Improving Uptime Monitoring with OCPP

v1.0 March, 2023

Table of Contents

1. Introduction . . . 2
2. Charging station downtime . . . 2
    2.1. Downtime per location, charger or EVSE? . . . 2
3. What causes downtime? . . . 3
4. How to detect downtime? . . . 3
    4.1. OCPP 1.6 . . . 4
        4.1.1. Websocket ping . . . 4
        4.1.2. Heartbeat interval . . . 4
        4.1.3. StatusNotification . . . 4
        4.1.4. Security event notifications . . . 5
        4.1.5. Uptime counter variable . . . 5
        4.1.6. Appropriate check per failure category . . . 5
        4.1.7. Certification . . . 7
    4.2. OCPP 2.0.1 . . . 7
        4.2.1. Websocket ping and heartbeat interval . . . 7
        4.2.2. StatusNotification . . . 7
        4.2.3. Security event notifications . . . 7
        4.2.4. Device model status information . . . 8
        4.2.5. Uptime counter variable . . . 9
        4.2.6. Device model monitoring . . . 9
        4.2.7. Appropriate check per failure category . . . 9
        4.2.8. Certification . . . 11
5. How to minimize downtime? . . . 11
    5.1. Charging station features . . . 12
    5.2. Configuration options . . . 13
    5.3. Offline behavior . . . 15
        5.3.1. Detecting service availability while offline . . . 15
    5.4. Remote assistance . . . 16

Abbreviations

CS	Charging Station
CSO	Charging Station Operator, also known as Charging Station Provider
CSMS	Charging Station Management System. The CSO backoffice system to which all charging stations connect.
EMSP	e-Mobility Service Provider, also known as Card Provider or Mobility Operator
EV	Electric Vehicle
EVSE	Electric Vehicle Supply Equipment. The part of the charging station that supplies energy to the EV.

1. Introduction

This white paper describes how OCPP can be used to detect and (in some cases) avoid downtime of a charging station. Note, that OCPP by itself is not responsible for downtime — it is just a set of messages between a charger and its management system. A charging station operator (CSO), however, can utilize certain messages and configuration options to allow for early detection of problems.

Before we can measure downtime we need to define which components are part of the downtime calculation, because a successful charging session involves more than just the charging station. A backoffice system is involved to authorize the user and track the charging session, and third parties, like an EMSP (card provider) or a smartphone app for authorization might also be involved. Last but not least, the EV itself needs to function properly for a charging session to be successful.

If the end goal is to measure successful charging sessions for a legitimate customer, then the whole chain needs to be taken into account. It may become quite complex to determine what goes wrong and where, and the information about failures may have to come from different locations. Since the CSO is tracking the charging session, this party is best suited to collect information about failures that are reported by or caused by the charging station and failures in the communication with roaming partners (EMSPs).

It is not hard for a CSO to discover that a charging station is not responding, but a failed authorization can have multiple causes. A customer may have an invalid or expired charging card, in which case it is correct to refuse the service, but it can also be caused by the fact that CSO has outdated information about validity of cards, or the communication with the EMSP is not working. Even when a customer is able to charge the EV, then the service may still not be on par with customer expectation. If a 150 kW charging station delivers only 15 kW, then this may be perceived as a failure by the customer. But at the same time, this situation can also be caused by the state of charge of the battery of the EV being too high or its temperature too low.

The bottom line is that it will be really hard to measure the rate of successful charging sessions and to attribute failures to the correct component in the chain. This paper will not try to come up with a solution for this, but instead focus on how a CSO can use OCPP to monitor the charging station.

2. Charging station downtime

For the purpose of this paper we define charging station downtime as the fraction of time that a legitimate customer cannot charge at the station compared with the normal operating time. It can be expressed as a percentage of time that the station cannot be used.

If a charging station is supposed to operate 24/7 and over a 30 day period it has been shown to be inoperative for a total of 3 days, then the downtime is calculated as 3/30 * 100% = 10%. The uptime is, of course, the inverse of downtime and is calculated as 100% - downtime. In the above-mentioned example that equates to: 90%.

2.1. Downtime per location, charger or EVSE?

A charging location may have multiple charging stations, and a charging station may contain more than one EVSE (EV Supply Equipment), in which case the station will also have more than one charging cable or connector. The EVSE is the actual unit that supplies power to the EV. If a charging stations has two EVSEs then it will be able to serve two EVs at a time.

