DIN/TS 70121

ICS 29.160.40; 43.120

Ersatz für
DIN SPEC 70121:2014-12

Elektromobilität –
Digitale Kommunikation zwischen einer Gleichstrom-Ladestation und einem Elektrofahrzeug zur Regelung der Gleichstromladung im Verbund-Ladesystem;
Text Englisch

Electromobility –
Digital communication between a d.c. EV charging station and an electric vehicle for control of d.c. charging in the Combined Charging System;
Text in English

Électromobilité –
Communication digitale entre la borne de recharge à courant continu et la véhicule éléctrique pour régler la charge à courant continu dans le système de charge combiné (CCS);
Texte en anglais

Gesamtumfang 227 Seiten

DIN-Normenausschuss Auto und Mobilität (NAAutomobil)

Contents

Figures

Tables

Foreword

This document has been prepared by Technical Committee NA 052-00-31 AA “Data communication” of the DIN Standards Committee Road Vehicle Engineering (NAAutomobil).

Technical Specifications are not part of the body of German Standards.

A Technical Specification is the result of standards work which is not yet being published as a standard by DIN because there are certain reservations regarding its content, or because it was not prepared in accordance with the normal procedures for standards work.

No draft of this Technical Specification has been published.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. DIN shall not be held responsible for identifying any or all such patent rights.

The document is prepared in English as it also used in other countries than Germany. The German definitions for the Terms mentioned in Clause 3 are given in Annex D.

For current information on this document, please go to DIN’s website (www.din.de) and search for the document number in question.

Amendments

Compared to DIN SPEC 70121:2014-12, the following changes have been made:

a) all new requirements have a number from 800 onwards;

b) the following requirements have been deleted:

   V2G-DC-546, V2G-DC-618, V2G-DC-641, V2G-DC-642, V2G-DC-626;

c) Annex D was added;

d) a new informative Annex E has been added which shows the newly modified requirements.

Previous editions

DIN SPEC 70121: 2012-08, 2014-12

Introduction

The time consumption for electrical vehicles (EVs) charging process is determined by the maximum power, which can be transferred to the EV. The alternating current (AC) charging has a fixed limited voltage (e.g., 400 V in Europe) and therefore an increasing of power supply can only be achieved by higher currents. This approach is limited by the cable weight because a higher current requires a greater cross section of the copper cable. As the user should be able to connect the EV with the EVSE the charging cable should not be heavier than a typical fuel tube of a filling Station today. A further disadvantage of the AC charging approach is that the rectifier is placed in the EV. This results in an increase of the overall EV weight and complexity of it as depending on a size of the rectifier, a special liquid cooling system may be required for it. The weight of a rectifier however increases with the power it can handle. These disadvantages can be prevented by using a DC charging approach. In this case the rectifier is placed in the EVSE and the used voltage is limited by the maximum battery voltage (typically 400-800 V), which is normally higher than the AC voltage.

The drawback of the DC charging solution is that the communication real time requirements on the EV-EVSE interface increases as the rectifier and the battery management system requires to communicate via this interface. The prevention of an incorrect charging state requires a real time communication, which is tailored to the charging process. The real time requirements depend on the battery system and are subjects of further investigations.

The main aspects of this specification are:

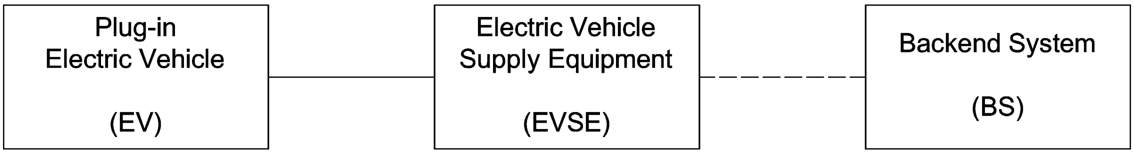
— Support of the Combined Charging System based on IEC 61851-23;

— Aligned charging process with SAE J2847/2;

— Ease of integration of SEP2.0 or similar XML based approach for V2G communications.

1 Scope

This document specifies the DC specific communication between battery electric vehicles (BEV) or plug-in hybrid electric vehicles (PHEV) and the Electric Vehicle Supply Equipment (EVSE, also known as charge spot) as shown in Figure 1. In the following, the BEV and PHEV are also named Electrical Vehicle (EV). Information exchange to other actors is not in scope of this document.



Key

__________ Communication in the scope of this technical specification

_ _ _ _ _ _ _ _ _ Communication not in the scope of this technical specification

Figure 1 — Communication relationship between EV, EVSE, and BS

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEEE 1901, IEEE Std. 1901-2010, *Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications, 2010*

IEC 61851-1, *Electric vehicle conductive charging system — Part 1: General requirements*

IEC 61851-23, *Electric vehicle conductive charging system — Part 23: d.c. electric vehicle charging station*

IEC 62196-2*, Plugs, socket-outlets, vehicle connectors and vehicle inlets — Conductive charging of electric vehicles — Part 2: Dimensional compatibility and interchangeability requirements for a.c. pin and contact-tube accessories*

IEC 62196-3, *Plugs, socket-outlets, vehicle connectors and vehicle inlets — Conductive charging of electric vehicles — Part 3: Dimensional compatibility interchangeability requirements for d.c. pin and contact-tube vehicle couplers*

ISO 17409, *Electrically propelled road vehicles — Connection to an external electric power supply — Safety requirements*

HomePlug Green PHY, *HomePlug Green PHY Specification Release Version 1.1.1, 2014*

IETF RFC 768, *User Datagram Protocol (August 1980)*

IETF RFC 793, *Transmission Control Protocol — DARPA Internet Program — Protocol Specification (September 1981)*

IETF RFC 1323, *TCP Extensions for High Performance (May 1992)*

IETF RFC 1624, *Computation of the Internet Checksum via Incremental Update (May 1994)*

IETF RFC 1981, *Path MTU Discovery for IP version 6 (August 1996)*

IETF RFC 2018, *TCP Selective Acknowledgment Options (October 1996)*

IETF RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification (December 1998)*

IETF RFC 2988, *Computing TCP's Retransmission Timer (November 2000)*

IETF RFC 3122, *Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification (June 2001)*

IETF RFC 3484, *Default Address Selection for Internet Protocol version 6 (IPv6) (February 2003)*

IETF RFC 6582, *The NewReno Modification to TCP's Fast Recovery Algorithm (April 2012)*

IETF RFC 4286, *Multicast Router Discovery (December 2005)*

IETF RFC 4291, *IP Version 6 Addressing Architecture (February 2006)*

IETF RFC 4294, *IPv6 Node Requirements (April 2006)*

IETF RFC 4429, *Optimistic Duplicate Address Detection (DAD) for IPv6*

IETF RFC 4443, *Internet Control Message Protocol (ICMP v6) for the Internet Protocol version 6 (IPv6) specification (March 2006)*

IETF RFC 4861, *Neighbour Discovery for IP version 6 (IPv6)*

IETF RFC 4862, *IPv6 Stateless Address Autoconfiguration (September 2007)*

IETF RFC 4884, *Extended ICMP to Support Multi-Part Messages (April 2007)*

IETF RFC 5095, *Deprecation of Type 0 Routing Headers in IPv6 (December 2007)*

IETF RFC 5220, *Problem Statement for Default Address Selection in Multi-Prefix Environments: Operational Issues of RFC 3484 Default Rules (July 2008)*

IETF RFC 5482, *TCP User Timeout Option (March 2009)*

IETF RFC 5681, *TCP Congestion Control (September 2009)*

IETF RFC 5722, *Handling of Overlapping IPv6 Fragments (December 2009)*

IETF RFC 5871, *IANA Allocation Guidelines for the IPv6 Routing Header (May 2010)*

IETF RFC 6298, *Computing TCP's Retransmission Timer (June 2011)*

IETF RFC 6335, *Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transfer Protocol Port Number Registry (August 2011)*

DIN SPEC 91286, *Electric mobility — Schemes of identifiers for E-Roaming — Contract ID and Electric Vehicle Supply Equipment ID*

SAE J1772, *SAE Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler*

SAE J2847/2, *Communication between Plug-in Vehicles and Off-Board DC Chargers*

W3C EXI 1.0, *Efficient XML Interchange (EXI) Format 1.0, W3C Recommendation (March 2011)*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

DIN and DKE maintain terminological databases for use in standardization at the following addresses:

— DIN-TERMinologieportal: available at https://www.din.de/go/din-term

— DKE-IEV: available at https://www.dke.de/DKE-IEV

3.1
amplitude map
transmit power-reduction factor for each subcarrier related to the tone map

3.2
central coordinator
CCo
manager of a HomePlug Green PHY network

3.3
channel access priority
CAP
method to prioritize the channel access (see HomePlug Green PHY specification)

3.4
charging session
time between the beginning (connection of the cable) and the end (disconnection of the cable) of a charging process

3.5
coexistence
ability of different HomePlug Green PHY systems to share the same physical media and to function simultaneously

3.6
communication media
physical media carrying the HomePlug Green PHY signal given by the cable assembly, which connects the charging infrastructure and the EV

3.7
communication setup timer
timer that monitors the time from plug-in until reception of the SessionSetupRes message by the EV

3.8
connection coordination
entity which provides the whole functionality for EV to EVSE matching and initialization, via the data link control SAP

Note 1 to entry: This entity also controls the relationships between the basic signalling and the upper layers.

Note 2 to entry: The entity indicates link status and error information to higher layers. The control of the low-layer communication Network Management parameters is handled over the Data Link Control SAP

3.9
crosstalk
capacitive or inductive coupling between two individual electric circuits, each providing a media for a HomePlug Green PHY network, in a way that the two networks are influenced by each other

3.10
DATA SAP
service access point that defines the interface between Layer 2 and Layer 3 for exchange of V2G related payload

3.11
electric vehicle communication controller
EVCC
entity which implements the communication to one or multiple SECCs

3.12
ETH SAP
layered access point that supports applications using Ethernet II class packets, including IEEE 802.3, with or without IEEE 802.2 (LLC), IEEE 802.1H (SNAP) extensions, and/or VLAN tagging

3.13
global address
IP address with unlimited scope

3.14
HomePlug Green PHY node
device equipped with a HomePlug Green PHY compatible modem chip

Note 1 to entry: It characterizes one logical and physical communication device that is attached to a physical media, and is capable of sending, receiving, or forwarding information over a HomePlug Green PHY communication channel.

3.15
high-level communication
HLC
digital communication used for the exchange of charging parameters

EXAMPLE Voltage and current.

Note 1 to entry: HLC starts with CM_SLAC_PARM.REQ and ends when the TCP connection is closed.

3.16
initialization
process of interaction between the EV, the EVSE and an external trigger, beginning from plug-in of the charging cable assembly until the decision for the charging mode to be applied

3.17
inter system protocol
ISP
protocol that enables various broadband power line systems to share power line communication resources in time (time domain multiplex), in frequency (frequency domain multiplex), or both

Note 1 to entry: For more information refer to IEEE 1901.

3.18
IP-address
IP-layer identifier for an interface or a set of interfaces

3.19
link-local address
IP address with link-only scope that can be used to reach neighbouring nodes attached to the same link

3.20
logical network
network defined solely on OSI-layer 2

Note 1 to entry: This is a set of HomePlug Green PHY Stations, which possess the same network key.

Note 2 to entry: Only members of the same logical network are able to exchange encrypted payload data and are visible for each other on higher layers. Different logical networks may exist on the same physical media at the same time and are typically used for network segmentation.

3.21
MAC address
unique identifier assigned to network interfaces for communication on the data link layer

3.22
matching
process to determine the HomePlug Green PHY node of the EV and the HomePlug Green PHY node of the EVSE that are physically directly connected to one another via the charging cable assembly

Note 1 to entry: The matching process starts with CM_SLAC_PARM.REQ and ends with CM_SLAC_MATCH.CNF.

3.23
message set
set of mandatory V2G messages and parameters for the EVCC or SECC covering one or multiple use case elements

3.24
message timer
timer that monitors the exchange of a pair of request/response messages

3.25
network segment
collection of devices that can exchange data on data link layer level directly via data link addresses

EXAMPLE Ethernet: All devices which can see each other via MAC addresses).

3.26
network segment size
number of devices that are handled in one network segment

3.27
node
device that is part of a communication network

EXAMPLE A device implementing IPv6 in an IP network.

3.28
performance time
non-functional timing requirement defining the time a V2G entity should not exceed when executing or processing a certain functionality

3.29
PHY
connects a link layer device to a physical medium

Note 1 to entry: Layer 1 in the OSI reference model.

3.30
pilot function controller
system that manages the control pilot line on the EVSE side

Note 1 to entry: According to IEC 61851-1.

3.31
profile
group of mandatory and optional message sets covering a set of similar charging scenarios for a specific identification means

3.32
QPSK modulation
phase modulation technique that transmits two bits in four modulation states (quadrature phase shift keying)

3.33
ready-to-charge timer
timer that monitors the time from plug-in until the first power delivery message

3.34
request-response message pair
request message transmitted between sender and receiver and its corresponding response message that is invoked by the request message in the receiver

3.35
request-response message sequence
sequence of one or more request-response message pairs

3.36
ROBO mode
communication mode, which uses QPSK only for carrier modulation within the orthogonal frequency division multiplexing (OFDM) to achieve higher robustness in transmission

Note 1 to entry: The ROBO mode can be set to three different performance levels: Mini-ROBO, Standard ROBO and High Speed ROBO.

Note 2 to entry: Refer to the HomePlug Green PHY specification for details.

3.37
SDP client
V2G entity that uses the SDP server to retrieve configuration information about the SECC being able to access the SECC

3.38
SDP server
V2G entity providing configuration information for accessing the SECC

3.39
sequence timer
timer that monitors a response-request message sequence

3.40
shared bandwidth
network capability that is used by multiple users or devices for transmitting data

Note 1 to entry: The resulting allocation is shared bandwidth.

3.41
signal coupling
circuit that describes the method of coupling the HomePlug Green PHY signal to the communication media

3.42
signal level attenuation characterization
SLAC
protocol to measure the signal strength of a signal between HomePlug Green PHY nodes (used for matching)

3.43
signal strength measurement
measurement between CM_START_ATTEN_CHAR.IND and CM_ATTEN_CHAR.RSP messages

3.44
supply equipment communication controller
SECC
entity which implements the communication to one or multiple EVCCs and which can be able to interact with secondary actors

3.45
TCP_DATA
socket/interface for data transfer based on TCP connection

3.46
timeout
specific time a V2G entity monitors the communication system for a certain event to occur

Note 1 to entry: If the specified time is exceeded, the respective V2G entity initiates the related error handling.

3.47
timer
device or piece of software used in an implementation for measuring time

Note 1 to entry: Depending on the specific use case a timer is used to trigger certain system events as well.

3.48
tone mask
set of OFDM carriers or „tones“ that can be used in a given regulatory jurisdiction or given application

3.49
V2G communication session
semi-permanent interactive information interchange between two specific V2G entities for exchanging V2G messages in a connection-oriented communication

3.50
V2G entity
primary or secondary actor participating in the V2G communication

3.51
V2G message
message exchanged on application layer

3.52
V2GTP entity
V2G entity supporting the V2G transfer protocol

4 Symbols and abbreviations

For the purposes of this document, the following abbreviations apply.

AFE	Analog Front End
BEV	Battery Electric Vehicle
CAP	Channel Access Priority
CCo	Central Coordinator
CP	Control Pilot
CPL	Control Pilot Line
CPLT	control pilot line transmission, according to IEC 61851-1 and SAE J1772
DLINK	Data Link
EV	Electric Vehicle
EVCC	Electric Vehicle Communication Controller
EVSE	Electric Vehicle Supply Equipment
	NOTE In the context of this document, the term EVSE refers to the charging equipment, external to the EV.
EXI	Efficient XML Interchange
f_{NMK} (x)	Function to derive the Network Management Key from a given string x, according to the HomePlug Green PHY specification.
HLC	High-Level Communication
HLE	Higher Layers Entities
HPGP	HomePlug Green PHY
ID	Identification
IP	Internet Protocol
ISP	Inter System Protocol
LC	Link Control (control and status of the HomePlug Green PHY link)
MAC	Media Access Control
MNBC	Multi-Network Broadcast
N/A	Not applicable
NACK	Negative Acknowledgement
NEK	Network Encryption Key
NMK	Network Management Key
NWP	Network Password
OEM	Original Equipment Manufacturer
PE	Protective Earth
PDU	Protocol Data Unit
PHEV	Plug-in Hybrid Vehicle
PLC	Power Line Communication
PnC	Plug and Charge
PSD	Power Spectrum Density
PWM	Pulse Wide Modulation
QPSK	Quadrature Phase Shift Keying
RESS	Rechargeable Electric Storage System

RF	Radio Frequency
RFC	Request for comments
SAML	Security Assertions Markup Language
SAP	Service Access Point
SDP	SECC Discovery Protocol
SDU	Service Data Unit
SE	Supply Equipment
SECC	Supply Equipment Communication Controller
SLAC	Signal Level Attenuation Characterization
SLAAC	Stateless address auto configuration
SNR	Signal to noise ratio
TCP	Transmission Control Protocol
TSO	Transmission System Operator
V2G	Vehicle-to-Grid Communication
V2G CI	Vehicle-to-Grid Communication Interface
V2GTP	V2G Transfer Protocol
V2GTPPT_EXI	V2G Transfer Protocol Payload Type for EXI messages
UDP	User Datagram Protocol
XML	Extensible Markup Language

5 Relation to other standards

The digital communication between an EV and an EVSE defined in this document is based on an early unpublished version of the ISO 15118 document series.

The assumptions on DC electric vehicle charging station in this document are based on the definitions in IEC 61851-23. The basic signalling is based on the definitions in IEC 61851-1.

6 Conventions

The conventions described in the following subclauses are applied by this document.

6.1 Definition of OSI based services

This document is based on the conventions discussed in the OSI service conventions (refer to ISO 10731) as they apply for the individual layers specified in this document.

This document describes requirements applicable to layer 1 - 7 according to the OSI layered architecture.

6.2 Requirement structure

6.2.1 Requirement format

This document uses a requirement structure i.e. a unique number identifies each individual requirement included in this clause. This requirement structure allows for easier requirement tracking and test case specification. The following formats are used:

"**[V2G-DC-*xyz*]**" requirement text where:

— "V2G-DC" represents the DIN/TS 70121;

— “*xyz*” represents the individual requirement number and;

— "requirement text" represents the actual text of the requirement.

EXAMPLE **[V2G-DC-001]**

6.2.2 Applicability

This clause specifies the requirements to be implemented by the EV and the EVSE to allow communication between both entities. Usually, the requirements of this standard are implemented by one vehicle entity and additionally at least by one EVSE entity depending on the EVSE network architecture.

The requirements of this specification will refer to as “The EV shall implement…” implying that the vehicle communication entity shall implement the required functionality if not explicitly stated otherwise.

The requirements of this specification will refer to as “The EVSE shall implement…” implying that at least one EVSE entity shall implement the required functionality if not explicitly stated otherwise.

The requirements of this specification will refer to as “The V2G entities shall implement…” implying that all V2G entities involved in the V2G communication shall implement the required functionality if not explicitly stated otherwise.

[V2G-DC-001] The EV shall implement all mandatory requirements defined in this document for an EV.

[V2G-DC-002] The EVSE shall implement all mandatory requirements defined in this document for an EVSE.

6.2.3 Usage of RFC references

When RFCs are referenced all “shall/shall not” requirements are mandatory.

[V2G-DC-003] If a referenced RFC has been updated by one or several RFC, the update is fully applicable for this standard.

[V2G-DC-004] If an update or part of an update applicable to an RFC referenced herein is not compatible with the original RFC or the implementation described by this standard the update shall not apply.

6.3 Notation used for XML schema diagrams

This document makes use of XML as a description format for V2G messages. For details with regards to the XML schema diagram notation used in this document refer to Altova XMLSpy Manual.

Allowing for an easy way to distinguish the types used for the XML schema definitions in this document, the following naming conventions apply:

— complex type use capitalized first letters;

— simple types use non capitalized first letters.

7 DC charging system architecture

7.1 System schematics of DC charging system

This document assumes a DC charging system as specified in IEC 61851-23, Annex C. For further details, please refer to IEC 61851-23.

7.2 Pilot function

This specification assumes the charging mode 4 as defined in IEC 61851-1. Mode 4 is used for DC charging. The Pilot Function is based on a PWM conforming to IEC 61851-1, Annex A. For further details, please refer to IEC 61851-1.

7.3 Communication stack

Figure 2 shows the communication stack for DC charging according to this document.

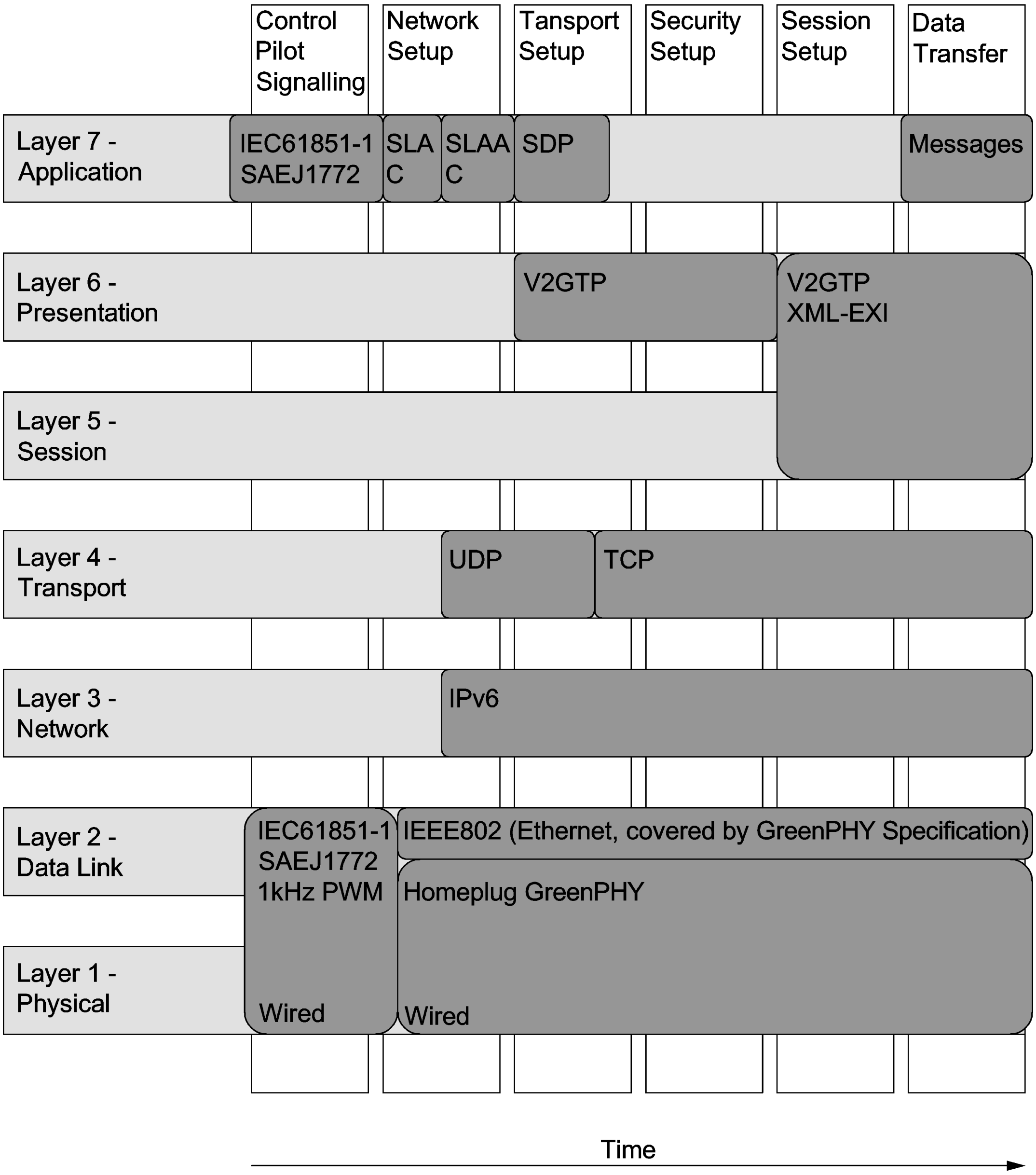


Figure 2 — Protocol stack for DC charging control without security

8 EV-EVSE communication

8.1 Basic requirements for V2G communication

8.1.1 General information and definitions

After establishing a physical connection, an IP based session shall be established. Within this session, messages are transferred between EV and EVSE. Communication is needed for controlling a battery charging session and the interaction between EV and EVSE.

