10.18280/jesa.560116>

ABSTRACT

Received: 25 January 2023

Accepted: 11 February 2023

Keywords:

messages testing tool, OCPP version 1.6, central system platform

The Open Charge Point Protocol (OCPP) has been considered the de-facto standard for communication between charge points and the central management system of the charge points. This paper presents a novel messages testing tool based on OCPP 1.6 to evaluate the conformance of the central system to the field category definitions specified by OCPP. The authors propose the test method, discuss the test case scenarios (positive and negative) used by the messages testing tool, and describe how to set up a test case to examine the system under test. Additionally, this study highlights the difference between the messages testing tool and the OCPP 1.6 Compliance Testing Tool (OCTT) provided by the Open Charge Alliance (OCA). To test its reliability, the message testing tool has been applied to an open-source central system platform. The result shows that the system under test is able to pass 100% of the positive test case scenarios, but only 33% of the negative test case scenarios. It is worth noting that similar results may occur in other central system platforms as well. The study's findings underscore the significance of using comprehensive messages testing tools during the development and release of central system platforms. This ensures conformance to the OCPP field category definitions and promotes reliable interoperability between charge points and central management systems.

1. INTRODUCTION

The use of battery-based electric vehicles has become more popular in many countries around the world. Electric vehicles have been seen as one of the important factors for sustainable transportation systems, due to their potential to reduce carbon emission levels, to improve air quality, and to contribute to the resilience of the national electricity grid [1]. The International Energy Agency (IEA) reports a significant increase in electric cars with about 16.5 million on the world's roads in 2021, up from 10 million in 2020. The global stock of electric vehicles (EV) is predicted to reach over 85 million vehicles in 2025 and 270 million vehicles in 2030 (excluding two- or three-wheeled vehicles) [2].

The adoption intent of electric vehicles is influenced by the availability of charging stations, which is primarily linked to subjective norms that support electric vehicles [3]. Although the current price of plug-in electric vehicles (PEVs) remains higher than the equivalent internal combustion engine vehicles (ICEVs), declining battery costs and other advances indicate that PEVs are approaching price parity with ICEVs [4, 5]. In addition to socioeconomic factors [6-8], government policies and incentives also play a role in shaping adoption intentions [9-14]. As of 2021, there were only 97 private and state-owned charging stations in Indonesia, but this number is expected to increase over time.

The paper by LaMonaca et al. [15] categorizes EV charging locations into at home with level 1 (slow charging) and level 2 (slow to fast charging using AC) chargers, at public places with level 2 (slow to fast charging using AC) and DC fast

chargers, and at work with level 2 (slow to fast charging using AC) chargers, whereas the IEC 62196-1 standard [16] defines mode 1 as 250/480 VAC at 16 A (slow), mode 2 as 250/480 VAC at 32 A (slow to semi-fast using AC), mode 3 as 250/480 VAC at 63 A (semi-fast to fast using AC), and mode 4 as 400 VDC at 200 A (DC fast charging). The paper by Yilmaz et al. [17] describes three methods for supplying power to battery-based electric vehicles, that is by doing battery swapping at an exchange station, charging by conduction at a charging station, and charging by induction without physical contact at a certain area, while the paper by Afshar et al. [18] classifies different types of EV supply equipment, including the fixed charging station, the mobile charging station, and by using contactless charging technologies. The topic in this paper is closely related to the fixed charging stations for public use that are typically equipped with the Open Charge Point Protocol (OCPP).

The fixed charging station system that is based on the OCPP is shown in Figure 1. Actors and roles that are present in the system are as follows:

- The **charge point operator (CPO)** operates charge points (charging stations) and provides charging services for EV users.
- The **charge point** delivers energy to recharge electric vehicles.
- The **central system** manages charge points in the CPO domain and has information for authorizing EV users for using the charge point.
- The **EV user** directly connects to a charge point for energy request. Prior registration with the CPO is required to access charge points belonging to the CPO.

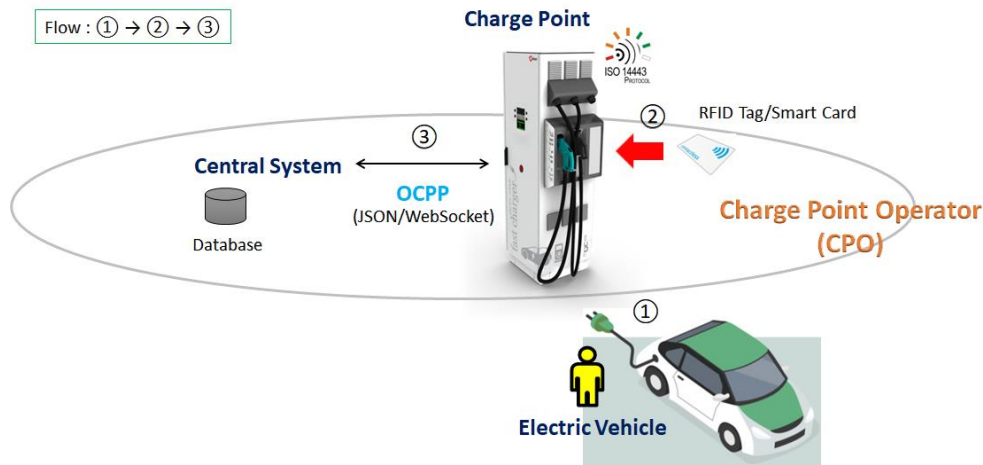


Figure 1. The fixed charging station system

One charging cycle for a transaction that starts at the charge point in accordance with the OCPP version 1.6 scheme can be as shown in Figure 2, and consists of the following steps below:

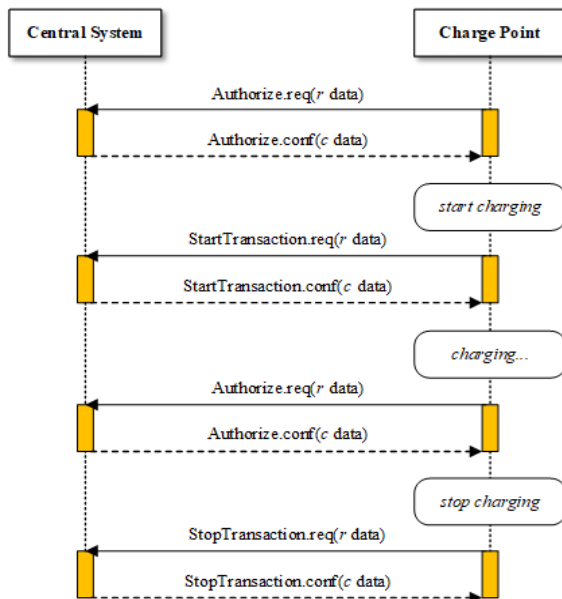


Figure 2. One charging cycle based on OCPP 1.6

(1) The charge point authorizes the EV user who initiates a charging session using RFID tags or smart cards, which are recognized by the central system as valid user tokens. Prior registration with the central system is required to obtain the valid user token. The central system provides the authorization results to be executed by the charge point.

(2) If charging occurs, the charge point informs the central system that a charging session has started.

(3) The charge point periodically reports its meter values during the charging session to the central system.

(4) To stop the charging session, the charge point needs to verify whether or not the EV user is the one that initiated the charging session. Therefore, the EV user should get the authorization once more from the charge point using the valid user token.

(5) Once authorized and the event stop, the charge point notifies the central system that the charging session has ended.

In the large-scale EV charging networks, the central system could also communicate with other central systems to provide

EV roaming to EV users based on roaming agreement between charge point operators. EV roaming enables EV users to make use of any charge point belonging to other networks with one user registration only. However, the topic is beyond the scope of this paper. Please refer to the studies [19-23] for more details on EV roaming.

The central system usually has many features. Aji et al. in the study [24] reported that they developed a central system based on the OCPP 1.6 called SONIK, as shown in Figure 3, and included some features in order to monitor power in real time, to display the availability and location of the charge point, to show the energy consumption, and so forth, which then Renata et al. in the study [25] reported that they used the energy consumption data from SONIK to develop an energy consumption predictive model using Machine Learning to forecast the charge point's energy consumption for the next day based on the data of the previous days. Orcioni and Conti in the study [26] proposed an extension of the OCPP to enable EV users to select the best solution according to their preference in the advanced reservation for using a charge point at the next few hours, where the EV users are negotiating the charging parameters (i.e. initial time, duration, location, price, percentage of final charge, and power required) and the central system provides solutions based on the user flexibility. Hsaini et al. in the study [27] introduced a smart management system that had been developed for managing the operations of photovoltaic-based charging stations by optimizing charging schedules with binary integer programming approach, where users can make reservations via mobile application by specifying the charging parameters such as start time, duration, charging power, and the type of electrical current. On the other hand, Galbis et al. in the study [28] reported that they developed a charging management system that has an adapter based on the OCPP at the backend to address the different needs of EV users and grid operators in the emobility value chain with a more user-central approach.

Among the sophisticated features of a central system, the most important aspect is its conformity to the OCPP. This compliance ensures that the central system can communicate effectively with charging stations, manage and monitor their operations, and provide real-time data on the status of the charging stations and the vehicles being charged. Moreover, compliance with the OCPP ensures that the central system is secure and reliable. Therefore, OCPP conformity is a must for a central system that manage EV charging stations.

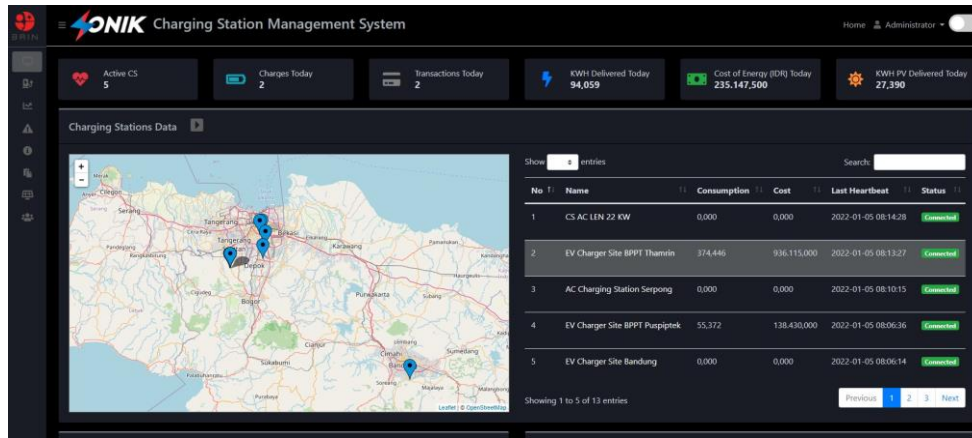


Figure 3. SONIK’s user interface

This paper presents a new approach that uses a combination of positive and negative test case scenarios for assessing the central system. The focus is on areas not currently covered by the OCPP 1.6 Compliance Testing Tool (OCTT), a proprietary tool provided by the Open Charge Alliance (OCA). The output of the new approach, called the messages testing tool (MTT), can be used during the development or prior to the release of the central system platform to evaluate its conformance to the field category definitions specified by OCPP.