The question arises whether uptime should be calculated per location, per charging station or per EVSE. A CSO might install multiple charging stations at a location to achieve redundancy, and argue that as long as at least one charger is operational, the location should be considered operational. A customer, on the other hand, who connects to a charging station, only to find out that it does not work, might not agree with this point of view. The same reasoning applies to a charging station with multiple EVSEs.

In the end it is up to the legislator to decide the granularity with which uptime is to be reported: location, charging station or EVSE. In order to be prepared for whatever is decided, we recommend to calculate the uptime per EVSE rather than per charging station or location. It is always possible to aggregate this data to the level of charging station or location if that is required.

NOTE There are cases where the EVSE is equipped with two cables with different plug types: CHAdeMO and CCS. Even though there are two cables, there is only one power converter, and only one EV can be served at a time. This is therefore counted as one EVSE for uptime calculations.

3. What causes downtime?

There can be many reasons why a customer cannot charge. It varies from a lack of power from the grid, or having a bad cellular data connection to configuration errors on the management system. In order to facilitate the discussion we divide them in the following categories:

Table 1. Failure categories causing downtime

Type	Description
Grid faults	Power outage; undervoltage; frequency deviation.
Physical damage	Damaged charging station or charging cable.
Electrical safety	RCD tripped or overcurrent protection.
Data communication loss	Loss of (cellular) data connection to backoffice.
Server down	Charging station management system is down.
RFID reader failure	Unable to read RFID charge cards.
EV handshake failure	Initial handshake between EV and charging station fails (includes isolation test).
Authorization error	Valid card is not accepted, e.g. error in local whitelist, or roaming failure. This also includes failure to start via smartphone app.
Power electronics failure	Failure to deliver power to EV.
Connector lock failure	Failure to lock or unlock connector.

It is by no means said that failures must be classified in above-mentioned categories, but it makes sense to classify failures for reporting purposes in a limited number of categories similar to the above.

4. How to detect downtime?

Each of the above-mentioned categories requires a different approach to detect or to report the situation. OCPP 2.0.1 has considerably more options to report abnormal situations then OCPP 1.6. We will therefore discuss each

version separately.

4.1. OCPP 1.6

The main mechanisms in OCPP 1.6 that allow a management system to check for availability of the charging station are the **websocket ping mechanism** and the configurable **heartbeat interval**. Next to this the charging station is able to report errors when they occur by sending a **StatusNotification** message.

4.1.1. Websocket ping

The websocket connection, which is a layer on top of the TCP/IP internet connection, has a built-in mechanism to ensure that the connection is not dropped when there is no traffic. At constant intervals, typically in the order of one or several minutes, the charging station sends a "ping" message, to which the management system will reply with a "pong" message to prove that the connection is still open. The length of the interval is specified by the configuration variable `OCPPCommCtrl.WebSocketPingInterval`. The interval length will depend on the properties of the network connection. An interval that is too long, will fail to keep the (mobile) communication channel open. Too short an interval will lead to excessive data costs.

Note, that although this proves that the websocket connection is working, it does not necessarily mean that the charging station application is fully functional.

4.1.2. Heartbeat interval

The heartbeat mechanism in OCPP ensures that at regular intervals a message is exchanged between charging station and management system. During initial connection, when charging station sends a BootNotification.req message, the management system will return a response containing an *interval* for the heartbeat mechanism. This tells the charging station that at least every *interval* seconds a message must be exchanged with the management system. if there is no message to be exchanged, e.g. because the station is idle, then the Heartbeat.req message can be used. The *interval* is usually configured as a value ranging from 15 minutes to 24 hours.

Since this check uses regular OCPP messages it provides more reassurance that the charging station application firmware is functioning as it should.

4.1.3. StatusNotification

The charging station itself can send a StatusNotification message to report the status of the station or a specific connector. The StatusNotification contains a *status* and an *errorCode* in addition to optional *info* and *vendorErrorCode* fields. The latter two fields are non-standardized, free format fields with information that is specific to the charging station vendor.

Status can be one of: `Available, Preparing, Charging, SuspendedEVSE, SuspendedEV, Finishing, Reserved, Unavailable, Faulted`. The values `Unavailable` and `Faulted` might indicate downtime.