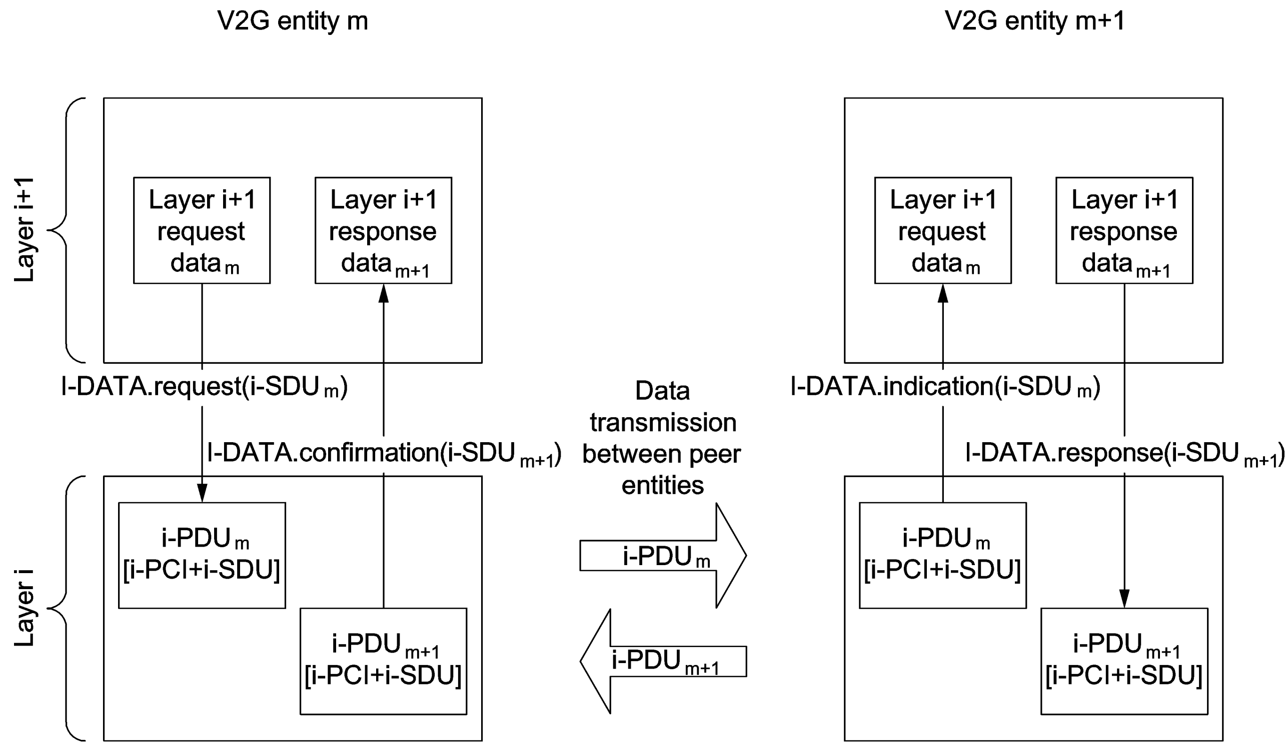
8.2 Service primitive concept of OSI layered architecture

8.2.1 Overview

This subclause explains how the OSI layered architecture applies within this document. It is intended to provide simple means for describing the interfaces between the individual communication protocol layers required by this document and furthermore allows for defining timing requirements more precisely.

Services are specified by describing the service primitives and parameters that characterize a service. This is an abstract definition of services and does not force a particular implementation.

Figure 3 depicts a simplified view of OSI layer interaction sufficient to understand the OSI layered architecture principles for the context of this document.



Key

PDUX Protocol data unit of network entity x
PCI Protocol control information
SDU_X Service data unit of network entity x

Figure 3 — OSI layered architecture principles

When a layer i+1 instance of V2G entity m exchanges data with a layer i+1 instance of V2G entity m+1 each instance uses services of an instance of layer i. A service is defined as a set of service primitives.

8.2.2 Syntax of service primitives

Service primitives are described with the following syntax:

[Initial of layer]-[NAME].[primitive type](parameter list)

whereas [initial of layer] is one out of the following seven:

[Physical, Data Link, Network, Transport, Session, Presentation, Application]

whereas [NAME] is the name of the primitive

EXAMPLE Typical examples for [Name] are CONNECT, DISCONNECT, DATA.

whereas [primitive type] is one out of the following four:

[request, indication, response, confirmation]

whereas (parameter list) includes a list of parameters separated by comma the user of the service is supposed to provide when using the respective service primitive; optional parameters are marked with brackets "[..]"

NOTE In this document, the primitive type ".indication" always indicates an event asynchronously to the upper layer.

8.3 Physical and Data Link Layer

8.3.1 System architecture

8.3.1.1 High-Level communication and basic signalling

8.3.1.1.1 Basic signalling

[V2G-DC-005] The basic signalling follows IEC 61851-1. All the timings shall be compliant with IEC 61851-1, Annex A.

During the charging process, a bidirectional signalling according to IEC 61851-1 is used to indicate EV-related information via CP states and EVSE-related information via the CP duty cycle.

[V2G-DC-006] For DC charging according to this document, Mode 4 according to IEC 6185 shall be used.

[V2G-DC-560] For DC charging according to this document, the EV shall apply CP State C or D during the energy transfer and CP State B otherwise, unless an error condition occurring at any point in time during the energy transfer requires changing to CP State B. For more details, please refer to clause 9.7.4.1.4, in particular [V2G-DC-880] and [V2G-DC-502].

[V2G-DC-561] An EVSE that supports only DC charging according to this document on a connector shall apply a CP duty cycle of 5 % on this connector after detection of CP State B, so that the high-level communication can start, and until high-level communication stops, unless an error condition requires applying a different CP duty cycle or CP voltage.

An EVSE that supports also other charging types than DC charging (according to this document) on a connector may, in accordance with the specification of those charging types, apply other CP duty cycles, including 100 %, on this connector until it sends a ChargeParameterDiscoveryRes message with ResponseCode equal to “OK”. Please refer to [V2G-DC-671].

NOTE 1 According to IEC 61851-23, if the SECC detects CP State A or an unexpected CP State B, it carries out, without any delay, an “EVSE-initiated emergency shutdown”, which includes turning off the CP oscillator, if it is turned on, and a fast reduction of the output current. For details, please refer to IEC 61851-23.

[V2G-DC-941] In case of any error detection CP State B signalization shall be done without any delay by the EV max. within 4.5s.

NOTE 2 This max. time applies for example to [V2G-DC-871].

[V2G-DC-942] In case of any error detection turning off the CP oscillator shall be done without any delay by the EVSE max. within 4.5s

NOTE 3 This max. time applies for example to [V2G-DC-962].

8.3.1.1.2 High-Level Communication (HLC)

High-level communication in DC charging is used for exchange of charging parameters e.g. voltage and current.

[V2G-DC-008] The HLC shall be used in addition to the basic signalling in order to enable a bidirectional communication for charging control.

[V2G-DC-485] This data link is performed via the Power Line Communication technology HomePlug Green PHY according to the HomePlug Green PHY specification.

8.3.1.2 Identification requirements

Identification (e.g. payment option) is out of scope of this document.

8.3.1.3 System requirements

8.3.1.3.1 Overview

This clause describes startup of EV and EVSE after the plug-in of the charging cable.

This clause also defines how the EVSE and the EV shall be designed for properly triggering the process of matching and initialization.

8.3.1.3.2 EVSE architecture

8.3.1.3.2.1 Control pilot and HomePlug Green PHY communication requirements

[V2G-DC-471] Each EVSE connector shall have its own dedicated pilot function controller.

[V2G-DC-562] Each EVSE connector shall have its own dedicated HomePlug Green PHY node.

The MAC matching is done between the HomePlug Green PHY node locally assigned to each connector of the EVSE and the HomePlug Green PHY node on the EV side.

NOTE 1 The present document only describes the point to point relationship between the HomePlug Green PHY nodes on EV and EVSE sides.

NOTE 2 Only “IEC 61851-1 Mode 4” EVSEs, that include an HLC module, are in the scope of this document.

8.3.1.4 Configuration of a HomePlug Green PHY node

This section is a summary of some key points of the HomePlug Green PHY technology according to the HomePlug Green PHY specification which this standard requires.

The Physical Layer of the HomePlug Green PHY technology is fully described in the HomePlug Green PHY specification.

[V2G-DC-018] The HomePlug Green PHY node on the EV side shall be configured to never become the CCo.

NOTE 1 This could be done with the APCM_SET_CCo.REQ primitive as defined in the HomePlug Green PHY specification.

[V2G-DC-019] The HomePlug Green PHY node on the EVSE side shall be configured to always be the CCo.

NOTE 2 This could be done with the APCM_SET_CCo.REQ primitive as defined in the HomePlug Green PHY specification.

8.3.1.4.1 ROBO modes and data rate

The HomePlug Green PHY Technology uses three robust (“ROBO”) modes of communication for:

— Beacon, data broadcast, and multicast communication,

— Session setup,

— Exchange of management messages,

— Data messages.

All ROBO modes use QPSK modulation. This mechanism introduces redundancy by a factor that depends on the type of ROBO mode.

Table 1 — ROBO modes and PHY data rate

ROBO mode	Number of copies	PHY rate *(Default tone mask)*
MINI-ROBO_AV	5	3.7716 Mbps
STD-ROBO_AV	4	4.9226 Mbps
HS-ROBO_AV	2	9.8452 Mbps

[V2G-DC-994] The HomePlug Green PHY node on the EV side shall support MINI-, STD- and HS ROBO mode as specified in Table 1.

[V2G-DC-995] The HomePlug Green PHY node on the EVSE side shall support MINI-, STD- and HS ROBO mode as specified in Table 1.

8.3.1.4.2 Shared bandwidth mechanisms

Coexistence mechanisms with other HomePlug technologies are addressed in the specifications HomePlug Green PHY and IEEE 1901.

8.3.1.4.2.1 Inter System Protocol (ISP)

HomePlug Green PHY implements the ISP as a coexistence mechanism. The ISP, defined by IEEE 1901, allows coexistence between devices that implement non-interoperable protocols. Using ISP, 1901 access, 1901 wavelet, 1901 FFT, LRWBS and ITU-T G.hn devices can coexist.

[V2G-DC-021] The HomePlug Green PHY node on the EVSE side shall be capable to detect the zero cross of the AC line cycle to support coexistence functionality. Zero cross detection is optional for the HomePlug Green PHY node on the EV side.

8.3.1.4.2.2 Coexistence with HomePlug AV technologies

In addition to the coexistence mechanisms with other HomePlug technologies, there is a specificity regarding the HomePlug AV technology.

In some cases, with a fully loaded HomePlug AV network, the HomePlug Green PHY network may be limited to 7 % in time. This might affect the average data rate.

8.3.2 Connection coordination

8.3.2.1 Plug-in phase

This phase covers the period of time after plug-in of the charge cable assembly.

[V2G-DC-507] After the EVSE detects CP State B, the EVSE's HomePlug Green PHY node shall be ready for communication in less than T_conn_max_comm.

[V2G-DC-807] An EVSE that supports only DC charging according to this document on a connector shall apply a CP duty cycle of 5 % on this connector not before its HomePlug Green PHY node becomes ready for communication (able to receive first SLAC message) and no later than 1 s after its HomePlug Green PHY node is ready for communication.

[V2G-DC-733] An EVSE that supports only DC charging according to this document on a connector shall apply a CP duty cycle of 100 % (with Vg=12 V constant voltage, according to IEC 61851-1) on this connector if its HomePlug Green PHY node is not ready for communication, unless an error condition requires applying a different CP duty cycle or CP voltage.

8.3.3 EV – EVSE matching process

8.3.3.1 Overview

Crosstalk of HomePlug Green PHY signals on several charge cables could lead to an incorrect pairing of an EVCC and a SECC. For a successful charging process, a correct pairing between the EV and the connector of the EVSE plugged into this EV is indispensable. This pairing is achieved by the matching process. After the matching process in layer 1 and 2, upper layers can proceed with their binding process. If an EVCC or a SECC is not member of a HomePlug Green PHY logical network, it is in "unmatched" state. When the connector is being plugged into the EV's inlet, the EVCC and SECC are initially in the "unmatched" state. In the following subclause, the term "matching" refers to "Layer 2 matching".

NOTE This standard covers a point-to-point matching.

Despite matching is mandatory at layer 2, it is also mandatory for higher layers.

8.3.3.2 Initialization of the matching process

The first phase of the matching process is called Initialization of the matching process. During this phase, the node is configured to enhance the matching process.

[V2G-DC-014] Each HomePlug Green PHY node shall provide a method to exchange authorized frequencies to be used for the communication to be in line with frequency restrictions. A HomePlug Green PHY node shall respect the spectrum limitation sent by the counterpart node.

All EVSEs should be able to update the set of frequencies to be used according to future legislative regulations.

[V2G-DC-996] The HomePlug Green PHY node on the EV side shall support channel access priority 1, 2 and 3.

[V2G-DC-997] The HomePlug Green PHY node on the EVSE side shall support channel access priority 1, 2 and 3.

[V2G-DC-020] Any SLAC related message transmitted shall be sent with CAP3 priority to speed up the matching process.

[V2G-DC-821] A SLAC request shall only be responded by an EVSE‘s HomePlug Green PHY node if all of the following conditions are fulfilled:

— The EVSE is connected to an EV, detected by the CP state

— The HomePlug Green PHY node of the EVSE is in “unmatched” state.

8.3.3.3 Discovery of the connected HomePlug Green PHY node on EVSE side

During the discovery process, the EV determines which EVSE is directly connected to its charge cord.

The method is based on a measurement of the HomePlug Green PHY signal strength. The EV will discover all the EVSE HomePlug Green PHY nodes in range. Based on a signal strength measurement for each HomePlug Green PHY node, the EV decides to which node it is directly connected.

[V2G-DC-029] The HomePlug Green PHY nodes on EVSE and EV side shall be compliant with the unsecured SLAC protocol defined in the HomePlug Green PHY specification.

[V2G-DC-030] Measurement of the signal strength on EVSE side shall be done according to the HomePlug Green PHY signal level attenuation characterisation (SLAC) (see also calculation example in Annex C).

[V2G-DC-031] Only the EV shall send SLAC requests

[V2G-DC-032] Only the EVSE shall answer to SLAC requests

In case of two or more EVs plug-in at the same time and there is crosstalk the EVSEs answer all SLAC requests, not only the first one.

[V2G-DC-568] The SECC shall be able to process and answer 5 SLAC requests in parallel.

[V2G-DC-034] The SLAC method shall be concluded by:

— EVSE_FOUND,

— EVSE_POTENTIALLY_FOUND, or

— EVSE_NOT_FOUND.

NOTE 1 The use of EVSE_POTENTIALLY_FOUND is optional.

[V2G-DC-819] Each EV HomePlug Green PHY node shall consider the status “EVSE_POTENTIALLY_FOUND” as “EVSE_FOUND” or as “EVSE_NOT_FOUND”. For this decision the EV can use the optional validation process as defined in 8.3.3.4.

[V2G-DC-569] The ATTEN_PROFILE within the message CM_ATTEN_CHAR.IND shall be calculated by arithmetic mean of the ATTEN_PROFILE of the CM_ATTEN_PROFILE.IND of previous M-SOUNDS corrected by the attenuation of the receive path AttnRxEVSE as described in 8.3.7.4.1.

[V2G-DC-570] The Average_Attenuation used for the matching decision shall be calculated by the arithmetic mean of all groups in the CM_ATTEN_CHAR.IND.

[V2G-DC-038] According to the result of the SLAC process, the HomePlug Green PHY node on the EV side shall send its decision using the respective messages:

— If EV_Discovering_Status = EVSE_FOUND, the EV HomePlug Green PHY node shall inform the EVSE that they are connected and shall start to set up the logical network. The EV HomePlug Green PHY node shall use the CM_SLAC_MATCH message.

— If EV_Discovering_Status = EVSE_POTENTIALLY_FOUND, the EV HomePlug Green PHY node shall inform the most probable EVSE that it will start a Validation process by using CM_VALIDATE messages

— If EV_Discovering_Status = EVSE_NOT_FOUND, the HomePlug Green PHY node shall inform HLE that matching is not possible using the primitive D-LINK_READY.indication(DLINKSTATUS = no Link) and keep the HomePlug Green PHY node in the “unmatched” state.

NOTE 2 Attenuation measurement is based on the PSD values of the North American carrier and spectral mask of the HomePlug Green PHY specification. − 50 dBm/Hz will give 0 dB attenuation during SLAC measurement.

As soon as the HomePlug Green PHY node on the EV side is active, it may start to discover its connected EVSE.

During this process, the EV shall send broadcast requests only. The EVSE measures the attenuation of this signal for different groups of frequencies, and answers to the EV. When the EV receives the results, it shall decide whether the received message was sent by a physically connected EVSE.

In case of doubt, the HomePlug Green PHY node on EV side may re-launch a SLAC process and/or ask the EVSE for a Validation step.

Figure 4 gives an overview of the discovery process from EV side. It shows the sequence to follow, from the discovery of the other HomePlug Green PHY nodes to the start of the communication.

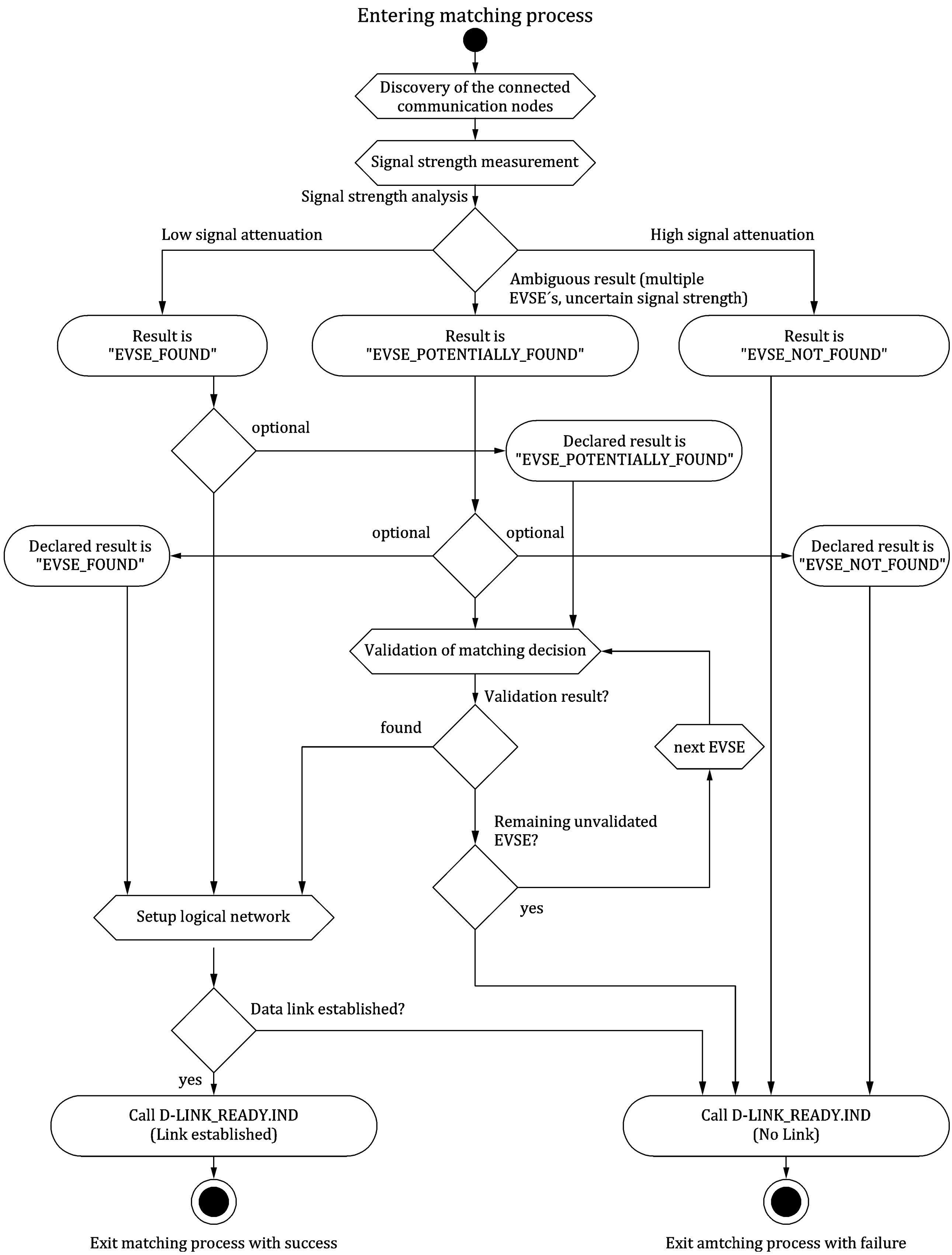


Figure 4 — EVSE discovery process from EV side

Figure 5 shows a high-level overview of the matching process. The default state of a HomePlug Green PHY node at plug-in is "unmatched". A CP state change from CP state A, B with 100 % CP duty cycle, E or F to B with 5 % CP duty cycle triggers the matching process (state "matching" in Figure 5) which determines the correct EV-EVSE pairing by a signal strength measurement. The matching process can also start at different times (not shown in Figure 5). A successful matching process leads to the state "matched".

[V2G-DC-751] Any transition from the state “matching” to state “matched” or “unmatched” shall trigger a call of the primitive D-LINK_READY.indication to indicate a successful or failed data-link setup.

[V2G-DC-752] A failure during the matching process, a loss of the control pilot or a request from HLE shall lead to a reset to state "unmatched".

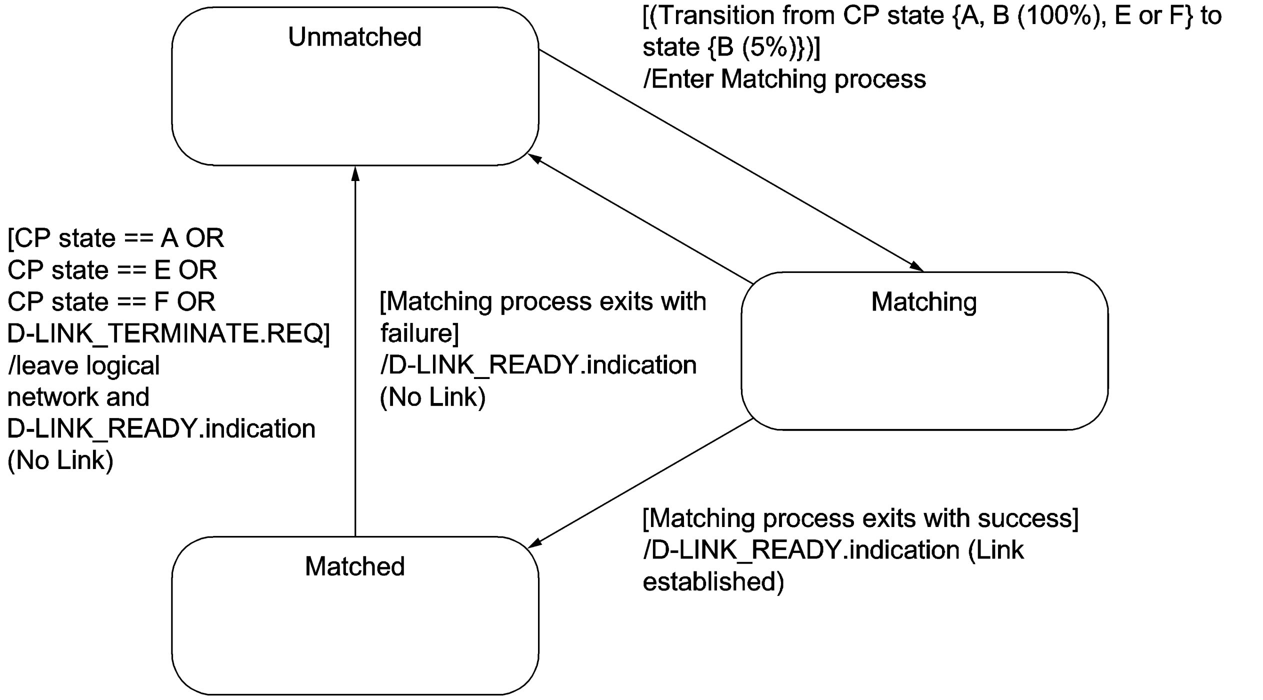


Figure 5 — Matching state machine

8.3.3.3.1 Matching process sequence diagram with SLAC messages

[V2G-DC-571] The matching process is based on the messages defined in the HomePlug Green PHY specification.

Figure 6 gives the complete sequence chart of the matching process. It shows the sequence to follow, from the discovery of the other HomePlug Green PHY nodes to the start of the communication.

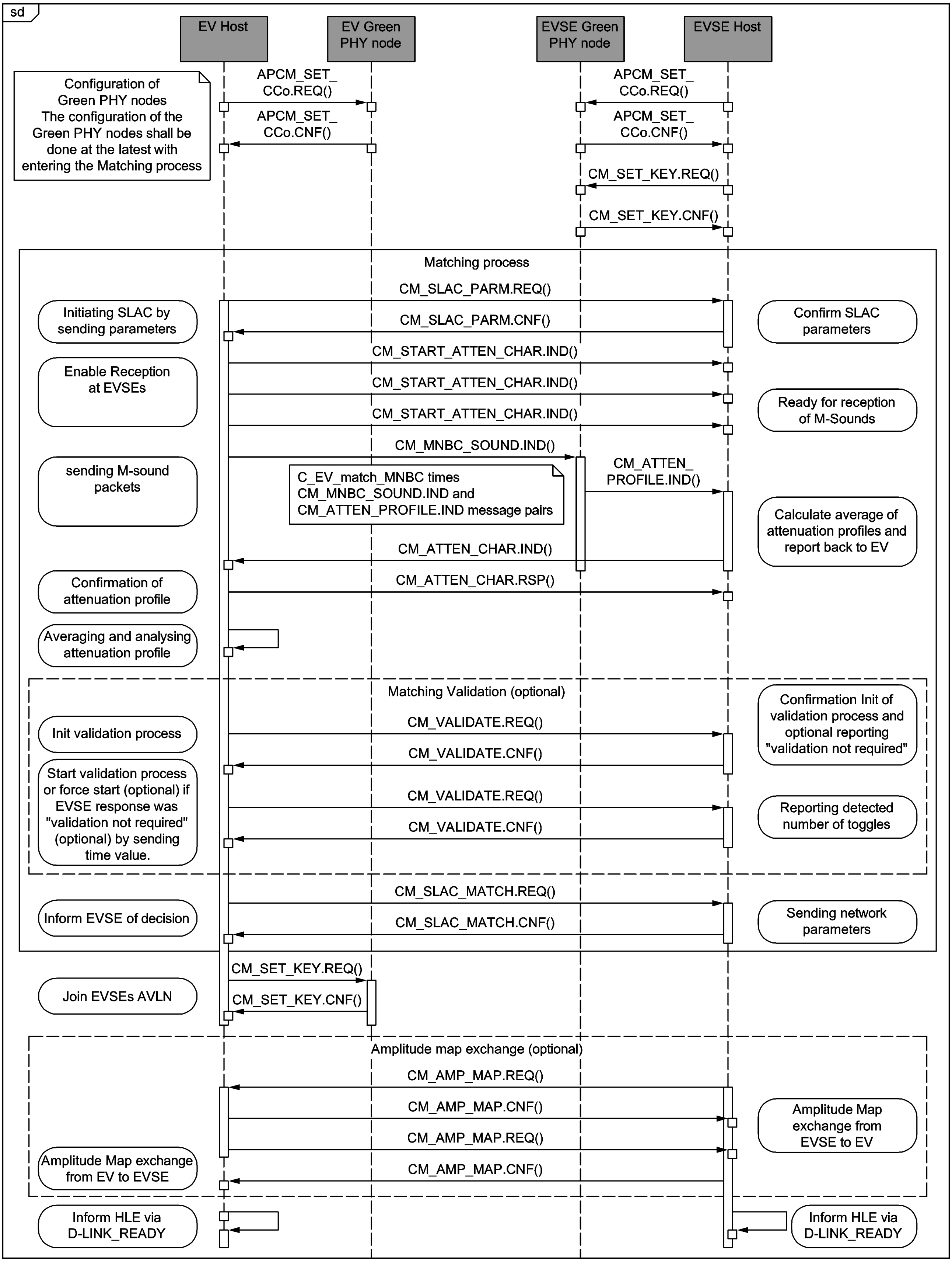


Figure 6 — Sequence chart of HomePlug Green PHY matching process

8.3.3.3.2 MME content for SLAC

HomePlug Green PHY provides management messages for data exchange between the central coordinator (EVSE) and Stations (EV).