2. THE OCPP AND ITS EXISTING TESTING TOOL

2.1 Open Charge Point Protocol (OCPP)

OCPP is an open communication protocol between charge points (technical term for EV charging stations) and a central system that manages the charge points. OCPP was designed to accommodate any type of charging technique [29]. The latest version to date is OCPP 2.0.1 [30]. This paper is only relevant to OCPP 1.6, which is the most popular version embedded in many charging stations nowadays. Hence, the word “OCPP” which appears elsewhere in this paper always means OCPP 1.6.

The functionalities that OCPP has can be used to share the identity of a charge point, to report the current condition of a charge point, to authorize EV users who want to start charging, to inform that a charging session has started, to report meter values of a charging session periodically, and to notify that a charging session has stopped [31]. OCPP can use JavaScript Object Notation (JSON) over WebSocket for communication between the charge point and the central system. WebSocket allows full duplex messaging over a single TCP connection between client and server [32]. In OCPP that employs JSON over WebSocket, the charge point acts as a WebSocket client, while the central system is the WebSocket server.

OCPP provides a set of messages for operations initiated by the charge point such as BootNotification, StatusNotification, Authorize, StartTransaction, and StopTransaction, as well as a set of messages for operations initiated by the central system such as RemoteStartTransaction, and RemoteStopTransaction.

In general, an OCPP message has the following structure:

id	rnd	title	info	object
----	-----	-------	------	--------

where, id is a code to indicate whether it is a request (‘2’) or a response (‘3’) or an error response (‘4’), rnd is a random number as specified by OCPP, title exists in the request to

indicate the correspond OCPP message, info exists in the error response and contains the description of the error, and object which contains JSON object of fields or an empty object in the message. In addition, an OCPP message consists of one or more required and optional fields. The category of fields in a message can be found in the OCPP specification document. One example is as shown in Table 1.

Table 1. Required and optional fields of StartTransaction

Field name	Category
connectorId	Required
idTag	Required
meterStart	Required
reservationId	Optional
timestamp	Required

According to the study [33], which is explicitly included in the OCPP specification document, the term “REQUIRED” means that the definition is an absolute requirement of the specification, while the term “OPTIONAL” indicates that an item is truly optional. Consequently, parties that declare their conformance to OCPP must prepare and check OCPP messages accordingly, and the parties must be able to determine whether or not an OCPP message follows the OCPP schema. Therefore, a new testing tool has been proposed with the capability to evaluate the conformance of the central system to the field category definitions specified by OCPP. The central system is the main focus, as its compliance will directly impact the charge point's ability to comply with the OCPP schema.

2.2 OCPP Compliance Testing Tool

The Open Charge Alliance (OCA) has released the OCPP 1.6 Compliance Testing Tool (OCTT) which can test systems (both the charge point and the central system) following the OCPP version for conformance to the guidelines specified in the OCPP version 1.6 specification document [34]. However, the OCTT is not designed to test for conformance to the field category definitions, as the test cases already been prepared to follow the OCPP schema. The tool focuses on evaluating the behavior and post-conditions of the system under test (SUT) for a given test case.

OCTT was developed in Java as a web-based application. It has 172 test cases, where 102 test cases are intended to test the

charge point and 70 test cases to test the central system. One rule apply to all test cases documented in the study [35] declares that all messages exchanged during any test event shall comply with the OCPP 1.6 schema. This schema describes the structure of an OCPP message, which in general consists of both the required and the optional fields. Therefore, the OCTT does not interested in checking what the SUT responds when given a non-standard field category definitions in the test case, such as no required fields exist, or incomplete required fields and so forth, since it assumes that the SUT must have followed the OCPP schema. Nevertheless, the OCTT also checks for a non-compliant value in the incoming OCPP message received, and will respond with an error code: “correct payload, but value incorrect” for such cases.

Basically, conformance to the specification is crucial for the successful completion of the test case. The OCTT performs the following evaluations on the SUT:

- The SUT should respond accordingly for a given test case (e.g. the cold boot charge point test case expects that first message exchange should be BootNotification, otherwise this test case fails).
- The message exchanges should be compliant to the standard guidelines (e.g. make use of JSON format).
- The message exchanges should be compliant to the schema defined by OCPP (e.g. Authorize, StartTransaction, MeterValues, StopTransaction, etc.).

3. METHODS

The methodology to examine the central system (the system under test, SUT) is based on the following rules:

- (1) For all test cases in the category of positive scenario, which includes complete fields, required fields only, and no fields exist, the SUT must respond in accordance with [31] in order to PASS, otherwise FAIL. It is important to note that the test case scenario of no fields exist is a positive scenario for Heartbeat only; otherwise, it is a negative scenario.
- (2) For all test cases in the category of negative scenario, which includes required fields not complete, optional fields only, unrecognized field value, wrong field types, wrong fields, and no fields exist, the SUT must respond in accordance with the study [36] or nothing in order to PASS, otherwise FAIL.

The above rules generate 38 distinct test cases, where each test case carries one of the following scenarios: complete fields, required fields only, required fields not complete, optional fields only, unrecognized field value, wrong field types, wrong fields, and no fields exist. These scenarios were implemented into the request messages of BootNotification, Authorize, StartTransaction, MeterValues, StopTransaction, Heartbeat, StatusNotification, and DataTransfer, which are messages for operations initiated by the charge point and classified as the Core profile in OCPP.

Each scenario in the test case can be described as follows:

- **Complete fields:** The test case contains the correspond OCPP request message which includes all the fields specified by OCPP.
- **Required fields only:** The request message includes all the required fields specified by OCPP, but without any optional field.
- **Required fields not complete:** The request message includes some of the required fields specified by OCPP.
- **Optional fields only:** The request message includes all the optional fields specified by OCPP, but without any

required field.

- **Unrecognized field value:** The request message has data which is not recognized.
- **Wrong field types:** The request message includes incorrect types of data, for instance integer instead of string, and so forth.
- **Wrong fields:** The request message includes fields not specified by OCPP.
- **No fields exist:** The request message contains empty object.