Errorcode is one of: `ConnectorLockFailure, EVCommunicationError, GroundFailure, HighTemperature, InternalError, LocalListConflict, NoError, OtherError, OverCurrentFailure, OverVoltage, PowerMeterFailure, PowerSwitchFailure, ReaderFailure, ResetFailure, UnderVoltage, WeakSignal`.

The capabilities of StatusNotification are limited in the sense that it does not handle multiple errors very well. When an error situation is no longer active, this can be reported by sending a StatusNotification without an *errorCode* or with a value of `NoError`. However, when more than one error is reported (by two or more StatusNotification messages), then it is unclear whether all errors are still active, or the first error is no longer applicable. Similarly, it is not possible to selectively clear one error, while keeping another error active.

If desired, the management system can force the charging station to send a StatusNotification message with the status of the charging station by sending it a TriggerMessage.req(*requestedMessage* = "StatusNotification", *connectorId* = 0). If *connectorId* is omitted, then the charging station will send a StatusNotification for both the charging station as a whole and for each connector separately.

NOTE There should be no need for a management system to send a TriggerMessage for a StatusNotification, unless a station has been offline for some period of time, and upon reconnection it does not automatically send StatusNotification messages (which it should).

4.1.4. Security event notifications

Security events are part of the OCPP 1.6 security extension that is described in the OCPP 1.6 Security White Paper. Security events are sent whenever an event happens that could impose a security risk. Not all of these events relate to downtime, but the events shown in the table below might be.

Table 2. OCPP 1.6 security events possibly pointing to downtime

Security Event	Description
StartupOfTheDevice	The Charging Station has booted
ResetOrReboot	The Charging Station was rebooted or reset
MemoryExhaustion	The Flash or RAM memory of the Charging Station is getting full
InvalidMessages	The Charging Station has received messages that are not valid OCPP messages, or signature of signed message incorrect
AttemptedReplayAttacks	The Charging Station has received a replayed message
TamperDetectionActivated	The physical tamper detection sensor was triggered

4.1.5. Uptime counter variable

As long a station remains connected, the backoffice will know how long it has been powered up, but if the connection has been lost for a while, the backoffice cannot know whether it was powered up all the time. For this case it is possible to introduce a read-only configuration variable that reports the time that a charging station has been powered up. Note, that such a variable is not specified in the OCPP specification, but OCPP allows for the introduction of new configuration variables.

When introducing such a variable, it is recommended to call it "UptimeSeconds". This variable holds the number of seconds since last power up.

4.1.6. Appropriate check per failure category

The table below shows for each category what the appropriate mechanisms are to detect the situation.

Table 3. Checks per failure category for OCPP 1.6

Type	Description
Grid faults	Power outage: the charging stations "dies" and the websocket connection is broken. This is first detected by the websocket ping mechanism. When the station comes back online it sends a BootNotification message (and possibly a security event "StartupOfTheDevice"), which can also serve as a clue to the backoffice. Overvoltage and undervoltage: this will be reported as a StatusNotification with *errorCode* = `OverVoltage` or `UnderVoltage`.
Physical damage	There is no OCPP *errorCode* to report this situation. It depends on the charging station and the kind of damage whether it will be able to detect the physical damage. If so, then it can be reported as a StatusNotification with *errorCode* = `OtherError` and a descriptive text in the *info* field, or a security event "TamperDetectionActivated".
Electrical safety	RCD tripped: Reported as a StatusNotification with *errorCode* = `GroundFailure`. Overcurrent protection: When EV is drawing more power than allowed, the charging station will stop the transaction. It is reported as a StatusNotification with *errorCode* = `OverCurrentFailure`.
Data communication loss	A weak signal can be reported as a StatusNotification with *errorCode* = `WeakSignal`. A complete loss will first be noted by the websocket ping mechanism. Charging station will switch to offline mode.
Server down	If data connection is lost, then behavior will be identical to "Data communication lost" above. If the websocket remains open, but the server is not responding, then this will be discovered by charging station firmware when it is not receiving any response. Charging station will switch to offline mode.
RFID reader failure	This is reported by a StatusNotification with *errorCode* = `ReaderFailure`.
EV handshake failure	The initial handshake between EV and charging station fails. This applies to both the mode 3 protocol for AC chargers and ISO 15118 or DIN for DC chargers. It is reported as a StatusNotification with *errorCode* = `EVCommunicationError`. This error may have been caused by the EV and should in that case not contribute to downtime.
Authorization error	An authorization error cannot be reported by the charging station as an error, because this is discovered by the management system. Failure to authorize is not necessarily an error. The charge card may simply be unknown or expired. If the management system is equipped with roaming interfaces to allow access to third party charge cards, then frequent authorization errors should trigger the CPO to check whether all roaming interfaces are operational.