[V2G-DC-042] For the SLAC process, the HomePlug Green PHY nodes shall use the following set of MMEs of the HomePlug Green PHY specification. The parameters to be used within the MMEs are defined in Table 2.

[V2G-DC-572] The Ethernet destination MAC address field shall be filled either as broadcast or unicast, as defined in Table 2.

NOTE 1 All SLAC messages are sent in multi-network broadcast (MNBC) even if the destination MAC address is unicast. See Annex B for details.

[V2G-DC-820] Little endian byte ordering shall be used for SLAC messages.

NOTE 2 This is specified in HomePlug Green PHY specification.

[V2G-DC-573] EVCC and SECC shall check the destination MAC address for received SLAC messages. Messages that are defined as unicast according to Table 2, Table 4, Table 5, Table 6 and Table 7 shall only be accepted if the destination MAC address equals the entities address. Messages that are defined as broadcast according to Table 2, Table 4, Table 5, Table 6 and Table 7 shall also be accepted as unicast messages.

[V2G-DC-574] The EVSE shall generate a random NMK after power-on and after every plug-out of an EV, and it shall use this random NMK for setting up a logical network.

NMK and NID should be generated after plug-out instead of plug-in to save time for data link setup.

[V2G-DC-575] The EVSE shall generate the NID out of the NMK as defined in the HomePlug Green PHY specification and shall use it for setting up a logical network.

[V2G-DC-734] For setting up a logical network the message CM_SET_KEY.REQ shall be used with the generated NMK and NID.

[V2G-DC-735] On EVSE side, the message CM_SET_KEY shall be sent after power on and after every plug out.

[V2G-DC-576] For NID generation, security level=0b00 shall be used.

Table 2 — SLAC MME parameter values

MME	Field	Octet No.	Field Size (Octets)	Value	Definition
CM_MNBC_SOUND.IND Ethernet Broadcast	APPLICATION_TYPE	0	1	0x00	EV-EVSE matching
	SECURITY_TYPE	1	1	0x00	No security
	MSVarField	—	—	—	M-Sound Variable Field See definition below

Table 2 (*continued*)

MME	Field	Octet No.	Field Size (Octets)	Value	Definition
MSVarField of CM_MNBC_SOUND.IND	SenderID	0-16	17	Filled with 0x00	
	Cnt	17	1	Variable	Countdown counter for number of Sounds remaining
	RunID	18-25	8	—	Random identifier created by the EV which initiates the SLAC process. This value shall be the same as the one sent in the CM_SLAC_PARM.REQ message by this sender.
	RSVD	26-33	8	Filled with 0x00	—
	Rnd	34-49	16	Variable	Random value
CM_SLAC_PARM.REQ Ethernet Broadcast	APPLICATION_TYPE	0	1	0x00	EV-EVSE matching
	SECURITY_TYPE	1	1	0x00	No security
	RunID	2-9	8	Variable	Identifier for a matching run, randomly chosen by the EV for each CM_SLAC_PARM.REQ message and constant for all following messages of the same run
CM_SLAC_PARM.CNF Ethernet Unicast	M-SOUND_TARGET	0-5	6	0xFFFFFFFFFFFF	Request sent in broadcast
	NUM_SOUNDS	6	1	0x0A	Number of M-Sounds to be transmitted by the EV GP station during the SLAC process
	Time_Out	7	1	0x06	Timeout for transmission of M-Sound MPDUs in multiple of 100 ms 0x00 = 0 ms 0x01 = 100 ms 0x02 = 200 ms…
	RESP_TYPE	8	1	0x01	Send to another GP STA (EV)
	FORWARDING_STA	9-14	6	MAC address of the EV HLE	Station which the measurement results shall be sent to
	APPLICATION_TYPE	15	1	0x00	Application Type 0x00: PEV-EVSE Association 0x01-0xFF: Reserved

Table 2 (*continued*)

MME	Field	Octet No.	Field Size (Octets)	Value	Definition
CM_SLAC_PARM.CNF Ethernet Unicast	SECURITY_TYPE	16	1	0x00	Security in M-Sound Messages 0x00: No Security 0x01: Public Key Signature 0x02-0xFF: Reserved
	RunID	17-24	8	variable	This value shall be the same as the one sent in the CM SLAC_PARM.REQ message by the EV
CM_START_ATTEN_CHAR.IND Ethernet Broadcast	APPLICATION_TYPE	0	1	0x00	EV-EVSE matching
	SECURITY_TYPE	1	1	0x00	No Security
	ACVarField	—	—	—	Attenuation characteristics variable fields See definition below
ACVarField of CM_START_ATTEN_CHAR.IND	NUM_SOUNDS	0	1	0x0A	Number of M-Sounds transmitted by the GP Station during the SLAC process
	Time_Out	1	1	0x06	Timeout for transmission of M-Sound MPDUs in multiple of 100 ms 0x00 = 0 ms 0x01 = 100 ms 0x02 = 200 ms…
	RESP_TYPE	2	1	0x01	Send to another GP STA (EV)
	FORWARDING_STA	3-8	6	MAC address of the EV HLE	Station which the measurement results shall be sent to
	RunID	9-16	8	variable	Random identifier created by the EV which initiates the SLAC process
CM_ATTEN_PROFILE.IND	EV MAC	0-5	6	variable	EV MAC address
	NumGroups	6	1	0x3A	Number of groups (=N) 0x00=0 Octets 0x01=1 Octet …
	RSVD	7	1	—	Reserved
	AAG[1]	8	1	variable	Average Attenuation of Group -1 0x00=0dB 0x01=1dB …

Table 2 (*continued*)

MME	Field	Octet No.	Field Size (Octets)	Value	Definition
CM_ATTEN_PROFILE.IND	...				
	AAG[N]	—	1	variable	Average attenuation of Group – N 0x00=0 Octets 0x01=1 Octet ...
CM_ATTEN_CHAR.IND Ethernet Unicast	APPLICATION_TYPE	0	1	0x00	EV-EVSE matching
	SECURITY_TYPE	1	1	0x00	No security
	ACVarField	—	—	—	Attenuation characteristics variable fields See definition below
ACVarField of CM_ATTEN_CHAR.IND	SOURCE_ADDRESS	0-5	6	variable	MAC address of the EV which initiates the SLAC process
	RunID	6-13	8	variable	Random identifier created by the EV which initiates the SLAC process
	SOURCE_ID	14-30	17	0x00	—
	RESP_ID	31-47	17	0x00	—
	NumSounds	48	1	variable	Number of M-Sounds used to generate the ATTEN_PROFILE
	ATTEN_PROFILE	—	Var.	—	Signal level attenuation see definition below
ATTEN_PROFILE of CM_ATTEN_CHAR.IND	NumGroups	0	1	0x3A	Number of groups (=N) 0x00 = 0 Octets 0x01 = 1 Octet and so on
	AAG[1]	—	1	Variable	Average attenuation of group – 1 0x00 = 0 dB 0x01 = 1 dB 0x02 = 2 dB and so on
	...				
ATTEN_PROFILE of CM_ATTEN_CHAR.IND	AAG[N]	—	1	Variable	Average attenuation of group – N 0x00 = 0 dB 0x01 = 1 dB 0x02 = 2 dB and so on

Table 2 (*continued*)

MME	Field	Octet No.	Field Size (Octets)	Value	Definition
CM_ATTEN_CHAR.RSP Ethernet Unicast	APPLICATION_TYPE	0	1	0x00	EV-EVSE matching
	SECURITY_TYPE	1	1	0x00	No security
	ACVarField	—	—	—	Attenuation characteristics variable fields See definition below
ACVarField of CM_ATTEN_CHAR.RSP	SOURCE_ADDRESS	0-5	6	6 Bytes	MAC address of the EV which initiates the SLAC process
	RunID	6-13	8	8 Bytes	Random identifier created by the EV which initiates the SLAC process
	SOURCE_ID	14-30	17	0x00	—
	RESP_ID	31-47	17	0x00	—
	Result	48	1	0x00	Success
CM_SLAC_MATCH.REQ Ethernet Unicast	APPLICATION_TYPE	0	1	0x00	EV-EVSE matching
	SECURITY_TYPE	1	1	0x00	No security
	MVFLength	2-3	2	0x3E	MatchVarField length
	MatchVarField	—	—	—	Field described below
MatchVarField of CM_SLAC_MATCH.REQ	EV ID	0-16	17	0x00	—
	EV MAC	17-22	6	—	EV MAC address
	EVSE ID	23-39	17	0x00	—
	EVSE MAC	40-45	6	—	EVSE MAC address
	RunID	46-53	8	—	Identifier given in the CM_START_ATTEN_CHAR.IND message
	RSVD	54-61	8	—	—
CM_SLAC_MATCH.CNF Ethernet Unicast	APPLICATION_TYPE	0	1	0x00	EV-EVSE matching
	SECURITY_TYPE	1	1	0x00	No security
CM_SLAC_MATCH.CNF Ethernet Unicast	MVFLength	2-3	2	0x56	MatchVarField Length
	MatchVarField	—	—	—	This field is described below

Table 2 (*continued*)

MME	Field	Octet No.	Field Size (Octets)	Value	Definition
MatchVarField of CM_SLAC_MATCH.CNF	EV ID	0-16	17	0x00	—
	EV MAC	17-22	6	—	EV MAC address
	EVSE ID	23-39	17	0x00	—
	EVSE MAC	40-45	6	—	EVSE MAC address
	RunID	46-53	8	—	Random identifier created by the EV which initiates the SLAC process. This value shall be the same as the one sent in the CM_SLAC_PRAM.REQ message by this sender.
	RSVD	54-61	8	0x00	Reserved
	NID	62-68	7	NID calculated from the random NMK that will be set	Network ID given by the CCo (EVSE)
	RSVD	69	1	—	—
	NMK	70-85	16	Random value	Private NMK of the EVSE (random value)
CM_SET_KEY.REQ Unicast to local HomePlug Green PHY Node	Key Type	0	1	0x01	Fixed value to indicate “NMK”
	My Nonce	1-4	4	0x00000000	Fixed value, encrypted payload not used
	Your Nonce	5-8	4	0x00000000	Fixed value, encrypted payload not used
	PID	9	1	0x04	Fixed value to indicate “HLE protocol”
	PRN	10-11	2	0x0000	Fixed value, encrypted payload not used
	PMN	12	1	0x00	Fixed value, encrypted payload not used
	CCo Capability	13	1	variable	CCo Capability according to the Station role.
	NID	14-20	7	variable	54 LSBs contain the NID 2 MSBs = 0b00
	NewEKS	21	1	0x01	Fixed value to indicate “NMK”
	NewKey	22-37	16	variable	NMK

[V2G-DC-577] Based on the signal attenuation read by the "CM_ATTEN_CHAR.IND" message, the EV_Discovering_Status is defined by Table 3.

Table 3 — EV_Discovering_Status definition

Status	Average_Attenuation		Description
	From	To	
EVSE_FOUND	—	C_EV_match_signal attn_direct	The EVSE is identified without any doubt
EVSE_POTENTIALLY _FOUND	C_EV_match_signal attn_direct	C_EV_match_signal attn_indirect	One or several EVSEs are identified. The next step of the matching process will allow deciding if the most probable candidate is the connected EVSE.
EVSE_NOT_FOUND	C_EV_match_signal attn_indirect	—	No direct connected EVSE is found

NOTE 3 Attenuation measurement is based on the PSD values of the North American carrier and spectral mask of the HomePlug Green PHY specification. -50dBm/Hz will give 0dB attenuation during SLAC measurement.

8.3.3.4 Validation of the matching EV-EVSE

The validation of the matching decision is a method to validate the results of the signal strength measurement via an additional independent path, based on the hardwired control pilot line.

After the signal strength measurement and based on the results thereof, the EV may decide to request from the EVSE to perform an additional Validation. As soon as the EVSE is ready to detect the CPL, the EV sends a random timer value to the EVSE (TP_EV_vald_toggle). After sending the message, the EV switches the CP state in the sequence B-C-B. This switching sequence is called BCB-Toggle in the following.

Depending on the EVSE architecture, the EVSE can answer to the request by a "not required" state. This is just a recommendation from the EVSE, the EV decides either to follow the EVSE recommendation and to skip the Validation or to continue the process and to perform the validation process.

During the time TP_EV_vald_toggle, the EV performs multiple B-C-B toggles. The number of toggles (C_vald_nb_toggles) is chosen randomly in the available time.

At the end of the timer, the EVSE sends a HomePlug Green PHY message with the number of toggles seen on the line and waits for the decision value.

If the result of the SLAC process is “EVSE_POTENTIALLY_FOUND”, a Validation step via the CPLT is optional.

This step allows the use of the CP states to assure the physical link between EV and EVSE.

[V2G-DC-043] The EV may support validation of the matching decision according to the signal sequence in Figure 7.

[V2G-DC-800] The EVSE shall support validation of the matching decision according to the signal sequence in Figure 7.

[V2G-DC-804] Legacy EVSEs that do not support validation of matching shall respond with 0x03=Failure in CM_VALIDATE.CNF message.

[V2G-DC-056] The EV may perform a “BCB-Toggle” as defined in Figure 7.

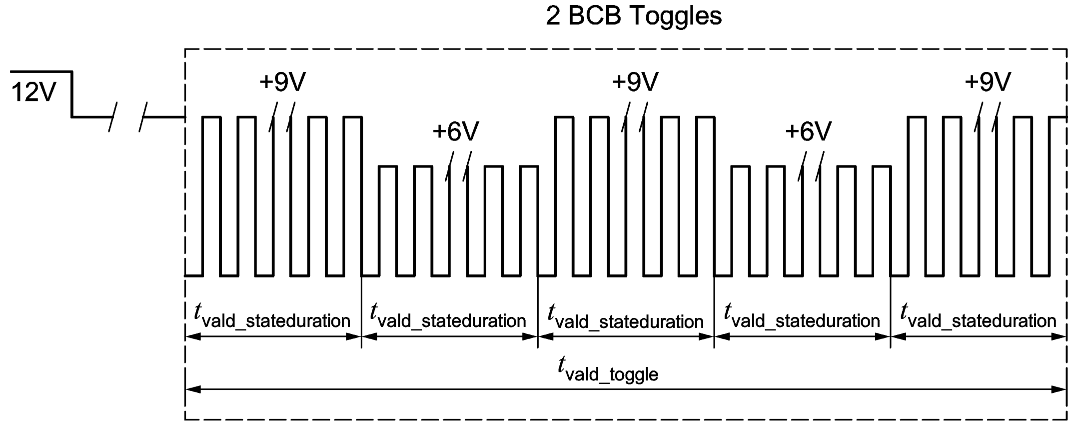


Figure 7 — Example of toggle sequence with 2 toggles

[V2G-DC-051] If the EV_Discovering_Status is "EVSE_POTENTIALLY_FOUND", the EV may start the validation process. Depending on the grid architecture, the EVSE can inform the EV that it is allowed to abort the validation step. The EV may decide to abort this step.

[V2G-DC-801] If the EVSE supports Validation of the matching decision, it shall be able to detect a variation of the CP State within T_vald_detect_time.

[V2G-DC-520] The duration of each CP State during the BCB toggle sequence is defined by T_vald_state_duration.

[V2G-DC-802] If the EVSE supports validation of the matching decision, it shall support a set of messages to exchange the number of BCB toggles, using the messages CM_VALIDATE.REQ/CNF.

[V2G-DC-737] If the EV implements validation of the matching, the EV shall support a set of messages to exchange the number of BCB toggles, using the messages CM_VALIDATE.REQ/CNF.

[V2G-DC-803] In case an EVSE receives a validation request while it is still in a Validation step with another EV or does not support validation, it shall give a negative response to this incoming request.

[V2G-DC-058] If the EV uses Validation of the matching, the EV shall perform C_vald_nb_toggles BCB toggle(s) during the validation step and shall adjust the TP_EV_vald_toggle according to the number of BCB toggles.

[V2G-DC-956] After a charge session, with the cable still connected, the EVSE shall be in B1 (100 % PWM signal) for at least 3 s, before it triggers the wake-up procedure.

[V2G-DC-521] The CP duty cycle shall have no impact on the validation step, neither on EVSE side nor on EV side.

[V2G-DC-049] Since the HomePlug Green PHY node on EV side concludes the signal strength measurement, the HomePlug Green PHY node on EV side shall also conclude validation of the matching, if it is used. The validation shall be concluded by:

— EVSE_FOUND, or

— EVSE_NOT_FOUND.

NOTE At this stage, the validation is physically performed with the CPL. The "EVSE_POTENTIALLY_FOUND" status is not possible.

[V2G-DC-523] According to the EVSE architecture, the EVSE can decide to answer to a CM_VALIDATE.REQ with "Not_Required" in CM_VALIDATE.CNF result field. In this case, the EV shall decide:

— either to continue the validation process by resending the request CM_VALIDATE.REQ or

— to skip the validation process, by informing the EVSE of the decision using the CM_SLAC_MATCH.REQ.

[V2G-DC-524] Both EV and EVSE shall comply with the sequence diagram defined in Figure 6.

[V2G-DC-525] If the EV uses validation of the matching, both EV and EVSE shall use the CM_VALIDATE messages to exchange the toggle related values, using the parameters defined in the Table 4.

[V2G-DC-580] If the EV uses validation of the matching, the Ethernet destination MAC address field shall be filled as unicast as defined in the Table 4.

Table 4 — MME parameters for Validation, first request-response

MME	Field	Octet No.	Field Size (Octets)	Value	Definition
CM_VALIDATE.REQ Ethernet unicast	Signal Type	0	1	0x00	Fixed value to indicate “EV S2 toggles on control pilot line”
	VRVarField	—	—	—	Validate request variable field. See definition below.
VRVarField of CM_VALIDATE.REQ	Timer	0	1	0x00	Fixed value. In the first VALIDATE request-response exchange the timer field shall be set to zero.
	Result	1	1	0x01	Fixed value. In the first VALIDATE request-response exchange, the result field shall be set to 0x01 = “Ready”.
CM_VALIDATE.CNF Ethernet unicast	SignalType	0	1	0x00	Fixed value to indicate “EV S2 toggles on control pilot line”
	VCVarFiled	—	—	—	Validate confirm variable field
VCVarField of CM_VALIDATE.CNF	ToggleNum	0	1	0x00	Fixed value. In the first VALIDATE request-response exchange the ToggleNum field shall be set to zero.
	Result	1	1	Var.	Result code: 0x00 = Not ready 0x01 = Ready 0x02 = Success 0x03 = Failure 0x04 = Not required 0x05-0xFF = Reserved

[V2G-DC-1001] For the first VALIDATE request-response exchange, the EV shall set the result field of the CM_VALIDATE.REQ with 0x01, to indicate that a validation process starts.

[V2G-DC-1002] For the first VALIDATE request-response exchange, the EVSE shall set the result field of the CM_VALIDATE.CNF to:

— “Not ready” to indicate that it is busy and temporarily not able to perform a validation (an immediate retry with the step 1 CM_VALIDATE.REQ by the EV is reasonable),

— “Ready” to confirm the readiness for a validation process,

— “Success” shall not be used,

— “Failure” to indicate that it does not support the validation procedure,

— “Not required” to indicate that a validation is not required.

Table 5 — MME parameters for validation, second request-response

MME	Field	Octet No.	Field Size (Octets)	Value	Definition
CM_VALIDATE.REQ Ethernet unicast	Signal Type	0	1	0x00	Fixed value to indicate “EV S2 toggles on control pilot line”
	VRVarField	—	—	—	Validate request variable field. See definition below.
VRVarField of CM_VALIDATE.REQ	Timer	0	1	Var.	Variable value indicating the time duration while the EVSE shall listen to BCB-toggles (TT_EVSE_vald_toggle): 0x00 = 100 ms 0x01 = 200 ms
	Result	1	1	0x01	Fixed value In the second VALIDATE request-response exchange, the result field shall be set to 0x01 = “Ready”.
CM_VALIDATE.CNF Ethernet unicast	SignalType	0	1	0x00	Fixed value to indicate “EV S2 toggles on control pilot line”
	VCVarFiled	—	—	—	Validate confirm variable field
VCVarField of CM_VALIDATE.CNF	ToggleNum	0	1	Var.	This value contains the number of BC-edges detected by the EVSE during TT_EVSE_vald_toggle
	Result	1	1	Var.	Result code: 0x00 = Not Ready 0x01 = Ready 0x02 = Success 0x03 = Failure 0x04 = Not Required 0x05-0xFF = Reserved

[V2G-DC-928] For the second VALIDATE request-response exchange, the EV shall set the result field of the CM_VALIDATE.REQ with 0x01, to indicate that a toggle process starts.

[V2G-DC-929] For the second VALIDATE request-response exchange, the EVSE shall set the result field of the CM_VALIDATE.CNF to:

— “Success” to confirm that the toggle process is done with success and the result field contains valid data;

— “Failure” to indicate that the toggle process could not be finished with success and the result field contains no valid data.

8.3.3.5 Set-up a logical network

After the correct pairing between EV and EVSE was determined by the matching process, the EV joins the logical network of the EVSE. Therefore, the broadcast domain is reduced to the HomePlug Green PHY nodes physically directly connected.

8.3.3.5.1 Requirements

[V2G-DC-059] The EV HomePlug Green PHY node shall try to join the logical network only if the EV_Discovering_Status is “EVSE_FOUND”.

[V2G-DC-060] In order to create the logical network both EV and EVSE shall use the "CM_SLAC_MATCH" MMEs defined in the HomePlug Green PHY specification using sequences defined in Figure 6.

[V2G-DC-061] As soon as the EV has successfully joined the logical network of the EVSE both entities are in the state "matched". The HomePlug Green PHY node of both EV and EVSE shall inform HLE via the D-LINK_READY.indication(DLINKSTATUS = Link established) that HLE’s binding process can begin.

[V2G-DC-581] In “matched” state no further SLAC messages other than CM_AMP_MAP.REQ and CM_AMP_MAP.CNF shall be sent or processed.

8.3.3.5.2 Amplitude map exchange

The amplitude map exchange is an optionally used function to request the counterpart HomePlug Green PHY node to reduce the transmission power for certain carriers. The requesting device sends a CM_AMP_MAP.REQ command, which is confirmed by the counterpart node by a CM_AMP_MAP.CNF message.

Data Link on EV and EVSE side is not synchronized with one HomePlug Green PHY chipset (Figure 8), therefore the timeouts given for AMP_MAP Exchange (e.g. TT_amp_map_exchange) can be violated and amplitude map Exchange might not work.

The transmission power limitation request is related to the PSD at EVs/EVSEs inlet/connector.

The CM_AMP_MAP.REQ MME has two different functions, depending on the source/destination of the message:

— A CM_AMP_MAP.REQ sent from one host to another host via the HomePlug Green PHY Link is designated to transmit the requested transmission power per carrier to the destination host to be included in its amplitude map calculation. This message does not have any direct influence on the HomePlug Green PHY node transmission power, its only used for information exchange;

— A CM_AMP_MAP.REQ sent from one host to the local HomePlug Green PHY chip causes the HomePlug Green PHY chip to modify its transmission power.

All transmission power values within the CM_AMP_MAP MMEs are related to a reference value of -75 dBm/Hz. The resolution of AMDATA is -2 dB, so that a AMDATA value of 0b0011 means a value -6 dB.

[V2G-DC-1003] The amplitude map exchange based reduced transmission power for certain carriers shall not be saved and shall be reset after a plug-out.

8.3.3.5.3 Example for request of a transmission power limitation

(For simplification within this example its total number of carriers is fictitiously set to 6)

Assumption:

EVSE requests a limitation of the power of carriers 2 and 3 to -103 dBm/Hz, while the other carriers keep unaffected. The EV has no limitation request to the EVSE.

— The EVSE sends a CM_AMP_MAP.REQ to the EV host MAC address with the following values: {0,14,14,0,0,0}

  AMDATA[0] = 0x00

  AMDATA[1] = 0x0E

  AMDATA[2] = 0x0E

  AMDATA[3] = 0x00

  AMDATA[4] = 0x00

  AMDATA[5] = 0x00

  Related to the reference of -75 dBm/Hz the requested value of -103 dBm/Hz leads to the difference of - 8 dB which in turn leads to a raw value of 14 (0xE) for 2dB resolution.