The mechanism to examine the conformance of the central system to the field category definitions specified by OCPP is based on the request-response between the messages testing tool (MTT) and the SUT as shown in Figure 4. The request contains the test case, while the response contains data to be analyzed to produce the result. For this purpose, the following procedures have to be conducted:

- (1) Run the central system platform (the SUT).
- (2) Register the MTT as a charge point managed by the SUT.
- (3) Create an arbitrary user along with the user token (RFID tags or smart cards) for authorization to be recognized by the SUT as the registered charge point user.
- (4) Connect the MTT to the SUT based on its endpoint.
- (5) Run the test cases that were provided by the MTT.

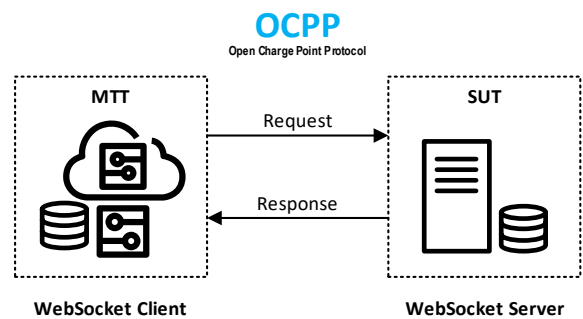


Figure 4. EV Charging Station Management Systems testing design

The decisions to determine whether or not the SUT passes the test are shown in Figures 5 and 6. In the positive scenario, the MTT checks whether the SUT responds within a certain period of time and verifies that the response conforms to the field category definitions specified by OCPP. In the negative scenario, the MTT checks whether the SUT responds with an error notification or nothing in order to pass the test.

The MTT was developed in Java™ 2 Platform, Standard Edition. It has been tested to run on Windows 10 and Linux OS. The user interface of the MTT is as shown in Figure 7, which has the following properties:

- **SUT text input** to specify the MTT’s endpoint in the SUT, along with Connect button to establish connection with the SUT.
- **RFID Tag text input** to specify the user token used by the MTT which is recognized by the SUT.
- **Test Case list** to select a certain test case, along with Execute button to run the selected test case.
- **Payload text pane** to display the request messages sent by the MTT and the response messages from the SUT in JSON format.
- **Execution Log text pane** to display the execution log of test cases.