Type	Description
Power electronics failure	It depends on the capabilities of the charging station which kind of failures can be detected. A StatusNotification with *errorCode* = `PowerMeterFailure` or `PowerSwitchFailure` can be used to report errors with the meter or power relais. Other types of errors will have to reported with `OtherError` and an *info* field or *vendorErrorCode* providing more information.
Connector lock failure	A failure to lock or unlock connector is reported with a StatusNotification with *errorCode* = `ConnectorLockFailure`. This situation will also lead to a StatusNotification with *status* = `Faulted` for the connector.

4.1.7. Certification

The OCPP 1.6 certification verifies a proper implementation of the above-mentioned mechanisms, as follows:

- Websocket ping — Ping mechanism is checked during certification. The PICS states whether the websocket ping is configurable, and which interval has been used during testing
- Heartbeat interval — Heartbeat mechanism is checked during certification. The PICS states the mininum and maximum supported heartbeat interval.
- StatusNotification — StatusNotifications are checked during certification with respect to *status* of a connector, e.g. `Available`, `Charging`, `Faulted`. There are no test cases that check that a correct *errorCode* is reported, e.g. for situations of overvoltage, overcurrent failure or reader failure, since these situations are mostly related to hardware or grid issues and therefore hard to test with the compliance testing tool.
- TriggerMessage for StatusNotification — The TriggerMessage is not part of the "Core" profile and as such not mandatory to be implemented. It can be certified as part of the "Remote Trigger" profile. TriggerMessage is, however, not needed to implement the above-mentioned checks.

4.2. OCPP 2.0.1

4.2.1. Websocket ping and heartbeat interval

OCPP 2.0.1 builds on top of OCPP 1.6, so it has the same mechanisms for **websocket ping** and **heartbeat interval** as described in OCPP 1.6.

4.2.2. StatusNotification

The StatusNotification message is slightly different in OCPP 2.0.1. Since errors are now reported via device model messages, the *errorCode* is no longer present in StatusNotification. The StatusNotification message is used to report the *connectorStatus* with values: `Available, Occupied, Reserved, Unavailable, Faulted`. The values `Unavailable` and `Faulted` might indicate downtime.

4.2.3. Security event notifications

Security events of OCPP 2.0.1 are almost identical to those of the OCPP 1.6 security extension, the only exception

being the additional MaintenanceLoginAccepted/Failed events. Security events are sent whenever an event happens that could impose a security risk. Not all of these events relate to downtime, but the events shown in the table below might be.

Table 4. OCPP 2.0.1 security events possibly pointing to downtime

Security Event	**Description**
StartupOfTheDevice	The Charge Point has booted
ResetOrReboot	The Charge Point was rebooted or reset
MemoryExhaustion	The Flash or RAM memory of the Charge Point is getting full
InvalidMessages	The Charge Point has received messages that are not valid OCPP messages, or signature of signed message incorrect
AttemptedReplayAttacks	The Charge Point has received a replayed message
TamperDetectionActivated	The physical tamper detection sensor was triggered
MaintenanceLoginAccepted	Successful login to the local maintenance interface.
MaintenanceLoginFailed	Failed login attempt to the local maintenance interface.

4.2.4. Device model status information

OCPP 2.0.1 has a so-called "device model" that describes the status and configuration of components inside the charging station. This allows for much more detailed status monitoring. The OCPP device model allows for any components inside the charging station to be reported. For this purpose the specification contains a list of 75 standardized components to use. If that is not enough, a manufacturer is allowed to add more components in the report. Standardized components are listed in the OCPP specification: "OCPP-2.0.1_part2_appendices_v13"

There is a limited subset of 17 components that are mandatory to be supported and reported by the charging station. The following **mandatory** components and variables are relevant in the context of monitoring charging station availability.

Table 5. Mandatory components related to charging station availability

Component	**Variable**	**Purpose**
AuthCtrlr	LocalAuthorizeOffline	To allow offline transactions for cached tokens
ChargingStation	Problem	True when component has a problem
ChargingStation	AvailabilityState	Status of the station, e.g. Available, Occupied, Faulted, ...
EVSE	Problem	True when component has a problem
EVSE	AvailabilityState	Status of the EVSE, e.g. Available, Occupied, Faulted, ...