  — The EV host compares the received values with its default PSD at the inlet, which are assumed to be

  — {-75, -75, -77, -77, -75, -75} dBm/Hz. For carriers 2 and 3 the requested values are less than the default ones:

  — Carrier 1: -75 dBm/Hz = -75 dBm/Hz: ok Carrier 2: -75 dBm/Hz < -103 dBm/Hz: n.ok. deviation: 28 dBm/Hz Carrier 3: -77 dBm/Hz < -103 dBm/Hz: n.ok. deviation: 26 dBm/Hz Carrier 4: -77 dBm/Hz < -75 dBm/Hz: ok Carrier 5: -75 dBm/Hz = -75 dBm/Hz: ok Carrier 6: -75 dBm/Hz = -75 dBm/Hz: ok

  — The carriers 2 and 3 shall be reduced in power by 28 dBm/Hz and 26 dBm/Hz respectively.

  — The EV host confirms the CM_AMP_MAP.REQ with a corresponding CM_AMP_MAP.CNF

  — Based on the calculated attenuation values for the carriers 2 and 3 the EV host subtract these values from the default HomePlug Green PHY chip setting and writes them to the HomePlug Green PHY chip using the CM_AMP_MAP.REQ and its local destination address.

[V2G-DC- 805] Amplitude Map exchange shall be supported on both EVCC and SECC.

NOTE 1 If EVCC or SECC start AMP MAP exchange is implementation specific and optional. But if the counterpart station requests an AMP map exchange the SECC and EVCC support it.

[V2G-DC-064] In the case that a HomePlug Green PHY node requires additional carriers to be notched, it shall send the amplitude map to the remote HomePlug Green PHY node as soon as the logical network is set up. Therefore, the sequence described in Figure 6 and 8.3.3.5.2 for detailed primitives shall be used. The amplitude map for further communication shall be the intersection of the amplitude map of the local HomePlug Green PHY node and the received amplitude map from the remote HomePlug Green PHY node.

[V2G-DC-582] As long as no exchange of an amplitude map is triggered by one of the HomePlug Green PHY nodes, the default amplitude map of HomePlug Green PHY specification shall be used.

NOTE 2 Local legislation on authorized / forbidden frequencies, in the frequency band 2 – 30 MHz might apply to the EVSE. The EVSE is expected to be able to be reconfigured to future evolution of restrictions.

[V2G-DC-583] The Ethernet destination MAC address field shall be filled as unicast, as defined in Table 6 and Table 7.

[V2G-DC-584] To exchange an amplitude map, the “CM_AMP_MAP.REQ” message of the HomePlug Green PHY specification shall be used. The content is shown in Table 6.

[V2G-DC-585] To confirm an amplitude map exchange, the “CM_AMP.MAP.CNF” message of the HomePlug Green PHY specification shall be used. The content is shown in Table 7.

To exchange an amplitude map, the HomePlug Green PHY node can use the "CM_AMP_MAP.REQ" message defined in the HomePlug Green PHY specification. The content depends on the carriers to be used.

Table 6 — CM_AMP_MAP.REQ parameters

MME	Field	Oct. Num.	Field Size (Bits)	Value	Definition
CM_AMP_MAP.REQ Ethernet unicast	AMLEN	0-1	16	0x395	Number of amplitude data entries – n 0x00=zero 0x01=one ...and so on
	AMDATA[0]	2	4	Var.	Amplitude map data: first unmasked carrier
	AMDATA[n]	n	4	Var.	Amplitude map data: [n] unmasked carrier

To confirm an amplitude map exchange, the HomePlug Green PHY node shall use the "CM_AMP_MAP.CNF" message defined in the HomePlug Green PHY specification. The content is shown in Table 7.

Table 7 — CM_AMP_MAP.CNF parameters

MME	Field	Oct. Num.	Field Size (Bits)	Value	Definition
CM_AMP_MAP.CNF Ethernet Unicast	ResType	0	8	Var.	Response type: 0x00 = Success 0x01 = Failure

[V2G-DC-998] The HomePlug Green PHY node shall use response type 0x00=Success in CM_AMP_MAP.CNF to confirm successful change of amplitude map (PLC PSD of chipset).

[V2G-DC-999] The HomePlug Green PHY node shall use Response Type 0x01=Failure in CM_AMP_MAP.CNF if amplitude map (PLC PSD of chipset) change was not successful.

8.3.3.6 Leave the logical network

This section allows speeding up a disconnection / reconnection of another EV. This sub section avoids that the CCo of the logical network needs several minutes to discover that the EV has left the network.

[V2G-DC-065] With receiving a D-Link_TERMINATE.request from HLE, the HomePlug Green PHY node shall leave the logical network within T_match_leave. All parameters related to the current link shall be set to the default value and the HomePlug Green PHY node shall change to the state "unmatched".

[V2G-DC-527] When the HomePlug Green PHY node leaves the logical network, it shall inform HLE via the D-LINK_READY.indication(DLINKSTATUS = no Link).

[V2G-DC-526] In the following cases, the HomePlug Green PHY node shall leave the logical network within T_match_leave. All parameters related to the current link shall be set to the default value and the HomePlug Green PHY node shall change to the state "unmatched".

— The EVSE detects CP State A, E or F according to IEC 61851-1;

— The EV detects CP State A, E or F according to IEC 61851-1.

8.3.4 Timings, parameters and error handling

This section summarizes all physical and data link layer timings used in this document.

The following naming convention for timers is used in the physical and data link layer chapter. Timers are named according to this scheme:

TP_{EV/EVSE}_NAME = [X; Y] {s/ms}

TT_{EV/EVSE}_NAME = Y {s/ms}

Where the leading T stands for TIMER, followed by:

T for TIMEOUT

P for PERFORMANCE

X for MINIMUM TIME

Y for MAXIMUM TIME

EV/EVSE specifies on which side the timer is running. When nothing specific is specified, the timer runs on both sides.

EXAMPLE

— “TP_EV_match_MNBC_interval = [10; 20] ms“ is a performance timer on the EV side for the interval between transmitted M-SOUNDS with a minimum time of 10 ms and a maximum time of 20 ms.

— “TT_EV_match_response = 200 ms“ is a timeout timer on the EV side which counts the time between an EV request and the anticipated response by the EVSE with a timeout value of 200 ms.

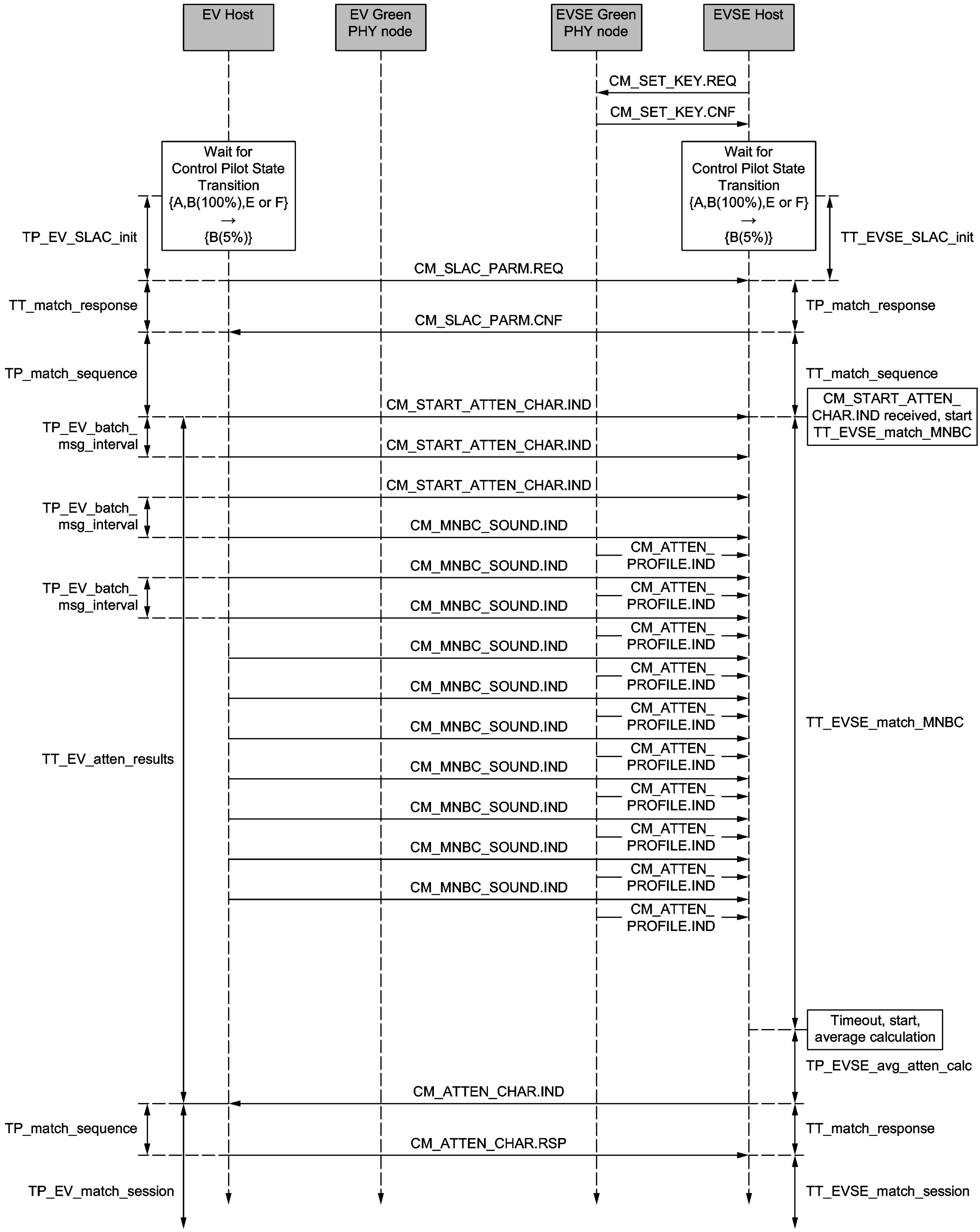
Difference between performance and timeout timer:

[V2G-DC-930] A performance timer on the EVSE side for example shall reach a value larger than the minimum time specified for this timer before the EVSE triggers the event associated with it (e.g. sending a specific message) but the EVSE shall trigger the event before the maximum time value specified for this timer is reached. If the maximum time value is exceeded, this does not constitute a critical error and no action is taken, but this should normally not happen.

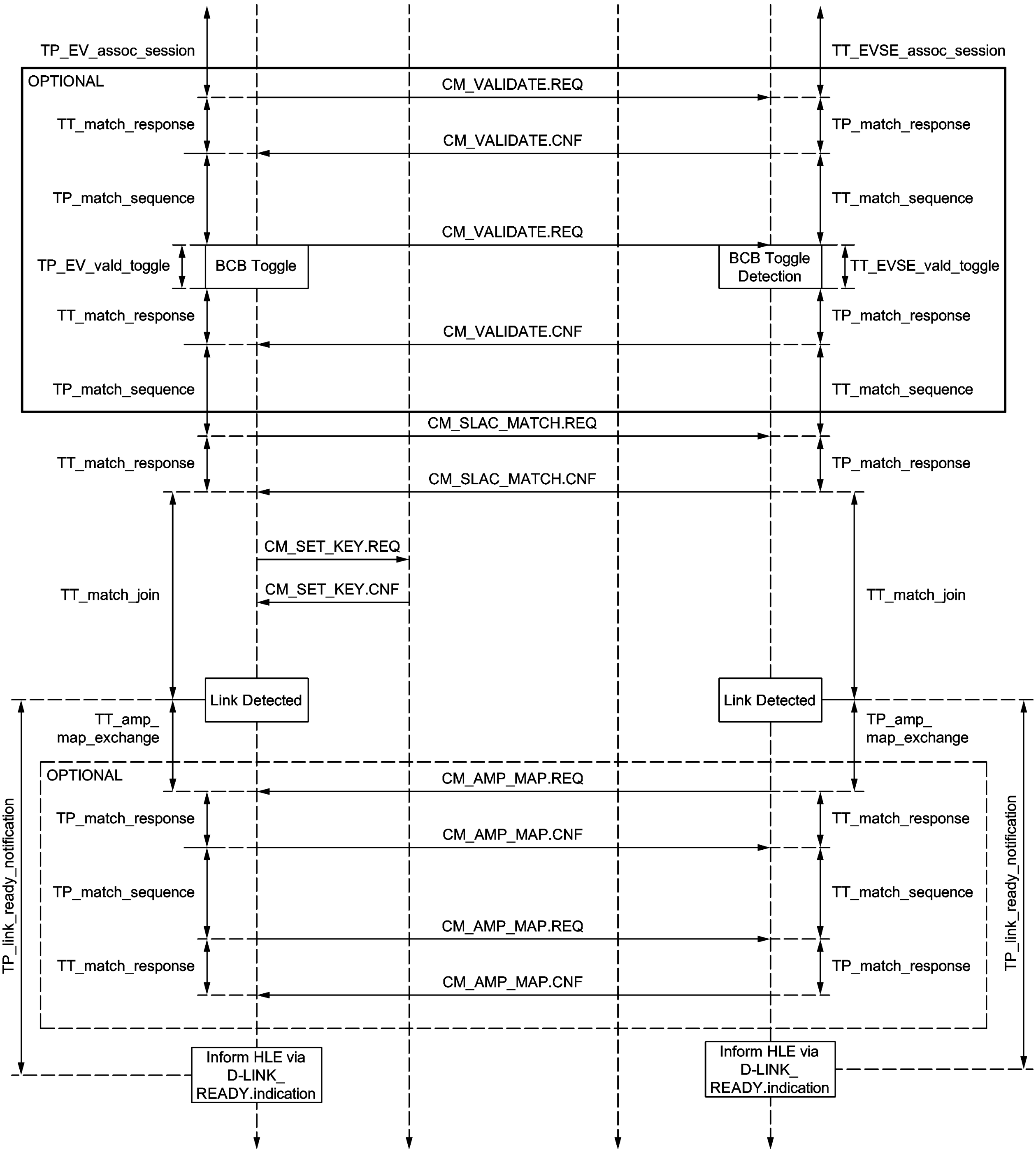
[V2G-DC-931] A timeout timer on the opposite side (EV) is associated with the same event and shall trigger an error handling procedure if it reaches its maximum value specified.

[V2G-DC-932] In a performance timer/ timeout timer pair, the maximum value of the timeout timer shall always be defined higher than the one of the performance timer (at least by twice the maximum possible message propagation time on the power line, but usually more).

[V2G-DC-586] All HomePlug Green PHY nodes shall comply with Figure 8 below.



a) Sequence start



b) Sequence continued

Figure 8 —Matching timing sequence chart

[V2G-DC-519] All HomePlug Green PHY nodes shall comply with Table 8.

Table 8 — Timings and parameters

Parameter	Description	Min	Typical	Max	Unit
TP_EV_SLAC_init	Time between detection of CP state B with 5 % duty cycle or wake-up and sending CM_SLAC_PARM.REQ by the EV	N/A	N/A	10	s
TT_EVSE_SLAC_init	Time the EVSE waits for receiving CM_SLAC_PARM.REQ after the EVSE detects CP state B with 5 % CP duty cycle	20	N/A	50	s
C_EV_match_retry	Number of retries of the corresponding message within the matching process	N/A	N/A	2	nbr
TP_match_sequence	General performance time for subsequent requests after a response to previous request has been received	N/A	N/A	100	ms
TP_match_response	General performance time for a response to a request	N/A	N/A	100	ms
TT_match_response	Time that the EV/EVSE shall wait for a response from the EVSE/EV	N/A	N/A	200	ms
TT_match_sequence	Time that the EVSE/EV shall wait for a request from the EV/EVSE	N/A	N/A	400	ms
TT_EVSE_match_session	Maximum time from CM_ATTEN_CHAR.RSP received and the reception of either CM_VALIDATE.REQ or CM_SLAC_MATCH.REQ	N/A	N/A	10	s
TP_EV_match_session	Performance time for the EV to start validation or SLAC_MATCH after sending CM_ATTEN_CHAR.RSP	N/A	N/A	500	ms
TT_match_join	Maximum time between CM_SLAC_MATCH.CNF and link establishment. If there is no link after this timeout expires, the EV retries the matching process and the EVSE resets its state machine.	N/A	N/A	12	s
TT_amp_map_exchange	Timeout timer that runs on both EV and EVSE side after link is detected. If an EV or EVSE does not want to start an AMP MAP exchange and no request is received by the other side within the timeout value of this timer, then it is to be assumed that no AMP MAP exchange will take place and a D-LINK_READY.indication notification shall be sent to the HLE	N/A	N/A	200	ms
TP_amp_map_exchange	Performance timer for the start of an AMP MAP exchange. Either the EV or the EVSE shall send a CM_AMP_MAP.REQ within the max value of this timer after link is detected to trigger an AMP MAP exchange	N/A	N/A	100	ms
TP_link_ready_notification	Performance timer from data link detected to HLE information with D_LINK_READY primitive	0,2	N/A	1	s
C_vald_nb_toggles	Number of BCB toggles	1	N/A	3	nbr
T_vald_state_duration	Duration of each CP State B or C within the BCB toggle	200	N/A	400	ms
TP_EV_vald_toggle	Duration of BCB toggle sequence	600	N/A	3500	ms
TT_EVSE_vald_toggle	Timeout timer for the EVSE to stop monitoring CP for BCB toggle. Value is received in CM_VALIDATE.REQ	N/A	N/A	3500	ms
T_vald_detect_time	Time to detect a variation of the CP state on EVSE side	N/A	N/A	200	ms

Table 8 (*continued*)

Parameter	Description	Min	Typical	Max	Unit
TP_EV_batch_msg_interval	Interval between two CM_START_ATTEN_CHAR.IND or CM_MNBC_SOUND.IND messages	20	N/A	50	ms
C_EV_match_MNBC	Number of M-Sounds sent for the SLAC	—	10	—	nbr
TT_EVSE_match_MNBC	Timeout on the EVSE side that triggers the calculation of the average attenuation profile	600	—	600	ms
TP_EVSE_avg_atten_calc	Performance time for the EVSE to calculate the average attenuation profile after reception of all M-SOUNDs or after TT_EVSE_match_MNBC has expired	N/A	N/A	100	ms
TT_EV_atten_results	Time the EV shall wait for CM_ATTEN_CHAR.IND messages from EVSEs. Timer starts with the sending of the first CM_START_ATTEN_CHAR.IND	N/A	N/A	1200	ms
C_EV_match_signalattn_direct	Limit for signal strength measurement for direct connected EVSE	N/A	10		dB
C_EV_match_signalattn_indirect	Limit for signal strength measurement if no direct connected EVSE is found	N/A	20		dB
T_match_leave	Maximum time to leave the logical network (e.g. after D-LINK_TERMINATE.request)	N/A	N/A	25	s
C_EV_start_atten_char_inds	Number of CM_START_ATTEN_CHAR.IND messages to be sent	N/A	3		nbr
T_conn_max_comm	Time after plug-in (detection of CP state B or Proximity pin) until the HomePlug Green PHY node shall be ready for communication.	N/A	N/A	8	s

8.3.5 Error handling

The following subclauses defines the error handling for the matching process.

Prerequisites:

— Max(timer_name) is the maximum value a timer is allowed to reach before a timeout occurs according to the Summary of timers and constraints;

— Min(timer_name) is the minimum value that a timer shall reach before an action bound to this timer is performed.

8.3.5.1 Error handling for signal strength measurement parameter exchange

Before the signal strength measurement starts, the EV broadcasts the parameters to be used for the following signal strength measurement sequence by means of the message CM_SLAC_PARM.REQ. Any “unmatched” EVSE that receives the parameter exchange broadcast sends a response to the EV by means of the message CM_SLAC_PARM.CNF. Figure 9 illustrates the corresponding message sequence.

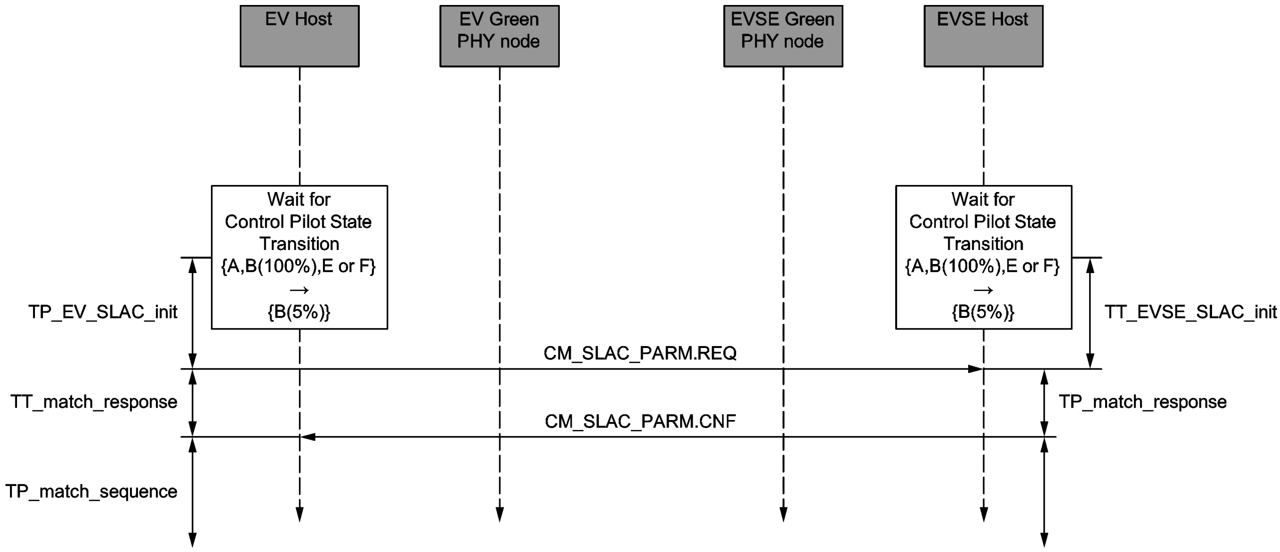


Figure 9 — CM_SLAC_PARM.REQ/CNF sequence chart

8.3.5.1.1 Requirements for EV side

[V2G-DC-960] If not in state "matched", the EV shall send at least one CM_SLAC_PARM.REQ before the timer TP_EV_SLAC_init expires if it wants to charge.

NOTE E.g. in case of 100 % SoC the EV might not start the SLAC process.

[V2G-DC-588] The start condition of the timer TP_EV_SLAC_init shall be CP State B with a CP duty cycle of 5 %.

[V2G-DC-589] In the SLAC parameter exchange phase, the EV shall send a CM_SLAC_PARM.REQ and wait for the maximum value of TT_match_response for CM_SLAC_PARM.CNFs from potential EVSEs.

[V2G-DC-590] The start condition of TT_match_response shall be the transmission of CM_SLAC_PARM.REQ

[V2G-DC-591] If the EV receives a CM_SLAC_PARM.CNF with invalid content, it shall be ignored. Content which deviates from the MME definition in Table 2 is invalid.

[V2G-DC-592] If no valid CM_SLAC_PARM.CNF arrives at the EV when TT_match_response expires, the EV shall retransmit the request and wait again for TT_match_response. The total number of retries is given by C_EV_match_retry. If C_EV_match_retry is reached, the matching process shall be considered as FAILED.

8.3.5.1.2 Requirements for EVSE side

[V2G-DC-593] If not in state "matched" and as long as the timer TT_EVSE_SLAC_init is not expired, the EVSE shall answer to valid CM_SLAC_PARM.REQs.

[V2G-DC-594] The start condition of the timer TT_EVSE_SLAC_init shall be CP State B with a CP duty cycle of 5 %.

[V2G-DC-596] If the EVSE receives a CM_SLAC_PARM.REQ with invalid content, the EVSE shall ignore it and shall not stop the timeout timer TT_EVSE_SLAC_init. Content which deviate from the MME definition in Table 2 is invalid.

[V2G-DC-597] When receiving a CM_SLAC_PARM.REQ, the EVSE shall answer by sending a CM_SLAC_PARM.CNF response within TP_match_response.

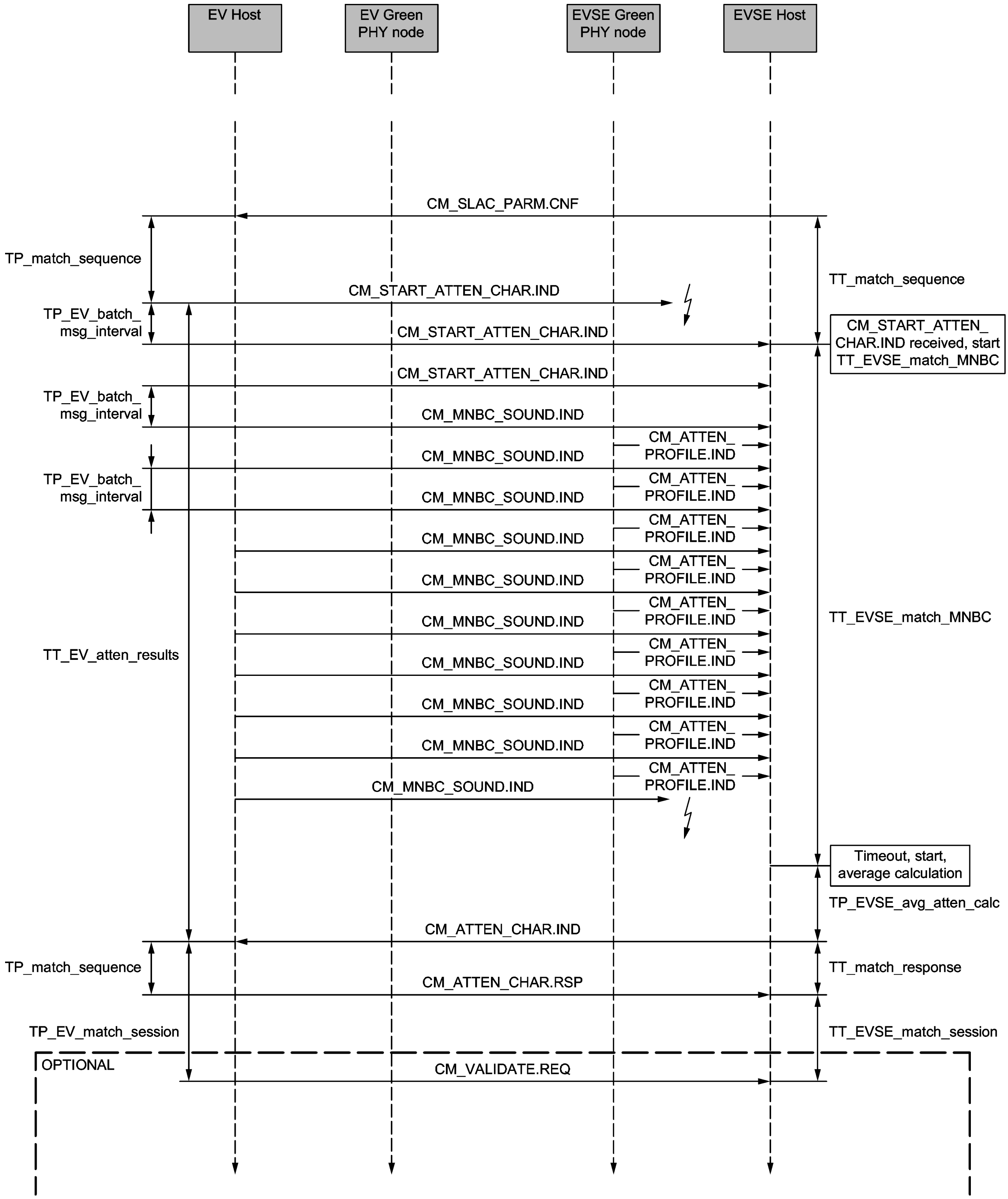
[V2G-DC-598] If during a matching process the EVSE receives a CM_SLAC_PARM.REQ from the EV which participates in the ongoing matching process, the EVSE shall restart its state machine and reply to this request because it shall be considered as a new retry by the EV.