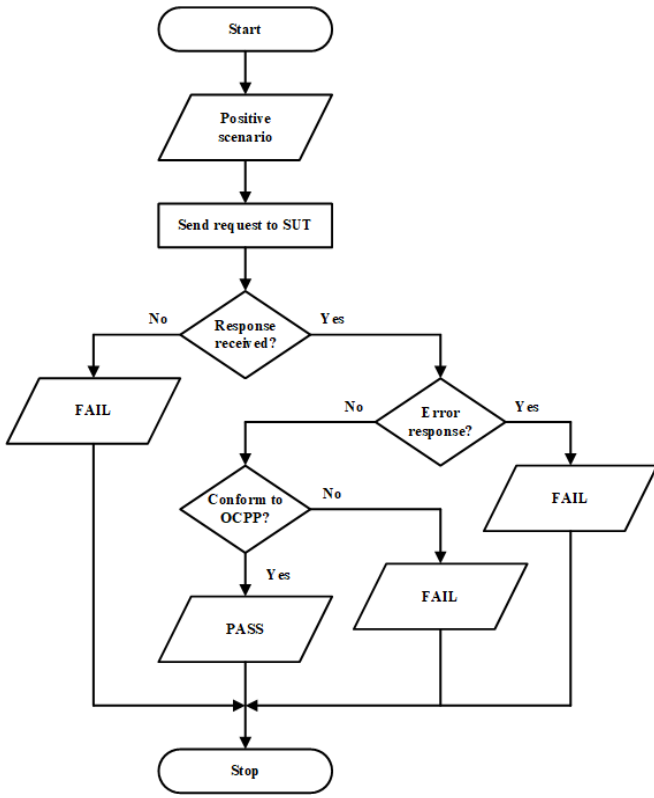


Figure 5. The positive scenario's test decision

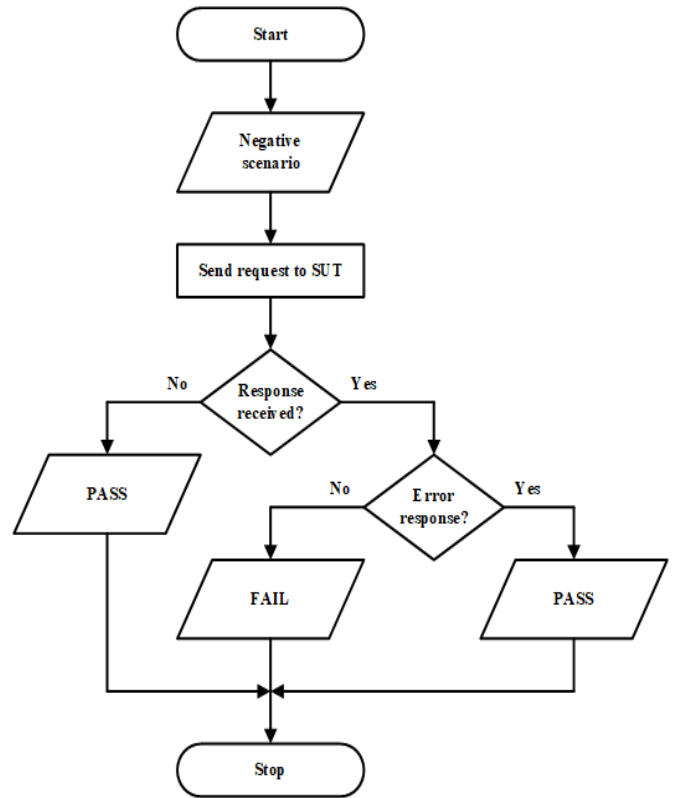


Figure 6. The negative scenario's test decision

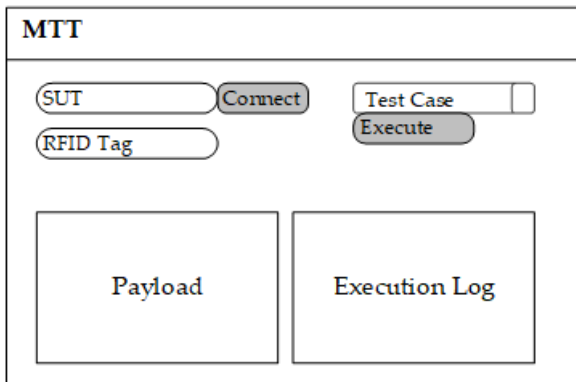


Figure 7. The MTT user interface design

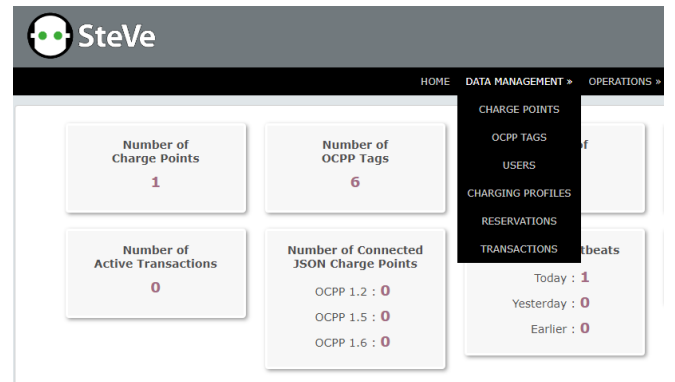


Figure 8. RWTH Aachen University's SteVe user interface

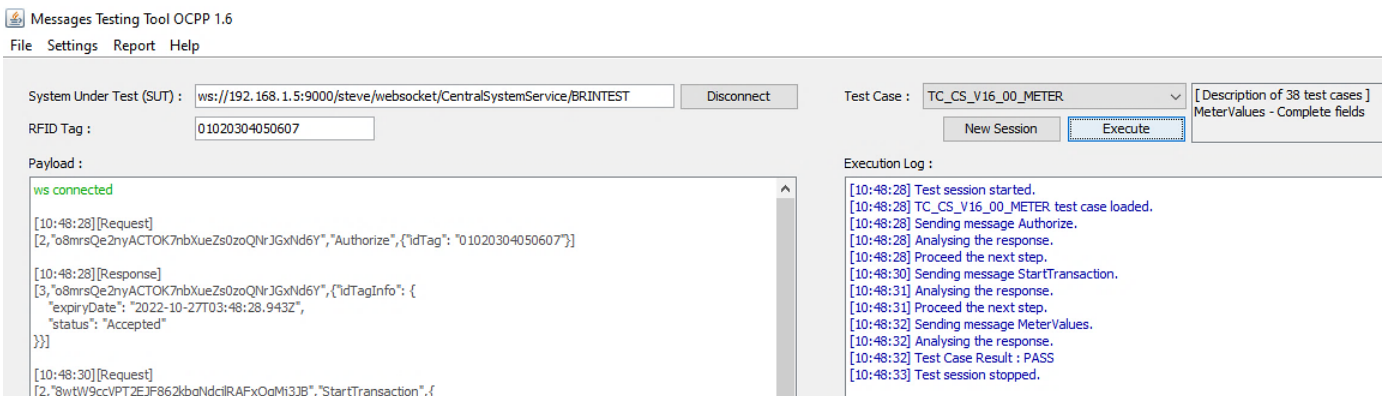


Figure 9. The MTT and the SUT interaction

To test its reliability, the MTT is applied to an open-source central system platform called SteVe [37] that was developed at RWTH Aachen University. SteVe has many sophisticated features to fulfill its role as a central system platform. The user

interface of SteVe has menu to register charge point partners as well as EV users, along with the user tokens (OCPP Tags), as shown in Figure 8.

For this experiment, one charge point is registered with the

identity (chargeBoxId) ‘BRINTEST’, and an arbitrary EV user along with the user token is also registered in SteVe. The MTT uses this registration to connect to SteVe using the following format:

“{endpoint_url}/{chargeBoxId}”, which in the experiment will be

“ws://192.168.1.5:9000/steve/websocket/CentralSystemService/BRINTEST”

as shown in Figure 9. Furthermore, the registered user token (typically in the form of an RFID or smart card) can be used for any test related to a charging session. Once the connection between the MTT and SteVe (the SUT) has been established, the test cases are ready to be executed.

There are two on-screen window panels that can be used for the observation during the execution of a test case. One with label ‘Payload’ which shows the request-response messages between the MTT and the SUT in JSON format, and the other with label ‘Execution Log’ which shows the execution steps as well as the test results.

4. RESULTS

The results by applying the MTT to the SUT (SteVe) is as shown in Table 2. The total number of failures for the given test cases was 16 out of 38, which constitutes approximately 42% and may be considered very significant number. Only the test cases implemented in Heartbeat gave 100% pass.