A DC fast charging station is likely to report many more components, such as an AcDcConverter or AirCoolingSystem, but these are not mandated by OCPP to be reported.

The *errorCode* of an OCPP 1.6 StatusNotification is no longer present in OCPP 2.0.1. Instead, any errors are reported by the NotifyEvent message. This message is used to report events for a specific component, and can contain optional free format fields *techCode* and *techInfo* with vendor-specific information. These events can be errors, warnings or measurements at regular intervals, that can be predefined in the firmware or have been

configured as 'monitors' by the CSO.

4.2.5. Uptime counter variable

As long a station remains connected, the backoffice will know how long it has been powered up, but if the connection has been lost for a while, the backoffice cannot know whether is was powered up all the time. For this case it is possible to introduce a variable that reports the time that a charging station has been powered up. Note, that such a variable is not specified in the OCPP specification, but the OCPP device model allows for the introduction of new variables.

When introducing such a variable, it is recommended to call it "UptimeSeconds" and report it as part of the component "ChargingStation". This variable holds the number of seconds since last power up.

4.2.6. Device model monitoring

The concept of a monitor in the device model exists to define when a change in a certain variable should be reported to the management system. The charging station will normally contain some built-in monitors that report common conditions, like the availability of an EVSE or the occurrence of a failure (e.g. RCD tripped or high temperature). The charging station operator, however, can also install custom monitors for any of the reported variables.

Monitors can be defined for upper and lower thresholds, value changes (delta) and periodic measurements.

An upper threshold monitor defines a threshold for the value of a variable above which a NotifyEvent message will be sent with a certain severity. For example, an event of severity "warning" can be sent when the temperature of an AcDcConverter exceeds 80 C, and an event of severity "error" can be sent at a temperature over 100 C.

A periodic monitor can be used to periodically report the value of a variable. For example, a periodic monitor can be used to report the AvailabilityState of an EVSE every 15 minutes, or report the charging station overall temperature during the day.

4.2.7. Appropriate check per failure category

Some checks described in OCPP 1.6, like websocket ping and hearbeat interval, also apply to OCPP 2.0.1. The reporting of errors is handled by the NotifyEvent message. Errors for many types of components can be reported, but most components are not mandatory to be reported, because it is not up to the OCPP specification to define which components are mandatory in a charging station.

Table 6. Checks per failure category for OCPP 2.0.1

Type	Description
Grid faults	Power outage: the charging stations "dies" and the websocket connection is broken. This is first detected by the websocket ping mechanism. When the station comes back online it sends a BootNotification message (and possibly a security event "StartupOfTheDevice"), which can also serve as a clue to the backoffice. Overvoltage and undervoltage: this can be reported by a NotifyEvent message for component "ElectricalFeed" and variable "ACVoltage" when crossing the respective upper or lower threshold for this value.
Physical damage	Depending on the type of damage, this can be reported by a NotifyEvent message for the variable "Active" of the component "ShockSensor" or "TiltSensor" (e.g. in case of a collision with the charging station housing) or the component "CableBreakAwaySensor" (e.g. when the cable has been damaged or forcibly removed), or a security event "TamperDetectionActivated"..
Electrical safety	RCD tripped: This can be reported as a NotifyEvent for the variable "Tripped" of component "RCD". Groud isolation protection: This can be reported by the variable "Active" or "Problem" of component "GroundIsolationProtection".
Data communication loss	A complete loss will first be noted by the websocket ping mechanism. Charging station will switch to offline mode. Signal loss can be reported by the variable "SignalStrength" or "Problem" of component "DataLink".
Server down	If data connection is lost, then behavior will be identical to "Data communication lost" above. If the websocket remains open, but the server is not responding, then this will be discovered by charging station firmware when it is not receiving any response. Charging station will switch to offline mode.
RFID reader failure	This can be reported by the variable "Problem" of component "TokenReader".
EV handshake failure	The initial handshake between EV and charging station fails. This applies to both the mode 3 protocol for AC chargers and ISO 15118 or DIN for DC chargers. This error may have been caused by the EV and should in that case not contribute to downtime. The situation can be reported by several variables, depending on the type of connector. - For a mode 3 connector: "CPPMWController", variables "Problem" and "State". - For a CHAdeMO connector: "CHAdeMOCtrlr", variables "Problem", "Tripped". - For an ISO 15118 connector: "ISO15118Ctrlr", variables "Problem", "Tripped". - If charging station supports the ConnectedEV component, then information reported by EV can also be shown, for example "ProtocolAgreed", which is empty if EV and EVSE could not agree on a protocol version, and "ChargingState", which reports things like high battery temperature or charging system error.