[V2G-DC-860] An EVSE shall at least wait for 20 s after first CM_SLAC_PARM.REQ received for SDP request, before it is allowed to switch off the SDP server.

NOTE For EVSEs supporting ISO 15118 and this document, wake up might cause the SDP Server to be active after pausing.

8.3.5.2 Error handling for signal strength measurement

Figure 10 illustrates the message sequence regarding the error handling for the signal strength measurement.



Key

Message does not reach its destination.

Figure 10 — Signal strength measurement Sequence Chart

8.3.5.2.1 Requirements for EV side

[V2G-DC-674] With receiving a CM_SLAC_PARM.CNF the EV shall start the timer TP_match_sequence. When the TP_match_sequence timer expires, the EV shall send a sequence of C_EV_start_atten_char_inds CM_START_ATTEN_CHAR.IND messages.

[V2G-DC-829] The EV shall be able to process and answer 5 CM_SLAC_PARM.CNF messages in parallel.

[V2G-DC-675] The time duration between consecutive CM_START_ATTEN_CHAR.IND messages shall be TP_EV_batch_msg_interval.

[V2G-DC-676] After sending the last message of the CM_START_ATTEN_CHAR.IND message sequence, the EV shall wait for TP_EV_batch_msg_interval before starting the CM_MNBC_SOUND.IND message sequence.

[V2G-DC-677] Within CM_MNBC_SOUND.IND message sequence the EV shall transmit the CM_MNBC_SOUND.IND message C_EV_match_MNBC times. With each message, the counter field “cnt” shall be decremented.

[V2G-DC-678] The time duration between consecutive CM_MNBC_SOUND.IND messages is defined to TP_EV_batch_msg_interval.

[V2G-DC-679] The EV shall start the timeout timer TT_EV_atten_results when sending the first CM_START_ATTEN_CHAR.IND.

[V2G-DC-680] While the timer TT_EV_atten_results is running, the EV shall process incoming CM_ATTEN_CHAR.IND messages. If the CM_ATTEN_CHAR.IND messages from all EVSEs are received, which were recognized during the parameter exchange, the EV is allowed to stop the TT_EV_atten_results timer and continue the matching process. To be able to service also EVSEs which were not received during the parameter exchange the EV may wait for incoming CM_ATTEN_CHAR.IND messages until the TT_EV_atten_results timer expires before continuing the matching process.

[V2G-DC-830] The EV shall be able to process and answer 5 CM_ATTEN_CHAR.IND messages in parallel.

NOTE This is necessary if crosstalk occurs and SLAC packets are received from not direct connected SECCs.

[V2G-DC-681] If no CM_ATTEN_CHAR.IND is received before TT_EV_atten_results expires, the matching process shall be considered as FAILED.

[V2G-DC-682] If a CM_ATTEN_CHAR.IND is received and its origin is an EVSE that has not sent a CM_SLAC_PARM.CNF before, the message shall be processed and not ignored.

[V2G-DC-683] If the TT_EV_atten_results timer expires and not all anticipated responses are received, the EV shall continue the matching process.

[V2G-DC-684] If the EV receives a CM_ATTEN_CHAR.IND with invalid content, it shall be ignored. Content which deviates from the MME definition in Table 2 is invalid.

[V2G-DC-806] If the NUM_SOUNDS field in a CM_ATTEN_CHAR.IND is <7, then the ATTEN_PROFILE has no significance and the EVCC shall stop the SLAC communication. It is up to the EV to decide what number of M-SOUNDS used for the attenuation profile is sufficient for its decision (i.e. whether to discard CM_ATTEN_CHAR.IND if NUM_SOUNDs is less than C_EV_match_MNBC).

[V2G-DC-686] On reception of CM_ATTEN_CHAR.IND, the EV shall answer by sending the CM_ATTEN_CHAR.RSP within TP_match_sequence.

[V2G-DC-687] After sending a CM_ATTEN_CHAR.RSP, the EV shall continue the matching process with the found and/or potentially found EVSEs by sending either a CM_SLAC_MATCH.REQ to the selected EVSE or a CM_VALIDATE.REQ to one of the potential EVSEs within TP_EV_match_session.

8.3.5.2.2 Requirements for EVSE side

[V2G-DC-688] With sending the CM_SLAC_PARM.CNF in the parameter exchange sequence the EVSE shall start a timer TT_match_sequence. The reception of a single valid CM_START_ATTEN_CHAR.IND message shall be sufficient stop the timer TT_match_sequence.

[V2G-DC-689] If the EVSE has not received a CM_START_ATTEN_CHAR.IND within TT_match_sequence, the matching process shall be considered as FAILED.

[V2G-DC-690] If the EVSE receives a CM_START_ATTEN_CHAR.IND with invalid content, it shall be ignored. Content which deviates from the MME definition in Table 2 is invalid.

[V2G-DC-691] On reception of a CM_START_ATTEN_CHAR.IND message, the EVSE shall start the TT_EVSE_match_MNBC timer.

[V2G-DC-692] While the timer TT_EVSE_match_MNBC is running, the EVSE shall receive and process incoming CM_ATTEN_PROFILE.IND messages. If the anticipated number of CM_ATTEN_PROFILE.IND messages is not achieved, the EVSE shall keep listening for incoming CM_ATTEN_PROFILE.IND messages until the timer TT_EVSE_match_MNBC expires.

[V2G-DC-822] If up to the number of C_EV_match_MNBC M-Sound messages are received by the EVSE or the TT_EVSE_match_MNBC timer expires the EVSE shall compute (analyze and average) up to number of C_EV_match_MNBC received CM_ATTEN_PROFILE.IND messages within the time window given by the TP_EVSE_avg_atten_calc timer. This performance timer shall be started as soon as TT_EVSE_match_MNBC expires.

[V2G-DC-694] After having computed all received data, the EVSE shall send a CM_ATTEN_CHAR.IND within TP_EVSE_avg_atten_calc and start the timer TT_match_response.

[V2G-DC-695] If the EVSE has not received a CM_ATTEN_CHAR.RSP within TT_match_response, it shall retransmit a CM_ATTEN_CHAR.IND when the TT_match_response expires, and reset this timer. After C_EV_match_retry attempts, if no CM_ATTEN_CHAR.RSP is received by the EVSE, the matching process shall be considered as FAILED.

[V2G-DC-696] If the EVSE receives a CM_ATTEN_CHAR.RSP with invalid content, it shall be ignored. Content which deviates from the MME definition in Table 2 is invalid.

8.3.5.3 Error handling for matching validation

The validation of the matching decision is a method to validate the SLAC based matching by means of the control pilot line. In the case the EV is not able to come to a clear matching decision based on the signal strength measurement, it may decide to do an additional validation step. The validation process is divided in two parts consisting of a CM_VALIDATE.REQ and CM_VALIDATE.CNF each. It is important to note that the first and second request-response differ in the contents of the messages. In the following, messages related to the first/second request-response are marked with “step 1” or “step 2”, respectively. Figure 11 illustrates the message sequence regarding the error handling for matching validation.

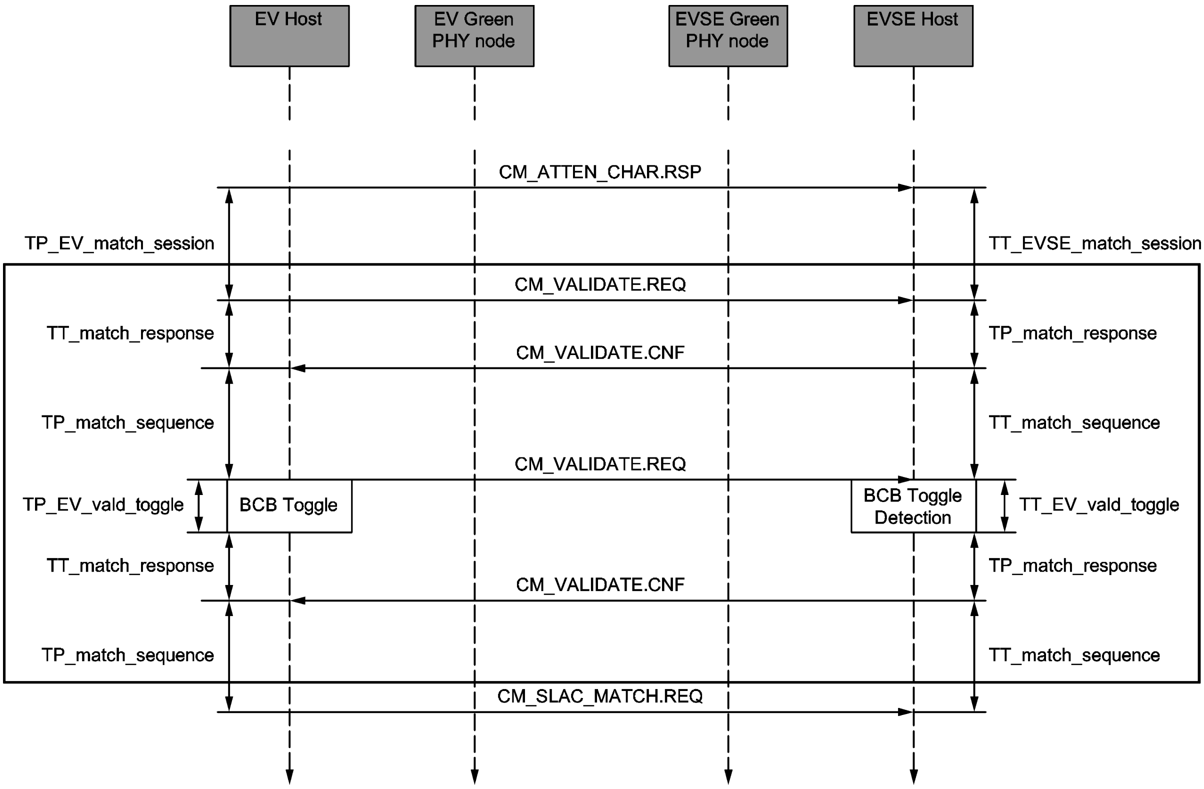


Figure 11 — Matching validation sequence chart

8.3.5.3.1 Requirements for EV side

[V2G-DC-697] With sending the step 1 CM_VALIDATE.REQ the EV shall start the timer TT_match_response. When the TT_match_response timer expires the EV shall resend the step 1 CM_VALIDATE.REQ with C_EV_match_retry number of retries. The timer shall be restarted with any retry. The TT_match_response timer shall stop with the reception of a valid CM_VALIDATE.CNF.

[V2G-DC-698] If the EV does not receive any valid CM_VALIDATE.CNF for all retries, the EV shall stop the Validation process with the current EVSE and continue the validation process with the next potential EVSE.

[V2G-DC-699] If any received CM_VALIDATE.CNF contains invalid content, it shall be ignored. Content which deviates from the MME definition in Table 4 and Table 5 is invalid.

[V2G-DC-700] If the result field of the step 1 CM_VALIDATE.CNF contains “Failure” or “Success” the EV shall stop the validation process with the current EVSE and continue the validation process with the next potential EVSE.

[V2G-DC-824] If the result field of the step 1 CM_VALIDATE.CNF contains “Not ready” the EV shall either retry within sequence according to [V2G-DC-825] or continue the validation with the next potential EVSE with the option to retry the validation with the “not ready” EVSE again afterwards.

[V2G-DC-702] If the result field of the step 1 CM_VALIDATE.CNF contains “not required” the EV can skip the validation process.

[V2G-DC-703] If the result field of the step 1 CM_VALIDATE.CNF contains “ready” the EV shall send the step 2 CM_VALIDATE.REQ within TP_match_sequence with a Timer value which covers the whole BCB toggle sequence.

[V2G-DC-704] With sending the step 2 CM_VALIDATE.REQ the EV shall start the timer TP_EV_vald_toggle and shall start the BCB toggle sequence. The timer TP_EV_vald_toggle shall be equal to the timer value sent within the step 2 CM_VALIDATE.REQ.

[V2G-DC-705] The BCB toggle sequence shall be finished before the TP_EV_vald_toggle timer expires.

[V2G-DC-706] When TP_EV_vald_toggle timer expires, the EV shall start the TT_match_response timer. While the TT_match_response timer is running, the EV shall listen for the incoming step 2 CM_VALIDATE.CNF MME. The TT_match_response timer shall stop as soon as a valid CM_VALIDATE.CNF was received.

[V2G-DC-707] If the TT_match_response timer expires, the EV shall stop the validation process with the current EVSE and continue the validation process with the next potential EVSE.

[V2G-DC-708] If the result field of the step 2 CM_VALIDATE.CNF contains “Failure”, “Not Ready” or “Not Required” the EV shall stop the validation process with the current EVSE and continue the validation process with the next potential EVSE.

[V2G-DC-709] If the result field of the step 2 CM_VALIDATE.CNF contains “success” the EV shall compare the ToggleNum field of the CM_VALIDATE.CNF message with the number of BCB toggles executed. If the numbers are equal, the EVSE matching status is confirmed as “EVSE_FOUND” otherwise the EV shall continue the validation process with the next potential EVSE.

[V2G-DC-710] If the matching status is confirmed by the validation, the EV shall continue with the CM_SLAC_MATCH.REQ within TP_match_sequence after receiving the step 2 CM_VALIDATE.CNF.

8.3.5.3.2 Requirements for EVSE side

[V2G-DC-711] With receiving a step 1 CM_VALIDATE.REQ from the EV which started the matching process and with a Timer field equal to “0x00”, the EVSE shall respond with a CM_VALIDATE.CNF within TP_match_response.

[V2G-DC-712] If any received CM_VALIDATE.REQ contains invalid content, it shall be ignored. Content which deviates from the MME definition in Table 4 and Table 5 is invalid.

[V2G-DC-713] If the EVSE receives another CM_VALIDATE.REQ with a timer field equal to zero after sending out the step 1 CM_VALIDATE.CNF, the step 1 CM_VALIDATE.CNF shall be resent.

[V2G-DC-714] If the EVSE is occupied by another running validation process, it shall set the result field to “Not Ready”.

[V2G-DC-715] If the EVSE is not occupied, it shall continue the validation setting the Result field to “Ready” or “Not Required”.

[V2G-DC-823] If the EVSE does not support any BCB toggle due to a failure, it shall set the result field to “Failure”.

[V2G-DC-717] If the EVSE has sent the step 1 CM_VALIDATE.CNF with result field set to “Not Required” it shall be prepared that the EV will continue the process with the step 2 CM_VALIDATE.REQ or a CM_SLAC_MATCH.REQ message.

[V2G-DC-825] With sending the step 1 CM_VALIDATE.CNF the EVSE shall start the timer TT_match_sequence. When the TT_match_sequence timer expires, the EVSE shall resend the step 1 CM_VALIDATE.CNF with C_EV_match_retry number of retries. The timer shall be restarted with any retry.

[V2G-DC-826] The TT_match_sequence shall stop with the reception of a valid step 1 CM_VALIDATE.REQ or a valid step 2 CM_VALIDATE.REQ or with a valid CM_SLAC_MATCH.REQ.

[V2G-DC-827] If the sequence according to [V2G-DC-825] expires the matching process shall be considered as FAILED from the EVSE.

[V2G-DC-719] If the EVSE does not receive any valid step 2 CM_VALIDATE.REQ or CM_SLAC_MATCH.REQ for all retries, the matching process shall be considered as FAILED.

[V2G-DC-720] If any CM_VALIDATE.REQ contains a result field other than “Ready”, the matching process shall be considered as FAILED.

[V2G-DC-721] With receiving a step 2 CM_VALIDATE.REQ from the EV which started the matching process, the EVSE shall start the timer TT_EVSE_vald_toggle.

[V2G-DC-722] While the timer TT_EVSE_vald_toggle is running, the EVSE shall count the BCB-Toggles on the control pilot.

[V2G-DC-723] After the timer TT_EVSE_vald_toggle is expired, the EVSE shall send a step 2 CM_VALIDATE.CNF with the number of counted BCB-Toggles and the result field set to “Success” within TP_match_response.

[V2G-DC-724] In case an error occurs on EVSE side while counting the BCB-Toggles, so that the ToggleNum field does not contain valid data, the result field shall be set to “Failure”.

[V2G-DC-725] With sending the step 2 CM_VALIDATE.CNF the EVSE shall start the timer TT_match_sequence. The timer shall stop as soon as a CM_SLAC_MATCH.REQ message is received.

[V2G-DC-726] When the timer TT_match_sequence expires, the matching process shall be considered as FAILED.

8.3.5.4 Error handling for logical network parameter exchange

After the EV has finished the matching decision, it requests the parameter for the logical network from the selected EVSE by means of the message CM_SLAC_MATCH.REQ. The selected EVSE responses to the EV request with a CM_SLAC_MATCH.CNF, which contains all parameters to be set to join the logical network of the EVSE. Figure 12 illustrates the message sequence regarding the error handling for the logical network parameter exchange.

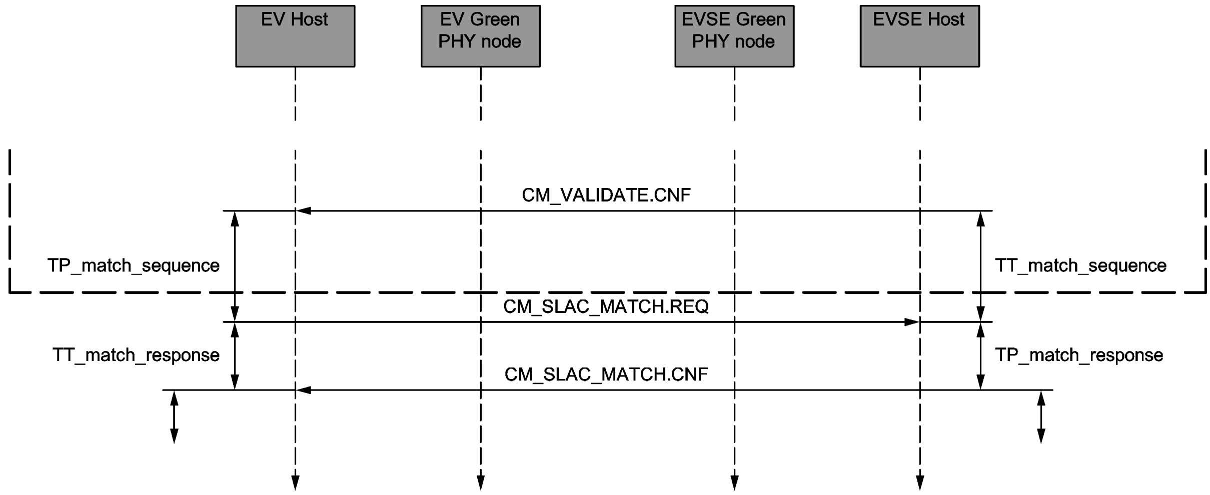


Figure 12 — Logical network parameter exchange sequence chart

8.3.5.4.1 Requirements for EV side

[V2G-DC-727] If the EV has sent a CM_SLAC_MATCH.REQ to the EVSE, but does not receive a valid CM_SLAC_MATCH.CNF within the max(TT_match_response), it shall retransmit the CM_SLAC_MATCH.REQ. The timer shall be restarted with any retry. A maximum of C_EV_match_retry retransmissions shall be performed. If after these retransmissions, the EV has not received valid response within max(TT_match_response), the matching process shall be considered as FAILED.

[V2G-DC-728] If the EV receives a CM_SLAC_MATCH.CNF with invalid content, it shall be ignored. Content which deviates from the MME definition in Table 2 is invalid.

8.3.5.4.2 Requirements for EVSE side

[V2G-DC-828] According to the decision of the EV, if the EVSE does not receive either CM_SLAC_MATCH.REQ or a Step 1 CM_VALIDATE.REQ within the maximum value of TT_EVSE_match_session, the EVSE shall assume that it is not connected to the EV and shall consider the matching process as FAILED.

[V2G-DC-730] If the EVSE receives another CM_SLAC_MATCH.REQ from the same EV, this means that the EV retransmitted its request for some reason (i.e. the CM_SLAC_MATCH.CNF was not received). The EVSE shall respond to the request again.

[V2G-DC-731] If the EVSE receives a CM_SLAC_MATCH.REQ with invalid content, it shall be ignored. Content which deviates from the MME definition in Table 2 is invalid.

[V2G-DC-732] After receiving a CM_SLAC_MATCH.REQ, the EVSE shall answer by a CM_SLAC_MATCH.CNF within TP_match_response.

8.3.5.5 Error handling for joining the logical network

After the right matching between EV and EVSE is determined and the network parameters are exchanged, the EV joins the logical network of the EVSE. Through the broadcast domain is reduced to the HomePlug Green PHY nodes directly connected. Figure 13 illustrates the message sequence regarding the error handling for joining the logical network.

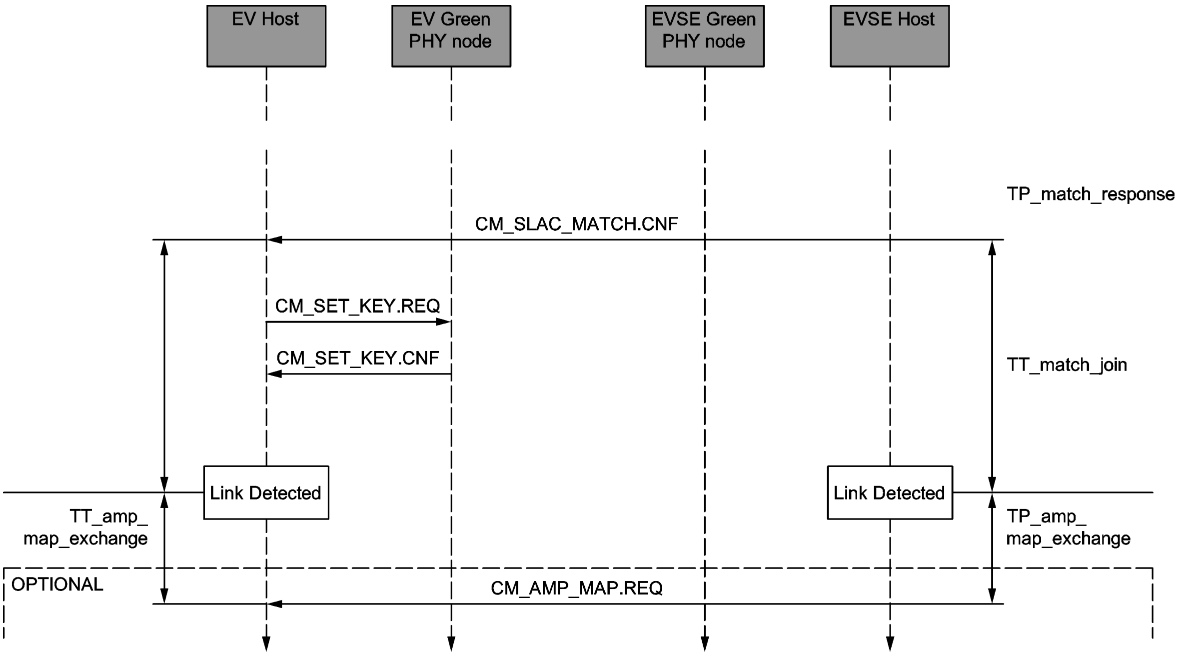


Figure 13 — Joining the logical network sequence chart

8.3.5.5.1 Requirements for EV side

[V2G-DC-599] After receiving a CM_SLAC_MATCH.CNF from the EVSE, the EV shall configure its HomePlug Green PHY node to the values (NID and NMK) from this message by using the CM_SET_KEY.REQ MME.

In a specific implementation, other methods than the CM_SET_KEY MMEs may also be used as long as the configuration result is equivalent.

NOTE It is up to the implementation how to handle a negative response in a CM_SET_KEY.CNF or a missing CM_SET_KEY.CNF at all.

[V2G-DC-600] If no link is established within the max(TT_match_join) after receiving CM_SLAC_MATCH.CNF, the EV shall consider the matching process as FAILED.