Table 2. The results of evaluation on SteVe

OCPP message	Test case scenario	Result	Failure rate
BootNotification	complete fields	PASS	0.60
	required fields only	PASS	
	required fields not complete	FAIL	
	optional fields only	FAIL	
	no fields exist	FAIL	
Authorize	complete fields	PASS	0.33
	unrecognized field value	PASS	
	no fields exist	FAIL	
	complete fields	PASS	
	required fields only	PASS	
StartTransaction	required fields not complete	FAIL	0.29
	optional fields only	PASS	
	unrecognized field value	PASS	
	wrong field types	FAIL	
	no fields exist	PASS	
MeterValues	complete fields	PASS	0.60
	required fields only	PASS	
	required fields not complete	FAIL	
	optional fields only	FAIL	
	no fields exist	FAIL	
StopTransaction	complete fields	PASS	0.57
	required fields only	PASS	
	required fields not complete	FAIL	
	optional fields only	FAIL	
	unrecognized field value	PASS	
Heartbeat	wrong field types	FAIL	0
	no fields exist	FAIL	
	no fields exist	PASS	
	wrong fields	PASS	
	complete fields	PASS	
StatusNotification	required fields only	PASS	0.20
	required fields not complete	FAIL	
	optional fields only	PASS	
	no fields exist	PASS	
	complete fields	PASS	
DataTransfer	required fields only	PASS	0.50
	optional fields only	FAIL	
	no fields exist	FAIL	
Summary: PASS=58%, FAIL=42%, N/A=0%			

Observation in Table 2 shows that the SUT could only pass 2 out of 5 test cases implemented in BootNotification. This result brought to the failure rate of 60% which may be classified medium to high. Furthermore, based on the test case scenarios that contribute with the failures, it can be presumed that the SUT did not do a proper check on the incoming message of the negative scenario for its conformance to the field category definitions specified by OCPP. This situation was similar to MeterValues which also gave the failure rate of

60%, and in StopTransaction and DataTransfer that have different number of test cases and gave the failure rate of 57% and 50% respectively.

One test case in Authorize was not responded appropriately by the SUT. The test case contains scenario of no fields exist which the SUT was expected to response with an error or at least no response given. But instead, the SUT responded with an empty field which could lead to the result from a query with an empty string to the database for the authorization. Thus, it

can be presumed that the SUT missed to do a proper check on that specific message of the negative scenario from the MTT.

In StartTransaction, the SUT failed to recognize a missing required field in the message sent by the MTT, which corresponds to the test case scenario of required fields not complete. The SUT was supposed to respond with an error or at least no response given. However, given an incomplete required fields in the message, the SUT responded as if it was receiving a message that meets the OCPP requirements. Thus, it can be presumed that the SUT missed to do a proper check on the message, since it was failed on that specific test case scenario. This situation was similar to StatusNotification which has different number of test cases.

Further analysis of the detailed result was carried out by grouping the results into test case scenario as shown in Table 3. Observation in Table 3 shows that the test case scenario of required fields not complete, optional fields only, wrong field types, and no fields exist gave the failure rate of 100%, 67%, 100%, and 62% respectively. Furthermore, by grouping the results into scenario classification, it was found that 58% of the negative scenario produced the failures, and the SUT performed accordingly for the positive scenario, as shown in Table 4.

Table 3. Failure rate by test case scenario

Test case scenario	Failure rate
complete fields	0
required fields only	0
required fields not complete	1
optional fields only	0.67
unrecognized field value	0
wrong field types	1
wrong fields	0
no fields exist	0.62

Table 4. Failure rate by scenario classification.

Scenario Classification	Failure Rate
the positive scenario	0
the negative scenario	0.67

5. DISCUSSION

In software development life cycle (SDLC) using traditional or agile approach, testing phase has an impact to the quality assurance (QA) of the software, since it provides the final review of the specification, design and program code [38]. Moreover, usually software testing takes out an estimated 40% of total software development costs [39] which constitutes a significant portion. There are two testing strategies in software engineering, namely the positive testing and the negative testing. Positive testing determines that the software works as expected when given a valid input, while negative testing ensures that the software is able to handle invalid input or unexpected conditions which may lead to system failures. It is considered a good practice to combine both strategies together in software development. This combination produces higher assurance of the software being tested as compared to using only one [40-42].

There are 6 basic steps in SDLC to testing anything [43]. These steps are define test criteria to determine whether or not the system under test passed the test when it is done, design test cases for the desired success criteria, build test cases to

create everything necessary using a specific tool, execute tests to be performed by both computers and people, verify test results to ensure that every criteria is covered, and store test cases to be used in future testings. These steps were followed to ensure that the messages testing tool (MTT) presented in this paper has a relatively high level of confidence.

The Open Charge Alliance (OCA) provides the OCPP 1.6 Compliance Testing Tool (OCTT) to examine both the charge point and the central system for conformance to the guidelines specified in the OCPP version 1.6 specification document. The OCTT focuses on evaluating the behavior and post-conditions of the system under test (SUT) for a given test case. The MTT fills the gap not covered by the OCTT by providing a standard and a non-standard field category definitions in the test cases to examine the central system. Therefore, it can be considered the MTT is a complement to the OCTT. This approach relies on the task at the entrance of the central system upon receiving an incoming OCPP message, whether to pass it to the internal system as a standard request and gives standard response in return, or responds immediately with an error or no response given. This approach expands the testing scope of the central system beyond the behavior and post-condition of the SUT to include conformance to the field category definitions specified by OCPP. This is expected to make a significant contribution to the quality assurance (QA) of the central system platform in the future.

6. CONCLUSIONS AND FUTURE WORK

This paper presents a novel messages testing tool (MTT) which is based on the Open Charge Point Protocol (OCPP) version 1.6 to evaluate the conformance of the central system to the field category definitions specified by OCPP. The MTT is designed to complement the existing OCPP 1.6 Compliance Testing Tool (OCTT) provided by the Open Charge Alliance (OCA). The MTT is intended to fill the gap between the field category definitions specified by OCPP and the OCTT which only concerns the behavior and post-condition of the system under test (SUT).