Type	Description
Authorization error	An authorization error cannot be reported by the charging station as an error, because this is discovered by the management system. Failure to authorize is not necessarily an error. The charge card may simply be unknown or expired. If the management system is equipped with roaming interfaces to allow access to third party charge cards, then frequent authorization errors should trigger the CPO to check whether all roaming interfaces are operational.
Power electronics failure	It depends on the capabilities of the charging station which kind of failures can be detected. This can be reported by a NotifyEvent message for the following component/variable combinations: - "AcDcConverter", variables "Problem", "Tripped", "Overload". - "PowerContactor", variable "Problem". - "ELVSupply", variable "Problem" - "OverCurrentProtection", variable "Operated".
Connector lock failure	This can be reported by a NotifyEvent message for the component "ConnectorPlugRetentionLock" and variable "Problem". This situation will also lead to a StatusNotification with *connectorStatus* = `Faulted` for the connector and/or a NotifyEvent message for the EVSE with "AvailabilityState" = `Faulted`.

4.2.8. Certification

The OCPP 2.0.1 certification verifies a proper implementation of the above-mentioned mechanisms, as follows:

- Websocket ping — Ping mechanism is checked during certification. The PICS states whether the websocket ping is configurable, and which interval has been used during testing
- Heartbeat interval — Heartbeat mechanism is checked during certification. The PICS states the mininum and maximum supported heartbeat interval.
- StatusNotification — StatusNotifications are checked during certification with respect to *connectorStatus* of a connector, e.g. `Available`, `Occupied`, `Faulted`.
- NotifyEvent — The support for NotifyEvent messages is checked, but the compliance testing tool cannot verify whether such a message will be sent when a hardware error occurs.
- Monitoring — Support for component monitoring is not part of the "Core" profile and as such not mandatory to be implemented. The compliance testing tool does check monitoring messages and behavior, when the "Advanced Device Model" profile has been implemented.
- TriggerMessage for StatusNotification — TriggerMessage is checked during certification. It is, however, not needed to implement the above-mentioned checks.

5. How to minimize downtime?

Following is a list of actions that a CSO can implement to monitor a charging station and detect problems as soon as possible.

1. Monitor websocket ping to discover as soon as possible when the connection is broken.
2. Set heartbeat interval to a value of 15 minutes (or 1 hour) to ensure that you receive a message from the charging station at least every 15 minutes (or 1 hour).
3. OCPP 1.6: Monitor *status* and *errorCode* from the StatusNotification messages that the charging station sends.
4. OCPP 2.0.1: Monitor NotifyEvent messages and *connectorStatus* from StatusNotification messages that the charging station sends.
5. OCPP 2.0.1: Enable monitoring for crucial components of the charging station.
6. OCPP 2.0.1: Tune the back-off retry parameters to improve reconnection time after connection loss.
7. Implement a check at the management system to verify the health of roaming interfaces to third parties.
8. Enable offline behavior at charging station to enhance perceived uptime for the customer. See Configuration options.

NOTE

There is no added value to requesting a StatusNotification every 15 minutes from the charging station by sending a TriggerMessage. The charging station will send a StatusNotification with an *errorCode* (OCPP 1.6), or a NotifyEvent message (OCPP 2.0.1), when an error occurs, and the heartbeat mechanism will ensure that messages are exchanged every 15 minutes or so.

5.1. Charging station features

CSOs are recommended check with vendors to which extent their charging stations support the following features, since these are not covered by the standard OCPP certification process. Having full support for below-mentioned features will greatly enhance an operator's capability to monitor uptime and determine reasons for downtime.

OCPP 1.6: *errorCode* support

Check that charging station is supporting all or most *errorCodes* in a StatusNotification message (as opposed to reporting no errors or only a generic `OtherError` code.)
ErrorCode values are:

ConnectorLockFailure	EVCommunicationError	GroundFailure	HighTemperature
InternalError	LocalListConflict	NoError	OtherError
OverCurrentFailure	OverVoltage	PowerMeterFailure	PowerSwitchFailure
ReaderFailure	ResetFailure	UnderVoltage	WeakSignal.