8.3.5.5.2 Requirements for EVSE side

[V2G-DC-601] After sending a CM_SLAC_MATCH.CNF containing a NMK and a NID, the EVSE shall start its TT_match_join timer. This timer ends when the CCo detects a link in its logical network.

[V2G-DC-602] If no link is detected when the TT_match_join timer expires, the EVSE shall consider the matching process as FAILED.

[V2G-DC-603] The EVSE shall configure its HomePlug Green PHY node to the NID and NMK values sent in CM_SLAC_MATCH.CNF at the latest after sending the CM_SLAC_MATCH.CNF. The configuration can also be done at any time before. (e.g. after unplugging of a previous EV). The configuration shall be done by sending a CM_SET_KEY.REQ.

If the EVSE needs to configure its NMK and NID after the CM_SLAC_MATCH.CNF, it should consider that the just sent CM_SLAC_MATCH.CNF may get lost and the EV will send a CM_SLAC_MATCH.REQ retry within TT_match_response. Within this time, the EVSE should not be blind, due to the configuration process, for the incoming CM_SLAC_MATCH.REQ retry.

8.3.5.6 Error handling for amplitude map exchange

Figure 14 illustrates the message sequence regarding the error handling for the amplitude map exchange.

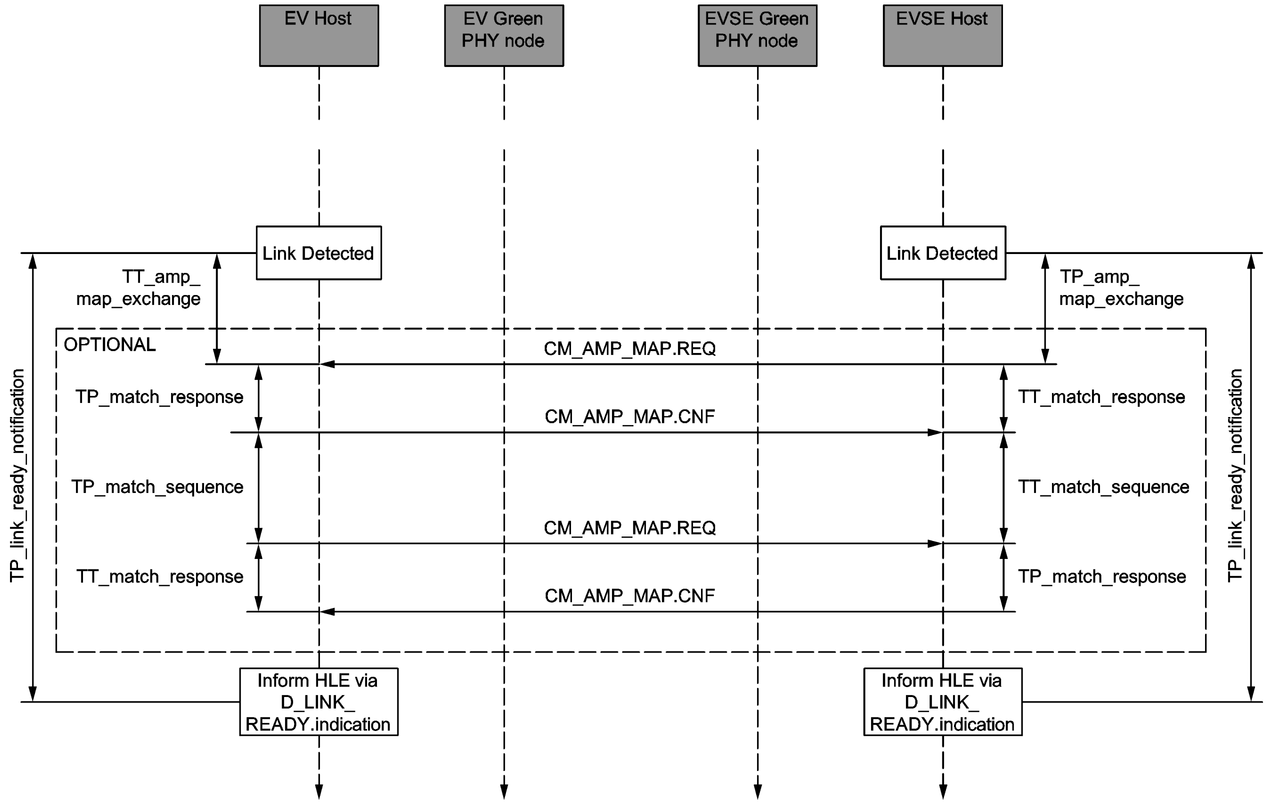


Figure 14 — Amplitude map exchange sequence chart

8.3.5.6.1 Requirements for EV and EVSE side

[V2G-DC-604] If a HomePlug Green PHY node needs to request an amplitude map exchange from the counterpart device, the CM_AMP_MAP.REQ shall be sent within TP_amp_map_exchange. The timer TP_amp_map_exchange shall be started with the detection of other Stations in the current logical network after the SLAC_MATCH exchange.

[V2G-DC-605] If a HomePlug Green PHY node has sent a CM_AMP_MAP.REQ, but does not receive a valid CM_AMP_MAP.CNF within the max(TT_match_response), it shall retransmit the CM_AMP_MAP.REQ. The timer shall be restarted with any retry. A maximum of C_EV_match_retry retransmissions shall be performed. If after these retransmissions, the HomePlug Green PHY node has not received valid response within max(TT_match_response), the matching process shall be considered as FAILED.

[V2G-DC-606] If a HomePlug Green PHY node receives a CM_AMP_MAP.REQ with invalid content, it shall be ignored. Content which deviates from the MME definition in Table 6 is invalid.

[V2G-DC-607] If a HomePlug Green PHY node receives a CM_AMP_MAP.CNF with invalid content, it shall be ignored. Content which deviates from the MME definition in Table 7 is invalid.

[V2G-DC-961] A Green PHY node shall send CM_AMP_MAP.CNF with ResType

— 0x00 success: when reduction of transmission power of certain carriers was successful;

— 0x01 failure: when reduction of transmission power of certain carriers was not successful or is not supported.

[V2G-DC-608] After receiving a CM_AMP_MAP.REQ within TT_amp_map_exchange, a HomePlug Green PHY node shall answer with a CM_AMP_MAP.CNF within TP_match_response. The timer TT_amp_map_exchange shall be started with the detection of other stations in the current logical network.

[V2G-DC-609] If the requested node receives another CM_AMP_MAP.REQ, this means that the EV retransmitted its request for some reason (i.e. the CM_AMP_MAP.CNF was not received), the requested node shall respond to the request again.

[V2G-DC-610] If the timer TT_amp_map_exchange expires without receiving a CM_AMP_MAP.REQ from the counterpart HomePlug Green PHY node and without sending a CM_AMP_MAP.REQ, the D_LINK_READY(LINK_Established) shall be sent to higher layers.

[V2G-DC-611] If an amplitude map exchange is initiated by one of the HomePlug Green PHY nodes in the logical network, the higher layer shall be informed about the valid HomePlug Green PHY Link not before the link is re-established after the amplitude map exchange and the local configuration of the modified amplitude map is done.

[V2G-DC-612] The time between the detection of other stations in the current logical network after the SLAC_MATCH exchange and the indication D_LINK_READY(LINK_Established) to higher layers shall not exceed the performance timer TP_link_ready_notification.

8.3.5.7 Miscellaneous error handling rules

[V2G-DC-613] In case the matching process fails due to an error, the matching process shall be quit in “unmatched” state (see Figure 5). This error shall be handled by the connection coordination module.

[V2G-DC-614] In case a plug-out is detected during the matching process by the proximity pin, the matching process shall be assumed as FAILED.

[V2G-DC-943] In any case a control pilot state E or state F is detected during the matching process, the matching process shall be stopped in “unmatched” state.

In case matching process is Failed the control pilot could stay at 5% or change to state B1. To allow re-start of the matching process, it should switch off the oscillator.

8.3.6 EMC requirements

[V2G-DC-616] All HomePlug Green PHY nodes shall notch out the carriers listed in Table 9 in the tone mask.

Table 9 — Notched carriers

HomePlug Green PHY carrier
0 - 85
140 - 167
215 - 225
283 - 302
410 - 419

Table 9 (*continued*)

HomePlug Green PHY carrier
570 - 591
737 - 748
857 - 882
1016 - 1027
1144 - 1535

NOTE This table equals to the North American tone mask of the HomePlug Green PHY specification.

Additional EMC requirements are defined in IEC 61851-1 and IEC 61851-23.

8.3.7 Coupling circuit

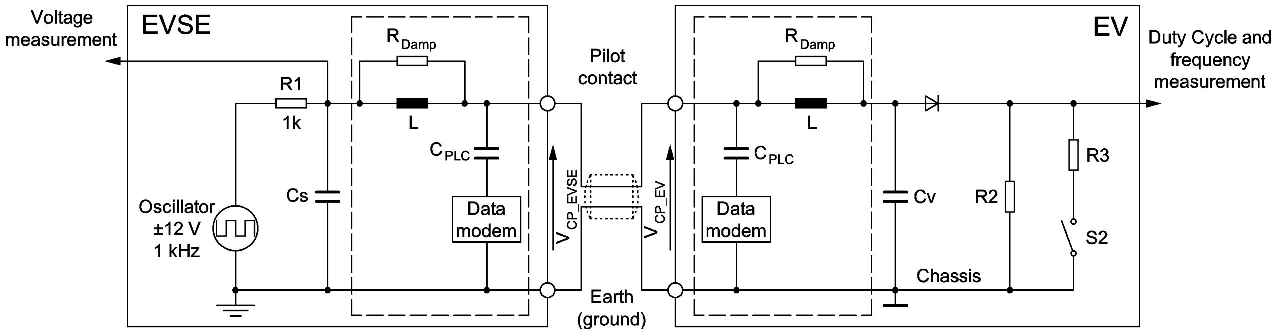
8.3.7.1 Overview

As shown in the previous sub-clauses the physical layer is divided into two parts: The HomePlug Green PHY communication and the basic signalling in accordance with IEC 61851-1. This subclause defines the requirements that need to be fulfilled to inject HomePlug Green PHY signals into the CPL to enable bidirectional HomePlug Green PHY communication between one EVSE and one EV.

HomePlug Green PHY signals are compliant with the HomePlug Green PHY specification. The target is to enable bidirectional HPGP communication between one EVSE and one EV, while the CPLT simultaneously works according to IEC 61851-1. As HomePlug Green PHY signal coupling is directly linked to the control pilot, the schematic in IEC 61851-1 and its Annex A is basis for definitions regarding the control pilot signal. To enable HomePlug Green PHY injection, the path from EVSE's output to the EV's input should not be considered as capacitive only, as it is sufficient for the low frequency control pilot signal. Instead, for HomePlug Green PHY communication (which are high frequency signals), the path should be considered as a transmission line. Considering high frequency, the control pilot circuit and the component values are specified more in detail. For the control pilot line, the additional capacity of the HomePlug Green PHY coupling circuit shall be considered. For the HomePlug Green PHY signal, the partitioning of the capacitive load of the EV, EVSE and the charging cord is important and is described herein.

8.3.7.2 General drawing for HomePlug Green PHY injection

[V2G-DC-071] All technical requirements described in this subclause shall assume to have a dedicated pair of HomePlug Green PHY chips implemented for the couple (EVSE, EV), linked by a CPL wire. Figure 15 depicts an implementation of parallel injection and gives definitions used further.



Key

Additional components for PLC coupling

Figure 15 — Implementation of parallel injection

Different topologies like point-to-multipoint are not covered and can require adaptations.

NOTE 1 The coupling drawing is valid for cases A, B and C, as defined in IEC 61851-1.

NOTE 2 The coupling capacitors are equivalent components, which can be implemented with more than one component, for example 2 capacitors, as soon as the requirements about values given in the table below is fulfilled.

NOTE 3 L and R_{Damp} associated with capacitors, form low pass filters, which enable the normal working of the CPLT and the Green PHY communication at the same time. Via the filters, low frequency signals can pass to the Control Pilot line and high frequency signals are available at the HomePlug Green PHY node. The R_{Damp} resistors limit resonance effects of the RLC filters.

[V2G-DC-072] Signal coupling shall operate on a point-to-point basis with only one HomePlug Green PHY node on EVSE side connected to one HomePlug Green PHY node on EV side.

[V2G-DC-075] In case of parallel injection, the HomePlug Green PHY signal shall be coupled between the CPL wire and the PE (Protective Conductor) wire.

[V2G-DC-076] In case of parallel injection, the HomePlug Green PHY injection circuit on EVSE side shall be wired to the CPL and the PE. Additional components, such as EMC or ESD protection, should not affect the HomePlug Green PHY signals.

[V2G-DC-077] In case of parallel injection, the HomePlug Green PHY injection circuit on EV side shall be wired to the CPL and PE. Additional components, such as EMC or ESD protection, should not affect the HomePlug Green PHY signals.

[V2G-DC-533] HomePlug Green PHY communication shall work with any CP duty cycle or CP State as defined in IEC 61851-1.

[V2G-DC-078] A proper formatting of the CPL raw signal shall be implemented on EVSE side to ensure compliance with IEC 61851-1 in presence of an additional HomePlug Green PHY communication signal.

It is highly recommended to apply at least a 1st order low pass filter with a cut-off frequency of 100 kHz to 200 kHz for measurements on the CPL signal (e.g. duty cycle or amplitude), in order not to disturb the measurement by high frequency HomePlug Green PHY signals.

8.3.7.3 Signal requirement for HomePlug Green PHY injection

The following table gives requirements that physical signals shall comply with to enable HomePlug Green PHY injection into the CPL, according to previous requirements.

The HomePlug Green PHY signal shall be added to the CPLT signal on the CPL. The resulting signal should be the algebraic sum of the PWM CPLT signal and the HomePlug Green PHY signal.

[V2G-DC-081] The CPL and the PE wires shall be regarded as a transmission line for the HomePlug Green PHY signal, with characteristic impedance as defined in Table 10.

Table 10 — Signal requirements for parallel HomePlug Green PHY injection to Control Pilot

Parameter	**Conditions and comments**	**min**	**typ**	**max**	**unit**	**Note**
Length of the charging cable				10	M	
$C_{Green\ PHY}$	See definition above	—	1.35		nF	3,4
R_{Damp}	See definition above	—	220	1k	Ω	4
L	See definition above	—	220		µH	3,4
C_S	See definition in IEC 61851-1				pF	1,3
C_V	See definition in IEC 61851-1				pF	1,3
Power spectral density of Green PHY signals at V_{CP_EVSE} and V_{CP_EV} Measurement method defined in 8.3.7.4.3	From 1.8 to 30 MHz, RBW=9 kHz on 50 Ω. Min/Typ applicable for 80% of all carriers Max applicable for 100 % of all carriers	-80	-75	-73	dBm/Hz	5,6
Peak-Peak Voltage of Green PHY signals at V_{CP_EVSE} and V_{CP_EV}	CPL signal steady at high or low level – 1 Green PHY communication actually emitting at a time (peak to peak) – EV connected to the EVSE with the charging cord. Measured at max PSD level of Green PHY communication with example of injection circuit given below.	—	1.3		Vpp	2,3
Conducted HomePlug Green PHY crosstalk from CPL to the mains (via power supply) (voltage between any live or neutral wire and PE/CPL)	From 2 MHz to 28 MHz			-40	dB	

Any capacity on the CPL which is directly connected between CPL and PE should be as small as possible, so as not to excessively attenuate the HomePlug Green PHY signal. Most of the capacitive load should be separated from the high frequency HomePlug Green PHY signal with the inductor L.

NOTE 1 At given impedance, the maximum peak to peak amplitude directly linked to the PSD is also defined in Table 10. For limiting the impact of the high frequency HomePlug Green PHY signal on the control pilot line, a simplification from definition in frequency domain to a peak-to-peak voltage is sufficient.

NOTE 2 The maximum values of Cs and Cv are defined by IEC 61851 Annex A.

NOTE 3 Parameters in Table 10 are to be considered with parallel injection only, as defined in Figure 15.

NOTE 4 For the min/typ PLC PSD Level additional, individual notching of carriers might be necessary/requested for EV or EVSE (e.g. amplitude map exchange). To allow additional notching the value only covers 80 % of the carriers from 1.8 Mhz – 30 MHz.

NOTE 5 If PLC transmission power is too high EMC limits for conductive emission on DC lines given in IEC61851-21-1 and -2 cannot be met. (crosstalk from CP Line to DC lines). This is why the max value is applicable to 100 % of the carriers from 1.8 Mhz – 30 MHz.

NOTE 6 It is recommended to use a twisted pair cable for CP/PE lines within the EV (from HomePlug Green PHY transceiver to vehicle inlet) and the EVSE (from HomePlug Green PHY transceiver to vehicle connector including the charging cable).

NOTE 7 If PLC transmission power, receiver sensitivity, attenuation of cabling and EMC emission of DC charger is well aligned, the cable might be longer than 10 m.

8.3.7.4 Signal transmission path and signal measurement

This sub-clause defines a typical transmission path for the HomePlug Green PHY signal. This includes PSDs, attenuations and measurement procedures. Especially the SLAC method for measuring the signal strength needs a well-defined power level for signal transmission.

8.3.7.4.1 Typical transmission path

Figure 16 shows the transmission path for the HomePlug Green PHY signal with example values for attenuations and PSDs. All PSD and attenuation values are intended as a list of values over carrier groups, single values in the following description are given for simplification only. The attenuations are assumed as example values as follows:

— AttnRxEV is the insertion loss of the receiving path between the HomePlug Green PHY transceiver and the charge coupler on EV side. In the example, a value of 5 dB is assumed (e.g. through to transformers, lines, coupling components);

— AttnTxEV is the insertion loss of the transmitting path between the HomePlug Green PHY transceiver and the charge coupler on EV side. In the example, a value of 4 dB is assumed (e.g. through to transformers, lines, coupling components);

— AttnRxEVSE is the insertion loss of the receiving path between the HomePlug Green PHY transceiver and the charge coupler on EVSE side. In the example, a value of 3 dB is assumed (e.g. through to transformers, lines, coupling components);

— AttnTxEVSE is the insertion loss of the transmitting path between the HomePlug Green PHY transceiver and the charge coupler on EVSE side. In the example, a value of 5 dB is assumed (e.g. through to transformers, lines, coupling components);

— AttnCord is the insertion loss of the charge cord itself. In the example, a value of 2 dB is assumed.

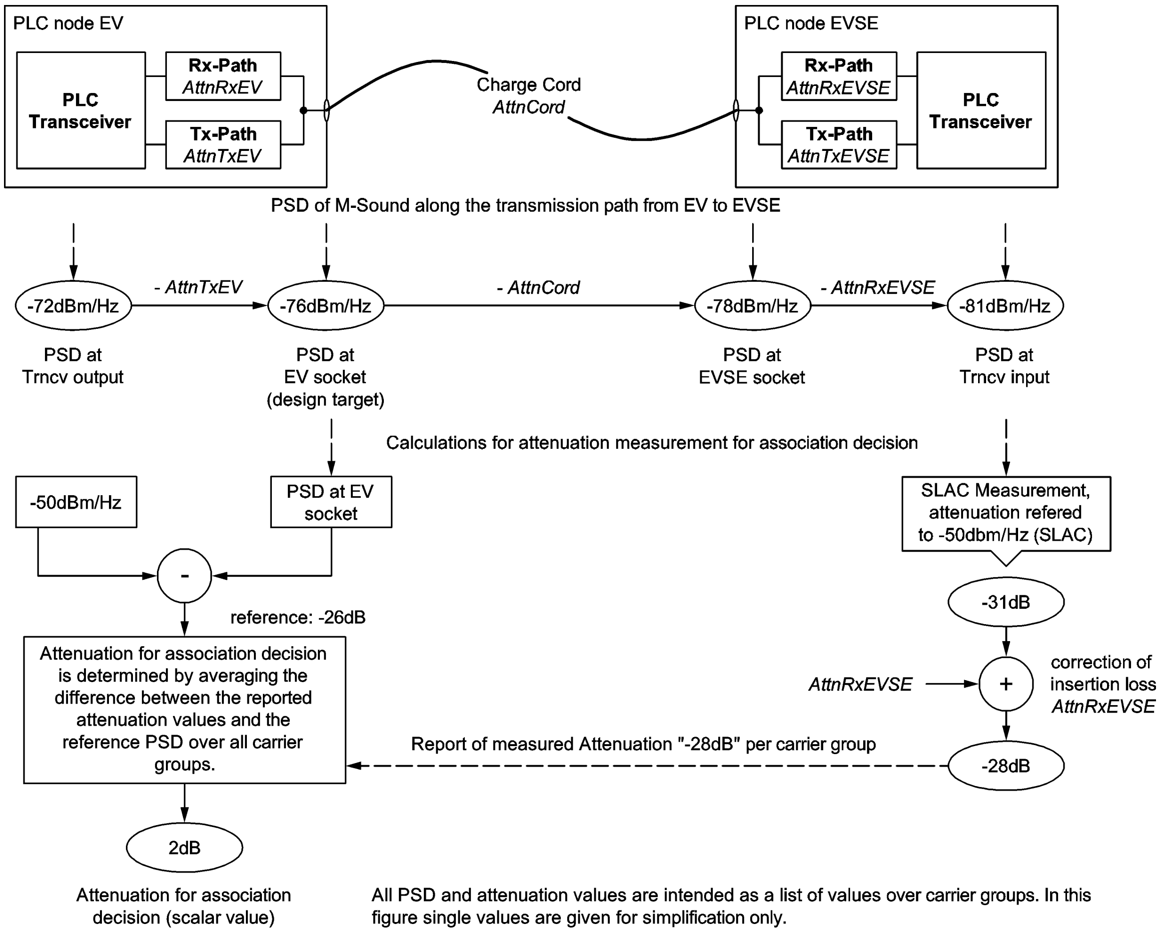


Figure 16 — HomePlug Green PHY-transmission path example with SLAC relevant attenuations and calculations

In the example, the design target is to achieve a PSD at the inlet of device of -76 dBm/Hz over the HomePlug Green PHY spectrum. Due to insertion losses in the transmission path (AttnTxEV), the output power of the transceiver (-72 dBm/Hz) should be higher to compensate the attenuation AttnTxEV. The way of measuring the PSD for the transmission path is given in 8.3.7.4.3.

The charge cord attenuates the HomePlug Green PHY signal by AttnCord, which leads in the example to a PSD at the counterpart socket-outlet of -78 dBm/Hz. Within the EVSE, the HomePlug Green PHY signal is also affected by an insertion loss (AttnRxEVSE) of the Rx path from the socket-outlet to the transceiver.

[V2G-DC-815] The attenuation of the section between EV inlet to the EV HomePlug Green PHY transceiver (AttnRxEV / AttnTxEV in Figure 16) shall be:

— 0 dB < Average over all carrier groups < 8 dB;

— 0 dB < Maximum per single carrier groups < 12 dB.

[V2G-DC-816] The Attenuation of the section between EVSE connector to the EVSE HomePlug Green PHY Transceiver (AttnRxEVSE / AttnTxEVSE in Figure 16) shall be:

— 0 dB < Average over all carrier groups < 12 dB;

— 0 dB < Maximum per single carrier groups < 14 dB.

NOTE 1 For DC charging stations the cable is always attached to the station (Case C according to IEC 61851-1).

NOTE 2 With EVSE connector the interface between the charger and the vehicle is meant. E.g. cable connector, pantograph, etc.

[V2G-DC-817] At a DC charging station with permanently attached cable (case C according to IEC 61851-1) the reference for all PSD and Attenuation values shall be the vehicle connector including the cable.

[V2G-DC-818] At a DC EV with a vehicle inlet (Case C according to IEC 61851-1) the reference for all PSD and attenuation values shall be the vehicle inlet.

NOTE 3 Cable impact can be neglected for cross talk detection with SLAC (direct connection has attenuation < 10 dB). To allow blackbox testing (without removing cables), the cable is considered if attached permanently and is not considered if removable.

NOTE 4 A charging station with permanently attached charging cable (Case C) is adjusted to a PSD value according to Table 10 (-75dBm/Hz typ.) at the vehicle connector for all used carriers.

[V2G-DC-944] The HomePlug Green PHY node of the EV shall have a receiver sensitivity of at least -105 dBm/Hz measured at EV inlet.

[V2G-DC-945] The HomePlug Green PHY node of the EVSE shall have a receiver sensitivity of at least -101 dBm/Hz measured at EVSE connector.

[V2G-DC-946] EV: At -105dBm/Hz the charging communication according to this document shall run from CM_SLAC_PARM.REQ to ContractAuthenticationRes without error or TCP re-transmission at the EV. Test to be done at EV inlet (Signal Reference).

[V2G-DC-947] EVSE: At -101dBm/Hz the charging communication according to this document shall run from CM_SLAC_PARM.REQ to ContractAuthenticationRes without error or TCP re-transmission at the EVSE during charge operation (active EVSE power modules). Test to be done at EVSE connector (signal reference).

NOTE 5 During charging (active EVSE power modules) noise might influence communication, therefore the receiver sensitivity on module (SECC/EVCC) level is higher.

NOTE 6 Requirement for Receiver Sensitivity on module (SECC/EVCC) level considers additional attenuation within EV and EVSE due to wiring.

8.3.7.4.2 Calibration and correction

Rx-Path on EVSE side:

The HomePlug Green PHY node on EV side does not know the value of AttnRxEVSE. Since this value has an impact on the SLAC measurement, the EVSE shall correct the measurement values by AttnRxEVSE before reporting the values back to the EV.