The MTT has been applied to an open-source central system platform for experimental purposes and to test its reliability. It can be concluded that conducting such examination is very important, since the results indicated that the targeted central system was unable to pass some of the test cases given. Similar result may occur in other central system platforms as well. Therefore, it is recommended to use a suitable testing tool such as the MTT presented in this paper during the development of a central system platform or prior to the release of the platform to evaluate its conformance to the field category definitions specified by OCPP.

Currently, the MTT only covers OCPP messages that are classified as the Core profile, which is the only required profile in OCPP. Other profiles are Firmware Management, Local Auth List Management, Remote Trigger, Reservation, and Smart Charging. These optional profiles consist of 12 OCPP messages. Some of them are used for operations initiated by the charge point, and some others are for operations initiated by the central system.

For future work, the plan is to extend the functionalities of the MTT by adding test cases that target the optional profiles. Additionally, evaluating the charge point's conformance to the field category definitions specified by OCPP would also be an interesting subject to explore.

ACKNOWLEDGMENT

The authors would like to thank Beti Tuntari, Melyana, Eka Setianingsih, Wahyu Cesar and Firson Satriasta for their contribution in the development of the work.

REFERENCES

- [1] Ferwerda, R., Bayings, M., Van der Kam, M., Bekkers, R. (2018). Advancing E-roaming in Europe: Towards a single “language” for the European charging infrastructure. *World Electric Vehicle Journal*, 9(4): 50. <https://doi.org/10.3390/wevj9040050>
- [2] International Energy Agency. (2022). *Global EV Outlook 2022*. <https://www.iea.org/reports/global-ev-outlook-2022>.
- [3] White, L.V., Carrel, A.L., Shi, W., Sintov, N.D. (2022). Why are charging stations associated with electric vehicle adoption? Untangling effects in three United States metropolitan areas. *Energy Research & Social Science*, 89: 102663. <https://doi.org/10.1016/j.erss.2022.102663>
- [4] Nykvist, B., Sprei, F., Nilsson, M. (2019). Assessing the progress toward lower priced long range battery electric vehicles. *Energy Policy*, 124: 144-155. <https://doi.org/10.1016/j.enpol.2018.09.035>
- [5] Lander, L., Kallitsis, E., Hales, A., Edge, J.S., Korre, A., Offer, G. (2021). Cost and carbon footprint reduction of electric vehicle lithium-ion batteries through efficient thermal management. *Applied Energy*, 289: 116737. <https://doi.org/10.1016/j.apenergy.2021.116737>
- [6] Ruoso, A.C., Ribeiro, J.L.D. (2022). The influence of countries' socioeconomic characteristics on the adoption of electric vehicle. *Energy for Sustainable Development*, 71: 251-262. <https://doi.org/10.1016/j.esd.2022.10.003>
- [7] Munshi, T., Dhar, S., Painuly, J. (2022). Understanding barriers to electric vehicle adoption for personal mobility: A case study of middle income in-service residents in Hyderabad city, India. *Energy Policy*, 167: 112956. <https://doi.org/10.1016/j.enpol.2022.112956>
- [8] Haidar, B., Rojas, M.T.A. (2022). The relationship between public charging infrastructure deployment and other socio-economic factors and electric vehicle adoption in France. *Research in Transportation Economics*. <https://doi.org/10.1016/j.retrec.2022.101208>
- [9] Langbroek, J.H.M., Franklin, J.P., Susilo, Y.O. (2016). The effect of policy incentives on electric vehicle adoption. *Energy Policy*, 94: 94-103. <https://doi.org/10.1016/j.enpol.2016.03.050>
- [10] Choi, S., Kwak, K., Yang, S., Lim, S., Woo, J. (2022). Effects of policy instruments on electric scooter adoption in Jakarta, Indonesia: A discrete choice experiment approach. *Economic Analysis and Policy*, 76: 373-384. <https://doi.org/10.1016/j.eap.2022.08.015>
- [11] Sahoo, D., Harichandan, S., Kar, S.K., Sreejesh. (2022). An empirical study on consumer motives and attitude towards adoption of electric vehicles in India: Policy implications for stakeholders. *Energy Policy*, 165. <https://doi.org/10.1016/j.enpol.2022.112941>
- [12] Ramesan, S., Kumar, P., Garg, S.K. (2022). Analyzing the enablers to overcome the challenges in the adoption of electric vehicles in Delhi NCR. *Case Studies on Transport Policy*, 10(3): 1640-1650. <https://doi.org/10.1016/j.cstp.2022.06.003>
- [13] Mohammadzadeh, N., Zegordi, S.H., Kashan, A.H., Nikbakhsh, E. (2022). Optimal government policy-making for the electric vehicle adoption using the total cost of ownership under the budget constraint. *Sustainable Production and Consumption*, 33: 477-507. <https://doi.org/10.1016/j.spc.2022.07.015>
- [14] Alali, L., Niesten, E., Gagliardi, D. (2022). The impact of UK financial incentives on the adoption of electric fleets: The moderation effect of GDP change. *Transportation Research Part A: Policy and Practice*, 161: 200-220. <https://doi.org/10.1016/j.tra.2022.04.011>
- [15] LaMonaca, S., Ryan, L. (2022). The state of play in electric vehicle charging services – A review of infrastructure provision, players, and policies. *Renewable and Sustainable Energy Reviews*, 154. <https://doi.org/10.1016/j.rser.2021.111733>
- [16] International Electrotechnical Commission. (2022). *Plugs, socket-outlets, vehicle connectors and vehicle inlets - Conductive charging of electric vehicles - Part 1: General requirements. IEC 62196-1:2022*.
- [17] Yilmaz, M., Krein, P.T. (2013). Review of Battery Charger Topologies, Charging Power Levels, and Infrastructure for Plug-In Electric and Hybrid Vehicles. *IEEE Transactions on Power Electronics*, 28(5): 2151-2169. <https://doi.org/10.1109/TPEL.2012.2212917>
- [18] Afshar, S., Macedo, P., Mohamed, F., Disfani, V. (2021). Mobile charging stations for electric vehicles - A review. *Renewable and Sustainable Energy Reviews*, 152. <https://doi.org/10.1016/j.rser.2021.111654>
- [19] EVRoaming Foundation. (2021). *Open Charge Point Interface 2.2.1*. <https://evroaming.org/app/uploads/2021/11/OCPI-2.2.1.pdf>, accessed on Oct. 3, 2022.
- [20] Hubject. (2020). *Open interchange protocol for charge point operators 2.3*. <https://github.com/hubject/oicp/tree/master/OICP-2.3/OICP%202.3%20CPO>.
- [21] Hubject. (2020). *Open InterCharge Protocol for Emobility Service Providers 2.3*. <https://github.com/hubject/oicp/tree/master/OICP-2.3/OICP%202.3%20EMP>.
- [22] Smartlab, ElaadNL. (2016). *Open Clearing House Protocol 1.4*. <https://github.com/c-clearing-net/OCHP>.
- [23] GIREVE. (2020). *eMIP Protocol - Protocol Description 1.0.14*. https://www.gireve.com/wp-content/uploads/2022/09/Gireve_Tech_eMIP-V0.7.4_ProtocolDescription_1.0.14-en.pdf, accessed on Sept. 11, 2021.
- [24] Aji, P., Renata, D.A., Larasati, A., Riza. (2020). Development of Electric Vehicle Charging Station Management System in Urban Areas. In *Proceedings of The 2nd International Conference on Technology and Policy in Electric Power & Energy (ICT-PEP)*, Jakarta, Indonesia. <https://doi.org/10.1109/ICT-PEP50916.2020.9249838>
- [25] Renata, D.A., Fauziah, K., Aji, P., Larasati, A., Halidah, H., Tasurun, D.P., Astriani, Y., Riza. (2021). Modeling of Electric Vehicle Charging Energy Consumption using Machine Learning. In *Proceedings of The 2021 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Jakarta,