OCPP 2.0.1: component support

Check that charging station supports EventNotifications for all or most components, that are mentioned in Checks per failure category for OCPP 2.0.1:

Component	**Variable**
AcDcConverter	Problem, Tripped, Overload, Temperature
AirCoolingSystem	Problem, FanSpeed

Component	Variable
CPPMWController	Problem, State
CableBreakAwaySensor	Active
ChadeMoCtrl	Problem, Tripped
ChargingStation	Problem, Temperature
ConnectedEV	ProtocolAgreed, ChargingState
ConnectorPlugRetentionLock	Problem
DataLink	Problem, SignalStrength
ELVSupply	Problem
EVSE	Problem, Power
ElectricalFeed	ACVoltage
GroundIsolationProtection	Active
ISO15118Ctrlr	Problem, Tripped
LiquidCoolingSystem	Problem, Temperature
OverCurrentProtection	Operated
PowerContactor	Problem
RCD	Tripped
ShockSensor	Active
TiltSensor	Active
TokenReader	Problem

OCPP 2.0.1: monitoring support

Check that charging station supports monitoring functionality for above-mentioned device model variables (where applicable). Monitoring these values periodically or when exceeding a threshold will provide an early warning of possible problems. Since extremely high or low temperatures are known to cause problems to the electronics it is beneficial to monitor the temperature of vital components.

5.2. Configuration options

This section provides some guidance on configuration settings that are have been mentioned above.

Table 7. Configuration settings for OCPP 1.6

Purpose	Configuration method
Websocket ping	Websocket ping is activated by setting the configuration variable *WebSocketPingInterval* to a value greater than 0. It represents the number of seconds between each ping message. Suggested range: 50 - 600.

Purpose	Configuration method
Heartbeat interval	The heartbeat interval is the interval of inactivity (no OCPP exchanges) with the management system after which the charging station should send a Heartbeat message. This interval is controlled by the configuration variable *HeartbeatInterval* as a number of seconds. It initially reflects the value of the field *interval* in the BootNotification.conf message that the station receives from the management system, but can be changed by a ChangeConfiguration message. Suggested range: 300 - 3600.
Offline behavior	The offline authorization of charge cards is enabled by the configuration variables *LocalAuthorizeOffline* and *AuthorizationCacheEnabled* and/or *LocalAuthListEnabled*. See section Offline behavior.
Allow offline transactions for unknown cards	The acceptance of any charge card when the station is offline, is enabled by the configuration variable *AllowOfflineTxForUnknownId*.

Table 8. Configuration options for OCPP 2.0.1

Purpose	Configuration method
Websocket ping	Websocket ping is activated by setting the variable *WebSocketPingInterval* from component *OCPPCommCtrlr* to a value greater than 0. It represents the number of seconds between each ping message. Suggested range: 50 - 600.
Heartbeat interval	The heartbeat interval is the interval of inactivity (no OCPP exchanges) with the management system after which the charging station should send a Heartbeat message. This interval is controlled by the variable *HeartbeatInterval* from the component *OCPPCommCtrlr* as a number of seconds. It initially reflects the value of the field *interval* in the BootNotificationResponse message that the station receives from the management system, but can be changed by a SetVariables message. Suggested range: 300 - 3600.
Offline behavior	The offline authorization of charge cards is enabled by the variables *LocalAuthorizeOffline* of the component *AuthCtrlr* and *Enabled* of the components *AuthCacheCtrlr* and/or *LocalAuthListCtrlr*. See section Offline behavior.
Allow offline transactions for unknown cards	The acceptance of any charge card when the station is offline, is enabled by the variable *OfflineTxForUnknownIdEnabled* of the component *AuthCtrlr*.
Back-off retry time parameters	(BackOffRepeatTimes, BackOffRandomTimeRange, BackOffMinimumWaitTime). Choosing those poorly can extend offline time after a connection loss greatly. We recommend low values for BackOffRepeatTimes and BackOffMinimumWaitTime, and a larger one only for BackOffRandomTimeRange to flatten the peak after a backend outage/restart. For example BackOffRepeatTimes=0, BackOffMinimumWaitTime=5, BackOffRandomTimeRange=300.