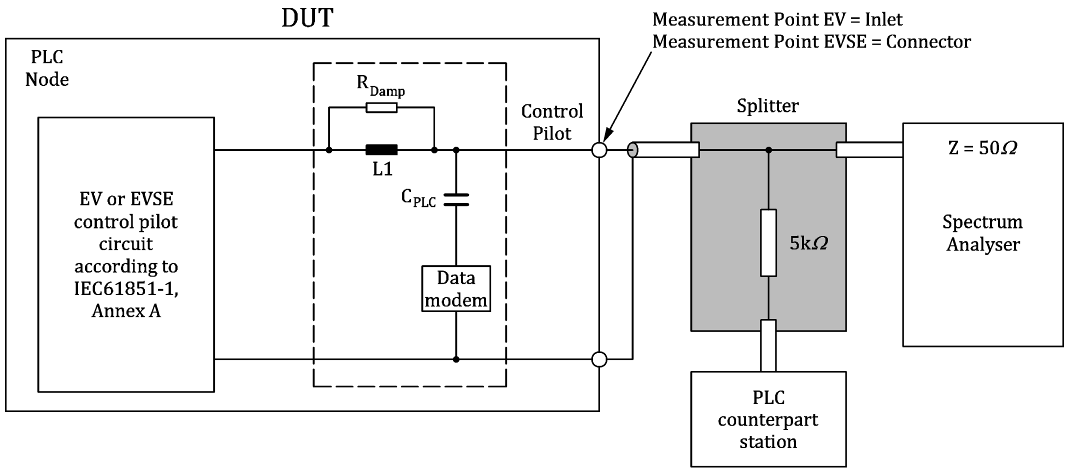
Tx-Path:

Any HomePlug Green PHY node shall comply with the transmission power given by the PSD range in Table 10. The measurement procedure to be used is given in 8.3.7.4.3

Beside the requirement for all HomePlug Green PHY nodes to comply with the defined PSD values, on EV side the exact knowledge of the PSD at the inlet is required as a reference for reported attenuation profiles from EVSEs. Any received attenuation profile from an EVSE shall be compared against the reference value given by the Tx-PSD at the inlet minus the -50 dBm/Hz reference defined for the SLAC measurement.

8.3.7.4.3 Conditions of measurement

In the following, a measurement setup (see Figure 17) and procedure is defined to determine the electrical characteristic of the transmitted signal of a V2G-device in frequency domain by means of a power spectrum density (PSD). The numeric values for the PSD is defined in Table 10 and assures a comparable signal characteristic within a certain tolerance range across V2G devices.



Key

(dashed box)	Additional components for PLC coupling
(coax symbol)	50 Ω coax cable
L1	Coil (part of the filter element)
R_{Damp}	Damping resistor
C_{PLC}	Capacitor control pilot line

Figure 17 — Measurement setup

The process defined in this section assumes a 50 Ω load between the control pilot terminal and the PE conductor. All output voltages are specified as the voltage measured at the control pilot and PE terminals of the HomePlug Green PHY node.

[V2G-DC-534] Measurements shall be made by using equipment confirming to CISPR16 specifications with a resolution bandwidth of 9 kHz.

[V2G-DC-535] A HomePlug Green PHY counterpart Station shall be connected to the device under test (DUT), to allow data communication during the measurement. The counterpart Station shall be separated by a 5 kΩ resistor to isolate the station‘s input impedance on the measurement.

[V2G-DC-536] The HomePlug Green PHY counterpart station shall comply with this document regarding coupling circuit and transmission power.

[V2G-DC-537] A spectrum analyzer with 50 Ω input impedance shall be connected to the communication line.

[V2G-DC-538] All passive components in the signal path shall be 50 Ω compliant parts and their insertion loss shall be considered.

[V2G-DC-540] During the whole measurement process, the DUT shall transmit with at least 20 % of the maximum PHY data rate.

[V2G-DC-617] For the measurement of the HomePlug Green PHY PSD, the HomePlug Green PHY node shall support an operation mode which allows to transmit/receive without control pilot signal. Otherwise, the measurement resistance of 50 Ω will force an undefined state on the control pilot.

[V2G-DC-541] The measurement shall be compliant with the following procedure:

— The input attenuator of the spectrum analyser should be set in a proper way to avoid overloading the measurement device;

— The instrument should be set to measure the peak power in a 9 kHz resolution bandwidth (dBm/9 kHz);

— Record the whole carrier band from 1.8 to 30 MHz with a hold time of at least 10 ms per sample point;

— Determine the spectrum analyser’s equivalent noise power bandwidth for the 9 kHz filter;

— Calculate the power spectrum density for the DUT by taking the values obtained in step 3 and subtracting 10 log (equivalent noise power bandwidth/1 Hz).

8.3.7.5 Parallel injection drawing example

Based on the signal and signal path specification above, the following subclause provides an implementation example for the parallel HomePlug Green PHY injection on the CPL.

The following schematic shows an implementation example based on the generic circuit in Figure 15.

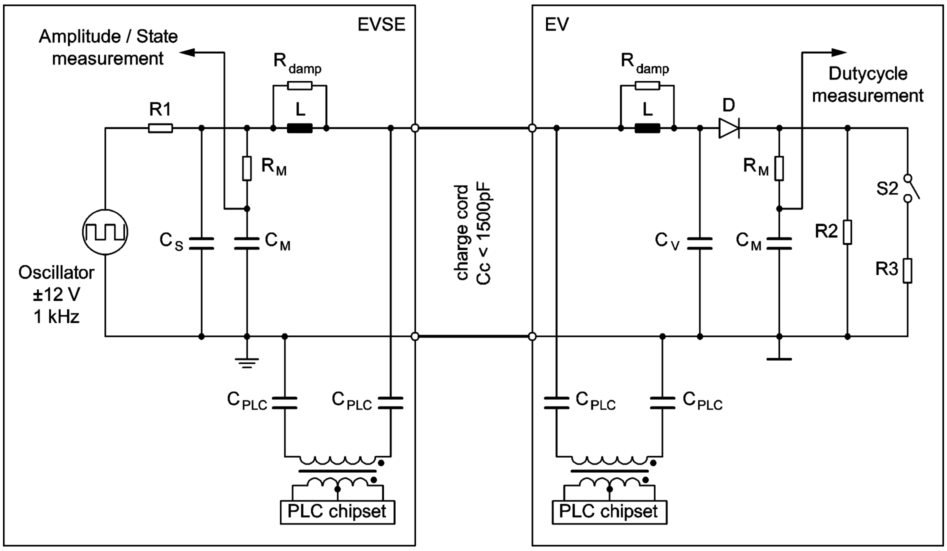


Figure 18 — Implementation example with a HomePlug Green PHY coupling transformer

For the implementation example in Figure 18, the component values defined in Table 11 should be applied.

Table 11 — Component values of HomePlug Green PHY coupling Implementation example

Component	Value
R1	IEC 61851-1
R2	IEC 61851-1
R3	IEC 61851-1

Table 11 (*continued*)

Component	Value
R_M	10 kΩ
C_M	100 pF
C_V	IEC 61851-1
C_S	IEC 61851-1
$C_{Green\ PHY}$	2.7 nF
T1/T2	Example: 3:3:3 Coupling transformer (Depends on the chips and the TX / RX band-pass filter)
D	IEC 61851-1
L	220 μH
R_{Damp}	220 Ω

8.3.7.6 Filtering requirements

8.3.7.6.1 Overview

An important source of crosstalk is the HomePlug Green PHY signal crosstalk from the CPL to the mains via the charging cord that is then conducted by mains connection to another charging cord, and crosstalk again to the other CPL (see also Figure 19). To reduce this source of crosstalk, a filter should be implemented on the mains wires, making it effective on all such linked CPLs.

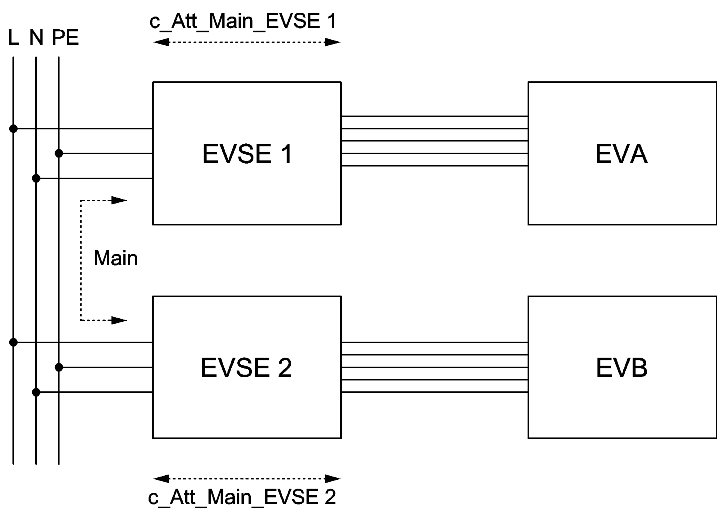


Figure 19 — Overview sources of crosstalk

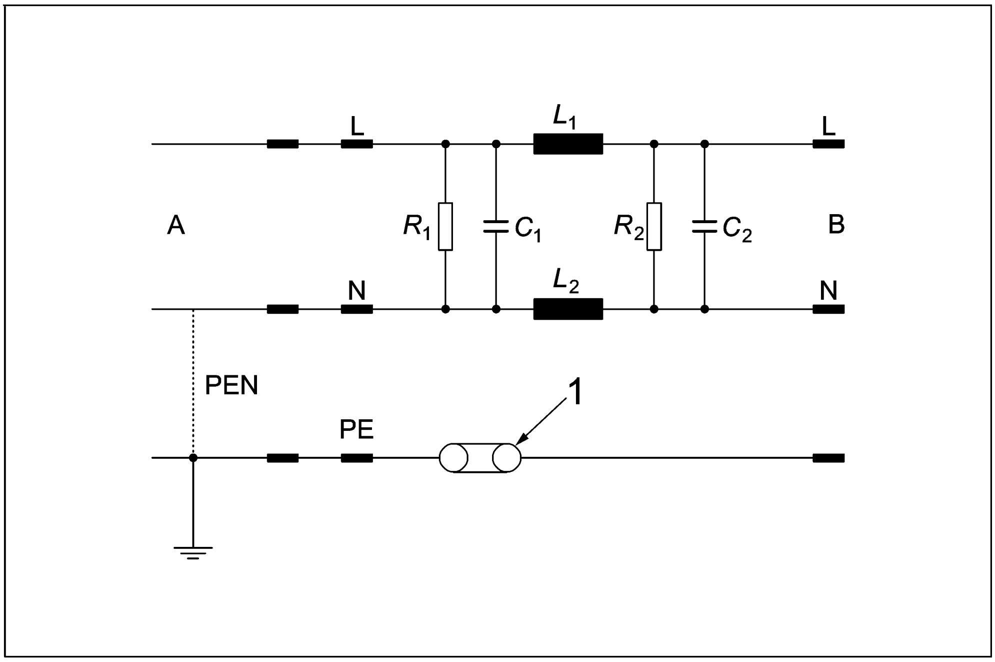
[V2G-DC-088] When designing EVSE and EV systems, the operational frequency band of the HomePlug Green PHY technology (approximately 2 Mhz – 30 MHz) shall be considered. Filters should only affect the used frequency band.

A ratio of 70 dB can be targeted between the wanted and the crosstalked signal on CPLs, especially to guarantee the SLAC function. The following requirement is based on the assumption of a 20 dB amplitude decrease when the HomePlug Green PHY signal crosstalk inside each charging cord.

It is highly recommended that the system has an attenuation better than 30 dB between the mains supplying the EVSE and the CPL in the charging cable, for the whole HomePlug Green PHY spectrum.

8.3.7.6.2 Example of crosstalk filter implementation

Figure 20 gives an example of a valid filter, in the case of crosstalking HomePlug Green PHY signal to the mains lines, including the PE. It can be improved with additional serial inductances on mains' side, to avoid excessive filtering of other HomePlug Green PHY networks on their side.



Key

1	Ferrite bead
A	To EVSE 1
B	To EVSE 2
R_1, R_2	See text below
C_1, C_2	100 nF
L_1, L_2	2 µH

Figure 20 — Example of valid crosstalk filter in the case of HPGP signals

The resistors R_1 *and* R_2 are intended to discharge the capacitor and prevent electrical shocks.

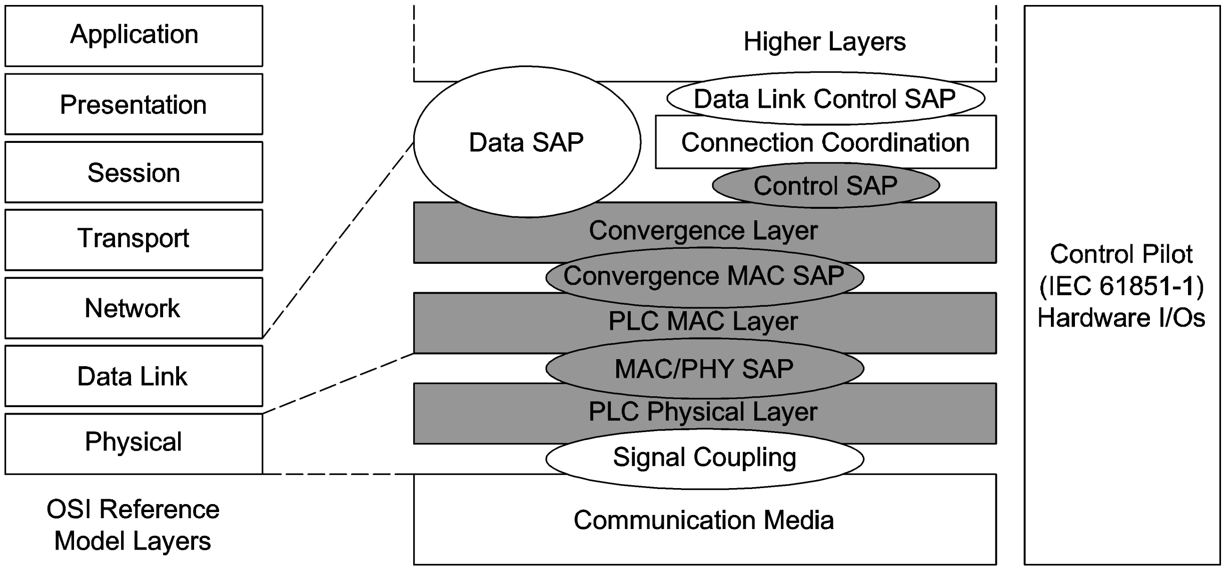
8.3.8 Layer 2 interfaces

8.3.8.1 Overview

As shown in Figure 21, the data link layer provides interfaces to higher layers:

— The Data SAP is the interface between the HomePlug Green PHY technology and the layer 3 (e.g. IPv6);

— The data link control SAP provides link status information, error information, control functionality and is located between the connection coordination and higher layers.



Key

Covered by communication technology specification

Figure 21 —OSI Layer 1-2 overview

8.3.8.2 Data Link Control SAP to layer 3

The data link control service access primitives provide methods for indicating the HomePlug Green PHY Link status, the status of matching and error messages to higher layers.

[V2G-DC-542] The D-LINK_READY.indication shall inform higher layers about a change of the HomePlug Green PHY link status. This indication shall be sent upon any change of the link status. The values of the DLINKSTATUS are independent of the states of the control pilot.

Primitive	**D-LINK_READY.indication**
Entity to support	**EV, EVSE**
Parameter Name	**Description**
DLINKSTATUS	Status of HomePlug Green PHY communication link — no link — Link established

[V2G-DC-543] The D-LINK_TERMINATE.request shall force lower layers to terminate the data link.

Primitive	**D-LINK_TERMINATE.request**
Entity to support	**EV, EVSE**

8.3.8.3 Data SAP

The Data SAP is the ETH SAP. The ETH SAP is completely covered by the HomePlug Green PHY specification.

Due to the convergence layer, the Data Link Layer of HomePlug Green PHY provides an Ethernet II-class SAP to higher layers.

This SAP supports applications using Ethernet II class packets, including IEEE 802.3 with or without IEEE 802.2 (BS), IEEE 802.1H (SNAO) extensions, and/or VLAN tagging.

The Ethernet II-class SAP includes following service primitives, defined in the HomePlug Green PHY specification:

— ETH_SEND.REQ;

— ETH_SEND.CNF;

— ETH_RECEIVE.IND.

8.3.9 Sleep and Wake-up Handling

[V2G-DC-808] To wake-up the EV, the pilot function controller on EVSE side shall be configured to the last known parameter set NMK and NID and the oscillator shall perform a B1 -> B2 transition.

[V2G-DC-809] The EVSE shall support a PLC connection based on either the last known parameter set NMK and NID or starting with a new SLAC process, while the EV is plugged in.

[V2G-DC-810] In case of an error, the EVSE shall only support a new SLAC process for starting a new PLC link.

[V2G-DC-811] The EV shall store the NMK and NID after stopping a charging session with SessionStopReq/Res and no error occurred until unplug if it wants to resume without a new SLAC process.

[V2G-DC-812] If the EV is not able to establish a PLC link using the last known parameter set NMK and NID it shall start a SLAC process.

NOTE 1 State B1 is 100 ms minimum and B2 is 1 s minimum.

The EVSE should limit the number of Wake-up cycles to the EV to 6 before control pilot state B1 is applied. After control pilot state A, another 6 wake-up cycles might be performed.

[V2G-DC-813] After waking up, the PLC node on EV side shall be configured to the last known logical parameter set of NMK and NID.

[V2G-DC-814] If the EV is asleep, it shall wake up by detecting a control pilot state B1/B2 transition and shall configure the communication module to the last known logical network parameter of NMK and NID.

The EV can use the old NMKw and NID or redo SLAC. The EVSE should be able to accept both behaviors of the EV.

NOTE 2 In contrast to ISO 15118, DIN/TS 70121 does not reuse the old SessionID for V2G session handling.

8.4 V2G communication states

This subclause describes the basic states of the communication between EVCC and SECC. The timer and timeout values used in this subclause are described in 9.6.

8.4.1 V2G communication of the EVCC

Figure 22 depicts the general communication states of the V2G communication from an EVCC perspective.

[V2G-DC-096] After the data link layer connection is established, the EVCC shall initiate the address assignment mechanism as defined in 8.5.3.2 and 8.5.3.3.

NOTE 1 In this document this is described by D-LINK_READY.indication(DLINKSTATUS = Link established).

[V2G-DC-097] After the Application Layer requests the start of a V2G communication session, the EVCC shall initiate the IP address assignment mechanism as defined in 8.5.3.2 and 8.5.3.3.

[V2G-DC-098] The EVCC shall process the IP address assignment mechanism as defined in 8.5.3.2 and 8.5.3.3.

[V2G-DC-100] After a link-local IP address is assigned, the EVCC shall perform the SECC discovery as defined in 8.9.3.

NOTE 2 In this document, this is described by N-IP_Address.indication(N_IP_STATUS = Link local address assigned).

[V2G-DC-101] The EVCC shall process the SDP according to 8.9.3.

[V2G-DC-102] The EVCC shall stop the SDP when V2G_EVCC_CommunicationSetup_Timer is equal or larger than V2G_EVCC_CommunicationSetup_Timeout.

[V2G-DC-103] After the SECC IP address is discovered, the EVCC shall establish the TCP connection to the SECC as described in 8.6.1.

NOTE 3 In this document, this is described by N-SECC_Address.indication(N_SECC_STATUS = SECC IP-address discovered).

[V2G-DC-105] The EVCC shall stop to attempt to establish a TCP connection when V2G_EVCC_CommunicationSetup_Timer is equal or larger than V2G_EVCC_CommunicationSetup_Timeout.

[V2G-DC-106] Once the TCP connection is established, the EVCC shall initiate the V2G communication session as defined in clause 9.

[V2G-DC-107] The EVCC shall stop the communication and terminate the TCP connection after the application layer requests to stop the TCP connection.

[V2G-DC-936] After receiving the SessionStop message, the EVCC shall terminate the TCP connection within <= 4s.

[V2G-DC-939] In case of any error detection or HLE request the TCP connection shall be terminated by the EVCC without any delay max. within 5 s.

NOTE 4 This max. time applies for example to [V2G-DC-975]

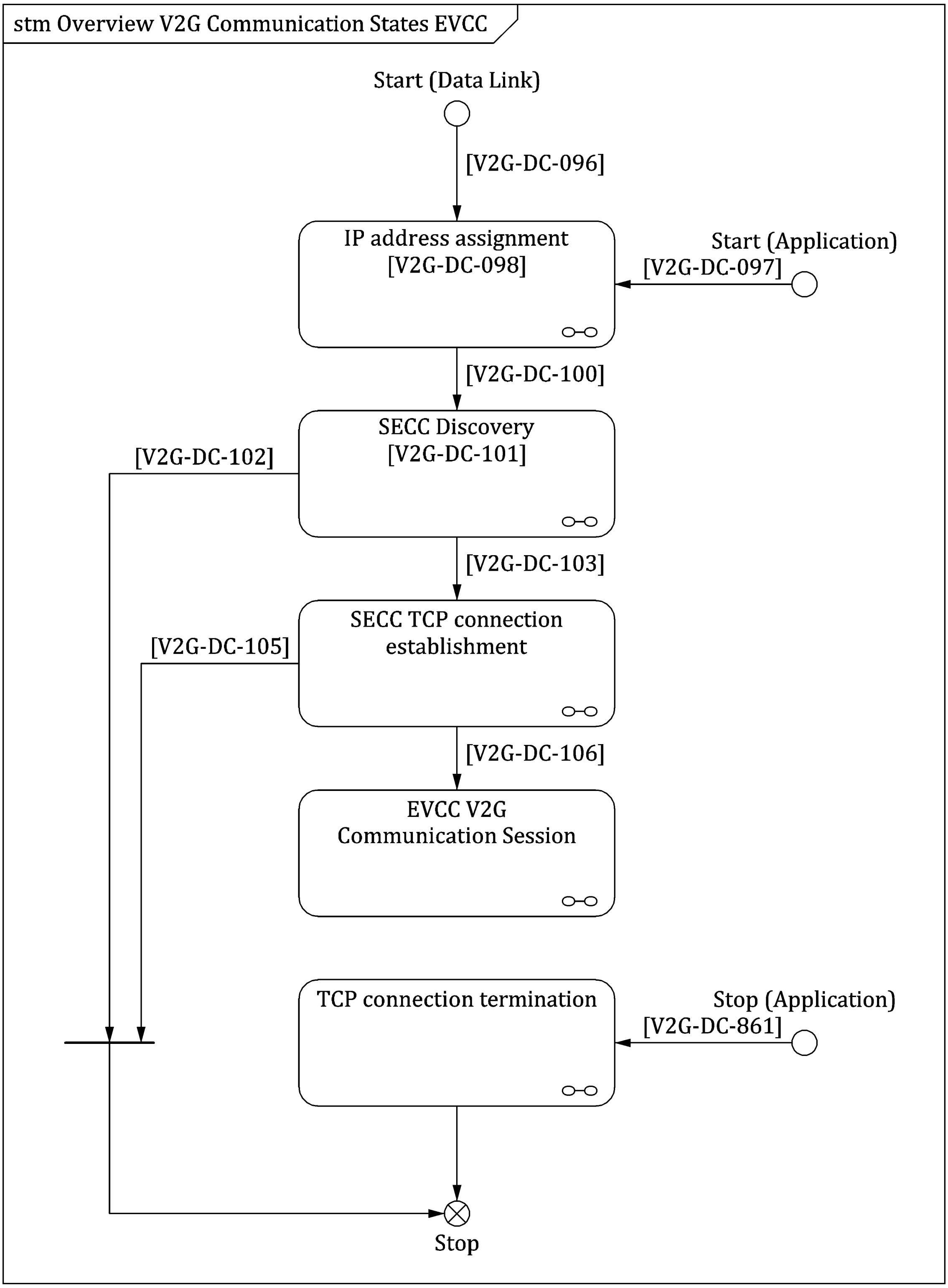


Figure 22 — Overview V2G communication states EVCC

8.4.2 V2G communication of the SECC

Figure 23 depicts the general communication states of the V2G communication from an SECC perspective.

[V2G-DC-108] The SECC shall configure an IP address (static or dynamic) by any appropriate mechanism.

[V2G-DC-109] The SECC discovery service shall be running when the SECC enters CP State B.

[V2G-DC-110] The SECC discovery service shall be updated after an IP address for the SECC is changed.

NOTE 1 It is not required that the SECC discovery service is implemented in the SECC directly. It is also possible to have a separate unit providing the SECC discovery service.

[V2G-DC-112] After the IP address is configured, the SECC shall wait for a TCP connection on the IP address that is distributed by the SECC discovery service.

NOTE 2 In this document, this is described by N-IP_Address.indication(N_IP_STATUS = Link local address assigned) or N-IP_Address.indication(N_IP_STATUS = Global address assigned).

[V2G-DC-113] As long as a TCP connection is not yet established, the SECC shall wait for a TCP connection at least until the SECC enters CP State A.

[V2G-DC-115] After the TCP connection is established, the SECC shall wait for the initialization of the V2G communication session as defined in Clause 9.

[V2G-DC-116] The SECC shall stop the communication and terminate the TCP connection after the application layer requests to stop the TCP connection.

[V2G-DC-937] After sending the SessionStopRes message, the SECC shall wait for the termination of the TCP connection by the EVCC for 5 s.

[V2G-DC-938] If the EVCC did not terminate the TCP connection after 5 s, the SECC shall terminate the connection after 1 s latest.

[V2G-DC-940] In case of any error detection or HLE request the TCP connection shall be terminated by the SECC without any delay max. within 5 s.

This max. time applies for example to [V2G-DC-986] If the SECC receives a request message that it expects according to the message sequence specified in this subclause, and if the SECC cannot process this request message, e. g. due to errors in the message parameters or due to impeding conditions in the EVSE, the SECC shall proceed as if the SECC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-663].