- Indonesia.
<https://doi.org/10.1109/ICACISIS53237.2021.9631324>
- [26] Orcioni, S., Conti, M. (2020). EV smart charging with advance reservation extension to the OCPP standard. *Energies*, 13(12): 3263. <https://doi.org/10.3390/en13123263>
- [27] Hsaini, S., Ghogho, M., Charaf, M.E.H. (2022). An OCPP-based approach for electric vehicle charging management. *Energies*, 15(18): 6735. <https://doi.org/10.3390/en15186735>
- [28] Galbis, A.Z., Garcia, M.A., Garcia, A.I.M., Karatzas, S., Chassiakos, A., Lazari, V., Ageli, O. (2022). Smart Tool Development for Customized Charging Services to EV Users. *World Electric Vehicle Journal*, 13(8): 145. <https://doi.org/10.3390/wevj13080145>
- [29] Open Charge Alliance. (2017). Open Charge Point Protocol 1.6. <https://www.openchargealliance.org/protocols/ocpp-16>.
- [30] Open Charge Alliance. (2018). OCPP 2.0. <https://www.openchargealliance.org/protocols/ocpp-201>.
- [31] Pruthvi, T.V., Dutta, N., Bobba, P.B., Vasudeva, B.S. (2019). Implementation of OCPP protocol for electric vehicle applications. *E3S Web of Conferences*, 87: 8-11. <https://doi.org/10.1051/e3sconf/20198701008>
- [32] Coward, D. (2014). *Java WebSocket Programming*. Oracle Press.
- [33] Bradner, S. (1997). Key words for use in RFCs to Indicate Requirement Levels. <https://www.rfc-editor.org/rfc/rfc2119>, accessed on Jul. 8, 2022.
- [34] Open Charge Alliance. (2020). OCPP 1.6 - OCPP Compliancy Testing Tool - USER MANUAL Version 1.4.2. <https://www.openchargealliance.org/protocols/test-tool>.
- [35] Open Charge Alliance. (2022). Test case document OCTT for OCPP 1.6 Version 1.4.3. <https://www.openchargealliance.org/protocols/test-tool>.
- [36] Open Charge Alliance. (2015). Open Charge Point Protocol JSON 1.6, OCPP-J 1.6 Specification. <https://www.openchargealliance.org/protocols/ocpp-16>.
- [37] The Intelligent Distributed Systems Group, RWTH Aachen University. SteVe. <https://rwth-i5-idsg.github.io/about/steve>, accessed on Feb. 7, 2020.
- [38] Najihi, S., Elhadi, S., Abdelouahid, R. A., Marzak, A. (2022). Software Testing from an Agile and Traditional view. *Procedia Computer Science*, 203: 775-782. <https://doi.org/10.1016/j.procs.2022.07.116>
- [39] Pal, K. (2020). Framework for reusable test case generation in software systems testing. *Software Engineering for Agile Application Development*, pp. 212-229. <https://doi.org/10.4018/978-1-7998-2531-9.ch009>
- [40] Bentil, S. (2022). What are positive and negative testing scenarios in software testing with examples. <https://testsigma.com/blog/positive-and-negative-testing-scenarios>, accessed on Oct. 10, 2022.
- [41] Strug, J. (2016). Mutation testing approach to negative testing. *Journal of Engineering* 2016. <https://doi.org/10.1155/2016/6589140>
- [42] Belli, F., Linschulte, M. (2008). On "Negative" Tests of Web Applications. *Annals of Mathematics, Computing & Teleinformatics*, 1(5): 44-56.
- [43] Bender RBT Inc. (2003). System Development Life Cycle: Objectives and Requirements. <https://www.benderrbt.com/Bender-SDLC.pdf>, accessed on Oct. 10, 2022.