5.3. Offline behavior

OCPP supports so-called offline behavior. This is a situation where the charging station has no connection to the management system, e.g. because the cellular data connection is down, or the entire management system is offline, but the station still allows certain operations.

While offline, an OCPP charging station will allow any on-going transaction to continue normally. All progress messages during the transaction, and the final stop message will be buffered by the charging station to be sent as soon as the connection is restored. This means that an offline period has zero effect for transactions that had already started, but authorizating new transactions may not be possible.

There are features that allow certain customers to start charging, even though their charge card cannot be authorized when the station is offline. This is controlled by configuration settings of the *authorization cache* and the *local authorization list*. The authorization cache keeps a copy of the authorization results of recently used charge cards. In an offline situation the charging station will check if the charge card occurs in the cache, and if the charge card was authorized the last time, then it assumes that the card is still valid. When the lifetime of the authorization cache is set to one week or one month, then at least all regular customers of this charging station will be able to charge normally. Even though no live authorization is done when offline, all transaction data is buffered and will be sent to the management system as soon as the station comes online. No transaction information will be lost; the transactions can still be invoiced.

As a CPO you can take it one step further. It is possible to upload a local authorization list to a station. This is a whitelist of all charge cards that are allowed to charge at the station. The local authorization list has version numbers and allows incremental updating to keep it in sync with the central database of the CPO. It is not part of the core profile of OCPP, however, so not all charging stations will have implemented this.

Finally, there is a configuration option *AllowOfflineTxForUnknownId* (OCPP 1.6) or *AuthCtrlr.OfflineTxForUnknownIdEnabled* (OCPP 2.0.1) that, when set to true, will allow any charge card to start charging at the station. Once the connection is restored, the management system will try to validate the charge card and transactions for known cards will be invoiced as usual. Invoicing for unknown charge cards, however, will not be possible. This is a financial risk that the CPO takes in return for a higher availability of the charging station for customers. In practice, this risk may be acceptable for level 2 AC chargers, but is often considered too high for DC fast chargers, which are much more expensive.

5.3.1. Detecting service availability while offline

Even when a station supports offline starting of transactions, and could as such be counted as being "up", there is no proof that it is working, since no message are exchanged. When the station comes back online, all offline transactions are sent to the backoffice, but failed authorization attempts or StatusNotifications are not sent, because only transaction-related messages are buffered while offline. Only if there were transactions during (most of) the offline period can you be certain that the service was available during the offline period.

For OCPP 2.0.1 an extra check is possible: the configuration variable OCPPCommCtrlr.QueueAllMessages can be set to true to instruct the station to queue all messages during an offline period. In that case any AuthorizationRequest or StatusNotificationRequests and NotifyEventRequest, that occurred in the offline period, will be sent to CSMS when connectivity is restored. This allows an operator to check for failed authorization attempts or error messages that occurred during the offline period.

5.4. Remote assistance

As long as the network connection with the charging station is up, there are number of messages that a CSO can send to try and solve issues.

Table 9. Remote support commands for OCPP 1.6

Command	Description
Reset.req hard/soft	Reset charging station
UnlockConnector.req	Unlock a stuck connector
TriggerMessage.req	Request charging station to send a message (e.g. StatusNotification)
GetDiagnostics.req	Download the diagnostics log file
UpdateFirmware.req	Update charging station firmware
ChangeAvailability.req	Change the availability state of the charging station
GetConfiguration.req	Get the value of configuration parameters
ChangeConfiguration.req	Change the value of configuration parameters

Table 10. Remote support commands for OCPP 2.0.1

Command	Description
ResetRequest Immediate/OnIdle	Reset charging station (or EVSE)
UnlockConnectorRequest	Unlock a stuck connector
TriggerMessage	Request charging station to send a message (e.g. StatusNotification)
GetTransactionStatusRequest	Check for remaining (unsend) transaction event messages
GetLogRequest	Download a diagnostics log file
UpdateFirmwareRequest	Update charging station firmware
ChangeAvailabilityRequest	Change availability state of the charging station
GetReportRequest/ GetBaseReportRequest	Request a report of charging station components and variables
GetVariablesRequest	Get the value of selected (configuration) variables
SetVariablesRequest	Set the value of selected (configuration) variables
SetVariableMonitoringRequest	Configure variables to be monitored
SetMonitoringLevelRequest	Set the monitoring level, ranging from 0 (Danger) to 9 (Debug)
SetDisplayMessageRequest	Show an informational message on the display