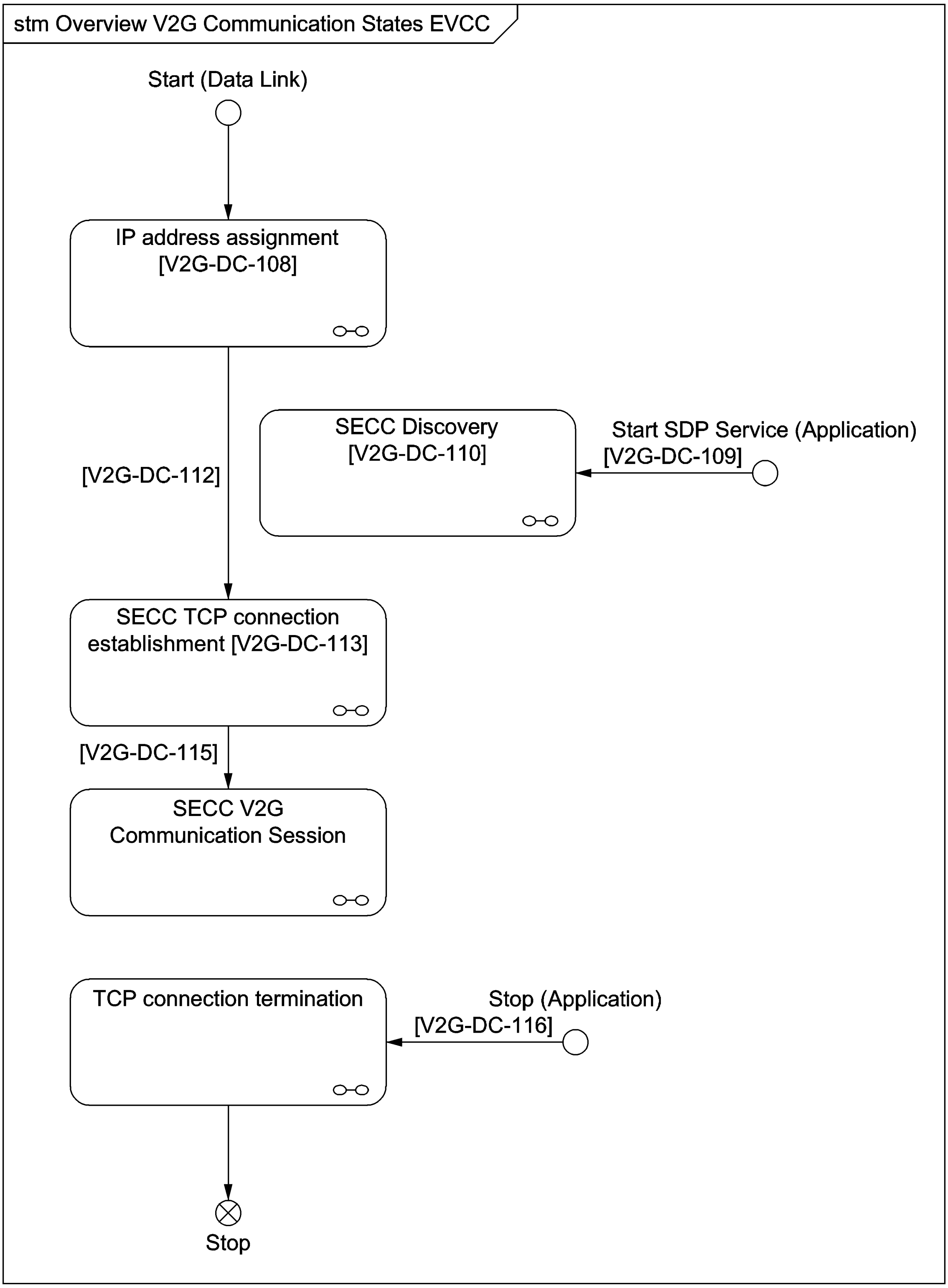


Figure 23 — Overview V2G communication states SECC

8.5 Network layer

8.5.1 General

The protocol is based on the internet protocol standard IPv6 (see IETF RFC 2460). The internet protocol is datagram based; unreliable and located on the network layer according to the OSI layered architecture model. IP is the first transmission medium independent protocol.

The process of how a node will acquire an IP address is described in subclause 8.5.3.

8.5.2 Applicable RFCs and limitations and protocol parameter settings

[V2G-DC-117] All V2G entities shall implement all requirements included in the following IETF RFCs if not explicitly excluded.

8.5.2.1 IPv6

[V2G-DC-118] All V2G entities shall implement IPv6 according IETF RFC 2460.

For EVCC side, only node requirements, and for SECC side, host requirements are applicable. Router functionality is only an optional requirement for SECC side.

[V2G-DC-119] All V2G entities shall skip the following IPv6 extension headers: Hop-by-hop options header, routing header (EVCC), and destination options header.

[V2G-DC-120] All V2G entities shall fulfil the IPv6 node requirements specified in IETF RFC 4294.

[V2G-DC-121] All V2G entities shall not implement IPsec, required as requested in IETF RFC 4294, chapter 8.

[V2G-DC-122] All V2G entities shall implement IETF RFC 5095, which updates IETF RFC 2460 and IETF RFC 4294.

The IANA allocation guidelines for the routing type field in the IPv6 routing header are described in [1] IETF RFC 5871. It is recommended to adhere to these guidelines.

[V2G-DC-123] All V2G entities shall implement path MTU discovery according to IETF RFC 1981.

[V2G-DC-124] All V2G entities shall support handling of overlapping IP fragments according to IETF RFC 5722.

[V2G-DC-125] All V2G entities shall implement IETF RFC 5220, which extends IETF RFC 2460.

The following general requirements apply to the handling of IP packets addressed to a V2GTP entity:

[V2G-DC-126] Any packets with a multi- or broadcast address as source IP address shall be ignored.

8.5.2.2 Neighbour discovery (ND)

Primary and secondary actors use IPv6 stateless address auto configuration for generating addresses for their interfaces. All interfaces have a link-local address. To ensure unique addresses and to support global addresses the neighbour broadcast protocol is used.

[V2G-DC-127] Each V2G entity shall implement ND as defined in IETF RFC 4861.

[V2G-DC-128] The EVCC and the SECC shall implement IETF RFC 4429 to allow assignment of IP address before ND has finished its work.

8.5.2.3 Internet control message protocol (ICMP)

The internet control message protocol (ICMP) is part of the Internet protocol suite and is used for sending error messages, e.g. to indicate that a requested service is not available or that a host could not be reached. Consequently, ICMP is a mandatory part of an IP stack implementation and is located on the network layer according to the OSI layered architecture model.

[V2G-DC-129] Each V2G entity shall implement ICMPv6 as specified in IETF RFC 4443.

[V2G-DC-130] ICMPv6 message types defined in IETF RFC 4443 shall only be implemented if included in Table 12.

[V2G-DC-133] Each V2G entity shall implement the IETF RFCs referred to in column “Reference” of Table 12 describing the implementation details for the respective ICMP message type.

Table 12 — Mandatory ICMP message set

ICMP message ID	ICMP message name	Reference
1	Destination unreachable	IETF RFC 4443
2	Packet too big	IETF RFC 4443
3	Time exceeded	IETF RFC 4443
4	Parameter problem	IETF RFC 4443
128	Echo request	IETF RFC 4443
129	Echo reply	IETF RFC 4443
133	Router solicitation	IETF RFC 4861
134	Router advertisement	IETF RFC 4861
135	Neighbour solicitation	IETF RFC 4861
136	Neighbour advertisement	IETF RFC 4861
137	Redirect message	IETF RFC 4861
141	Inverse neighbour discovery solicitation message	IETF RFC 3122
142	Inverse neighbour discovery advertisement message	IETF RFC 3122

8.5.3 IP addressing

8.5.3.1 General

This section specifies how an EV retrieves valid IP addresses to communicate over an IP-based network. The following addresses are considered in this standard:

— EV’s own IP address;

— IP address for connecting to the SECC.

NOTE An IPv6 host usually has multiple IP addresses assigned to one physical network interface. (e.g. link-local, site-local and global address; if there are multiple routers connected to a local link the host has even several global addresses).

8.5.3.2 Stateless auto address configuration (SLAAC)

[V2G-DC-134] For IPv6 each V2G entity shall support the configuration of a link-local IPv6 unicast address as specified in IETF RFC 4291.

[V2G-DC-135] The interface ID of the link-local address of a V2G entity shall be generated from its IEEE 48 bit MAC identifier according to the definition in IETF RFC 4291.

[V2G-DC-136] If present in the network, an IPv6 address shall be derived from the router advertisement messages according to IETF RFC 4862.

8.5.3.3 Address selection

[V2G-DC-137] The IPv6 default address selection shall be performed according to IETF RFC 3484.

8.5.4 Network Layer service primitive — N–IP_Address.indication

The N-IP_Address.indication notifies about the status of the IP address assignment. Table 13 describes the service primitive and its parameter(s).

Table 13 — N-IP_Address.indication service primitive

Primitive name	**N-IP_Address.indication**
Entity to support	EVCC, SECC
Parameter Name	Description
N_IP_STATUS	— Link local address assigned — Global address assigned — Error

N-IP_Address.indication (N_IP_STATUS = Link local address assigned) indicates the assignment of a local IP address.

N-IP_Address.indication (N_IP_STATUS= Global address assigned) indicates the assignment of a global IP address.

N-IP_Address.indication (N_IP_STATUS= Error) indicates any detected error during IP assignment.

8.5.5 SECC discovery

[V2G-DC-138] The EV shall support SECC discovery according to 8.9.3.

8.6 Transport Layer

8.6.1 Transmission control protocol (TCP)

8.6.1.1 Overview

The transmission control protocol (TCP) allows applications of V2G entities to establish a reliable data connection to other entities. To exchange in a reliable way and in-order delivery of sender to receiver data. Additionally, TCP provides flow control and congestion control and provides for various algorithms to handle congestion and influence flow control.

8.6.1.2 Applicable RFCs, limitations and protocol parameter settings

[V2G-DC-139] Each V2G entity shall implement TCP as specified in IETF RFC 793.

8.6.1.3 TCP Performance and checksum recommendations and requirements

The following requirements define TCP implementation details concerning congestion control, retransmission, timing, initial window size and selective acknowledgement for the purpose of improving the overall performance of TCP.

It is recommended to use the following congestion control and re-transmission algorithms in addition to the standard TCP methods:

Each V2G entity should implement TCP congestion control according to IETF RFC 5681.

Each V2G entity should implement the NewReno modification to TCP's fast recovery algorithm according to IETF RFC 5682.

Each V2G entity should compute TCP's retransmission timer according to IETF RFC 6298.

To increase TCP's performance, each V2G entity should implement TCP extensions for high performance according to IETF RFC 1323.

Each V2G entity should support TCP selective acknowledgment options according to IETF RFC 2018.

Each V2G entity should implement the user timeout option according to IETF RFC 5482.

[V2G-DC-146] The urgent pointer for TCP shall not be used by any V2G entity.

It is recommended to use the following checksum algorithm:

The checksum fields required in TCP headers should be implemented according to IETF RFC 1624.

[V2G-DC-953] Whenever a TCP retransmission timeout (RTO) is computed in a V2G entity, the RTO shall be lower than 400 ms.

[V2G-DC-954] Whenever a TCP retransmission timeout (RTO) is computed in a V2G entity, the RTO should be in the range of 50 ms (lower bound) and 110 ms (upper bound).

NOTE While [V2G-DC-953] defines the maximum upper bound allowed by this document, [V2G-DC-954] provides a recommendation.

8.6.2 User datagram protocol (UDP)

8.6.2.1 Overview

The user datagram protocol (UDP) is a connectionless protocol. UDP does not provide the reliability and ordering guarantees that TCP does. Packets can arrive out of order or can be lost without notification of the sender or receiver. However, UDP is faster and more efficient for many lightweight or time-sensitive purposes. UDP is located on the transport layer of the OSI layered architecture model.

8.6.2.2 Applicable RFCs, limitations and protocol parameter settings

[V2G-DC-148] Each V2G entities shall implement user datagram protocol according to IETF RFC 768.

8.7 V2G transfer protocol

8.7.1 General information

The V2G transfer protocol (V2GTP) is a compact communication protocol to transfer V2G messages between two V2GTP entities. It mainly consists of a header and payload definition that allows separating and processing

V2G messages efficiently. V2GTP is the standard transfer protocol between the EVCC and SECC but may also be used for communication with other V2G entities that support the V2GTP protocol.

8.7.2 Supported ports

V2GTP is based on TCP and UDP. In both transport protocols, a pair of IP addresses (source address and destination address) and a pair of port numbers (source port and destination port) are used to establish and identify a connection for bidirectional data exchange. The TCP connection is established from the source address and source port to the destination address and destination port. UDP is not connection-based; datagrams are exchanged between the source address and source port and the destination address and destination port. The ports listed in Table 14 are used by V2GTP entities.

Table 14 — Supported ports for V2GTP

Name	Protocol	Port number	Description
V2G_SRC_TCP_DATA	TCP (unicast)	Port number in the range of Dynamic Ports (49152-65535) as defined in IETF RFC 6335.	V2GTP source port at a primary actor (e.g. EVCC) that implements the V2GTP protocol.
V2G_DST_TCP_DATA	TCP (unicast)	Port number at V2GTP entity providing a V2GTP destination port. For SECC it will be dynamically assigned by the SDP mechanism in the range of dynamic ports (49152-65535) as defined in IETF RFC 6335	V2GTP destination port at a primary actor (e.g. SECC)
V2G_UDP_SDP_CLIENT	UDP (unicast)	Port number in the range of dynamic ports (49152-65535) as defined in IETF RFC 6335,	SDP client source port at the EVCC
V2G_UDP_SDP_SERVER	UDP (multicast)	15118	SDP server port which accepts UDP packets with a local-link IP multicast destination address.

For V2GTP entities implementing the V2GTP the following general requirements apply:

[V2G-DC-149] A V2GTP entity providing a destination port shall support at least one connection on the local port V2G_DST_TCP_DATA as defined in Table 14.

[V2G-DC-151] A V2GTP entity using a source port shall support at least one connection on the local port V2G_SRC_TCP_DATA as defined in Table 14.

Especially, for an EVCC and an SECC the following applies:

[V2G-DC-153] The EVCC shall use a source port V2G_SRC_TCP_DATA as defined in Table 14.

[V2G-DC-154] The SECC shall provide a destination port V2G_DST_TCP_DATA as defined in Table 14.

[V2G-DC-155] The EVCC shall support at least one TCP connection on port V2G_SRC_TCP_DATA.

[V2G-DC-156] The SECC shall support at least one TCP connection on port V2G_DST_TCP_DATA.

[V2G-DC-157] The EVCC shall use the port V2G_DST_TCP_DATA returned in the last SECC discovery response message for connecting the SECC.

[V2G-DC-158] An SDP client shall support the port V2G_UDP_SDP_CLIENT as defined in Table 14 for sending and receiving SDP messages.

[V2G-DC-159] An SDP server shall support the port V2G_UDP_SDP_SERVER as defined in Table 14 for receiving and sending SDP messages.

Depending on the implementation of the EVCC the dynamically assigned V2G_UDP_SDP_CLIENT port will be assigned once during or before the first transmission of a UDP packet to a SECC or can be dynamically re-assigned for each individual UDP request message and response. Also depending on whether messages are repeatedly sent, response messages may arrive asynchronously and may not be associated to the exact corresponding request anymore.

[V2G-DC-160] The SDP client shall be able to handle asynchronously arriving SECC discovery response messages.

8.7.3 Protocol data unit

8.7.3.1 Structure

The V2GTP PDU consists of a header and a body section as shown in Figure 24.

Figure 24 — V2GTP message structure

The payload contains the application data (e.g. a V2G message). The header separates the payloads (i.e. individual V2GTP messages) within a byte stream and gives information for the payload processing.

The V2GTP message header structure is shown in Figure 25 and described in Table 15. The supported payload types are described in Table 16.

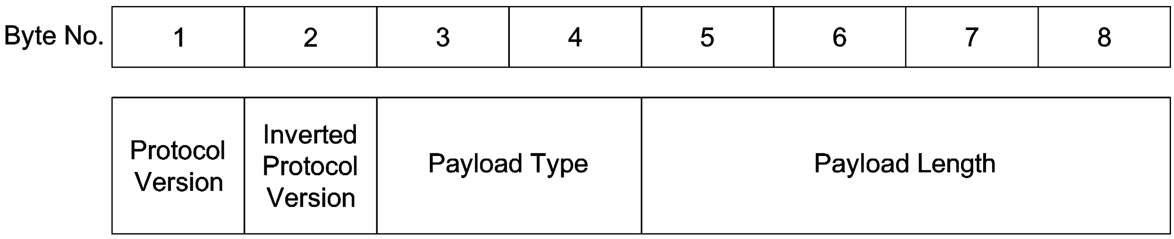


Figure 25 — V2GTP message header structure

[V2G-DC-161] A V2GTP entity shall use the header structure as shown in Figure 24.

[V2G-DC-162] A V2GTP entity shall send the 8 bytes of the V2GTP header in the order as shown in Figure 25. A byte with a lower number shall be sent before a byte with a higher number. The header starts with byte number 1 and ends with byte number 8.

[V2G-DC-163] A V2GTP entity shall send the fields “payload type” and “payload length” in big endian format: The most significant byte is sent first the least significant byte is sent last.

Table 15 — Generic V2GTP header structure

Type	Description	Value
Protocol Version	Identifies the protocol version of V2GTP messages.	0x01: V2GTP version 1 0x00, 0x02-0xFF reserved by document
Inverse Protocol Version	Contains the bit-wise inverse value of the protocol version which is used in conjunction with the V2GTP protocol version as a protocol verification pattern to ensure that a correctly formatted V2GTP message is received. Equals the <Protocol_Version> XOR 0xFF	0xFE: V2GTP Version 1
Payload type (GH_PT)	Contains information about how to decode the payload following the V2GTP header.	Refer to Table 16 for a complete list of currently specified payload type values.
Payload length (GH_PL)	Contains the length of the V2GTP message payload in bytes (i.e. excluding the generic V2GTP header bytes).	0x00...0xFFFFFFFF

Table 16 — Overview on V2GTP payload types

Payload type value	Payload type name	Specified in subclause
0x0000 — 0x8000	Reserved	Not applicable
0x8001	EXI encoded V2G Message	Clause 9
0x8002 — 0x8FFF	Reserved	Not applicable
0x9000	SDP request message	8.9.3.4
0x9001	SDP response message	8.9.3.5
0x9002 — 0x9FFF	Reserved	Not applicable
0xA000 — 0xFFFF	Manufacturer specific use	Not applicable

[V2G-DC-164] A V2GTP entity shall use the V2GTP message structure as shown in Figure 24 to send V2G messages as defined in Clause 9.

[V2G-DC-165] A V2GTP entity shall use the definitions as defined in Table 15 and Table 16.

[V2G-DC-166] For EXI encoded V2G messages (payload 0x8001) a V2GTP entity shall use a separate V2GTP message for each V2G message.

NOTE Requirement [V2G-DC-166] implies that the payload field can include neither a part of a message nor multiple messages.

8.7.3.2 Header processing

For the processing of the payload the V2GTP entity processes the header first. For this, a V2GTP entity that receives a V2GTP message checks the header field step by step. The header processing as defined before is shown in Figure 26.

[V2G-DC-167] A V2GTP entity shall process the V2GTP header as defined in Table 15 before processing the payload as defined in Table 16.

[V2G-DC-168] A V2GTP entity shall check the protocol version and inverse protocol version fields (synchronization pattern) before any other header fields.

[V2G-DC-169] A V2GTP entity shall close the TCP connection if it detects an error in the version and inverse version field as defined in Table 15.

[V2G-DC-170] A V2GTP entity shall check the payload type after the successful check of the version and inverse version field.

[V2G-DC-171] A V2GTP entity shall skip the payload following the header if the payload type is not supported.

[V2G-DC-172] A V2GTP entity shall check the payload length after the successful check of the payload type.

[V2G-DC-173] A V2GTP entity shall skip the payload bytes after a header and close the TCP connection if the payload size cannot be processed.

[V2G-DC-174] If the header processing was successful, the V2GTP entity shall process the payload.

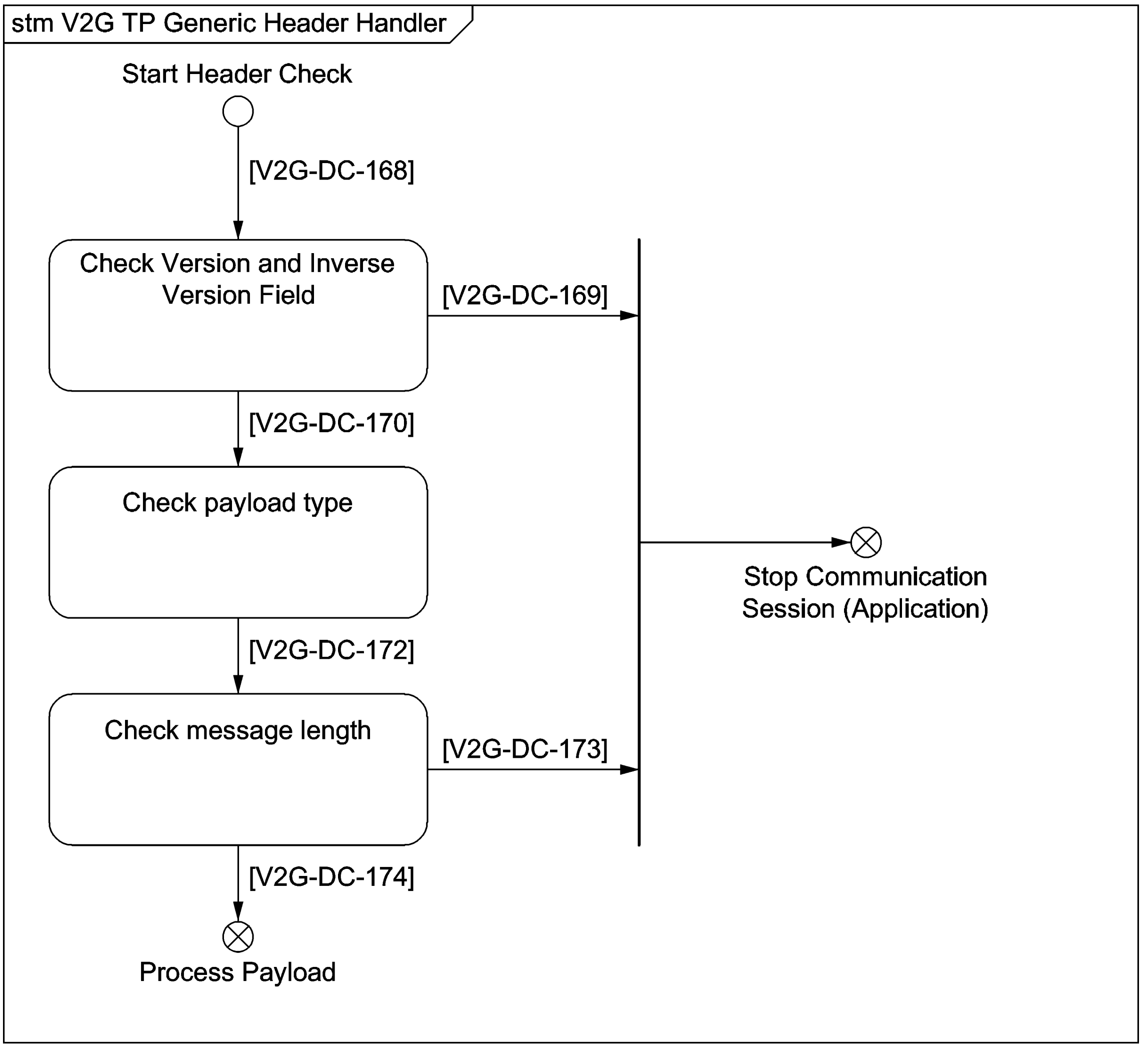


Figure 26 — V2GTP generic header handler

8.8 Presentation layer

8.8.1 XML and efficient XML interchange (EXI)

8.8.1.1 Overview

To describe the V2G message set the presentation layer uses the widely adopted XML data representation. For this purpose, this document defines messages (i.e. data structures and data types) based on XML Schema which allows the type aware use of XML and enables simplified validity evaluation of exchanged messages.

[V2G-DC-480] When transmitting V2G messages defined in this document by using XML, all V2G entities shall use an encoding format according to definitions in W3C EXI 1.0.

8.8.1.2 Efficient XML interchange

The efficient XML interchange (EXI) format allows to use and process XML-based messages on a binary level. Thus, the EXI format increases the processing speed of XML-based data as well as reduces the memory usage. Basically, EXI is a W3C recommendation. The EXI format uses a relatively simple grammar driven approach that achieves very efficient encodings for a broad range of use cases. It is not uncommon for EXI messages to be up to 100 times smaller than equivalent XML documents. The EXI specification describes in a predefined process how schema information are transformed into EXI grammar. The reason for doing so is that EXI grammar is much simpler to process, compared to XML Schema information. Nevertheless, the parsing can be performed in the same accurate way as it is possible in XML.

[V2G-DC-487] There are different kinds of coding mechanism with EXI. To meet the demands in terms of efficient processing, less resources usage, message size, and message extendibility, non-strict schema-aware settings shall be used.

In general, EXI streams can be created in a very efficient way if all encoded information (elements/attributes) are defined by an underlying XML schema (*Schema-informed Grammars*). Deviant information based on XML schema knowledge is encoded in a more generic way. The EXI coder encodes the qualified names (namespace and element/attribute name) of such unknown information in a string-based manner. However, simple types of the schema deviations may be still encoded type-aware.

EXI decoders can decode the efficient EXI streams by using the same underlying XML schema which was used for the encoding process. Schema deviations are recognized in the EXI stream. These deviations (unknown elements or attributes) can be either processed or skipped.

Figure 27 summarizes the efficient XML interchange concept for the DIN/TS 70121 domain. Due to the high limited resource restriction, the EV may only be able to handle the XML-based data using a corresponding data structure representation. Such data structures can be used to serialize or de-serialize DIN/TS 70121 application messages. Meanwhile, the EVSE may be able to handle the data as data structure and/or the more resource intensive document object style (DOM) or in a traditional plain-text XML variant.

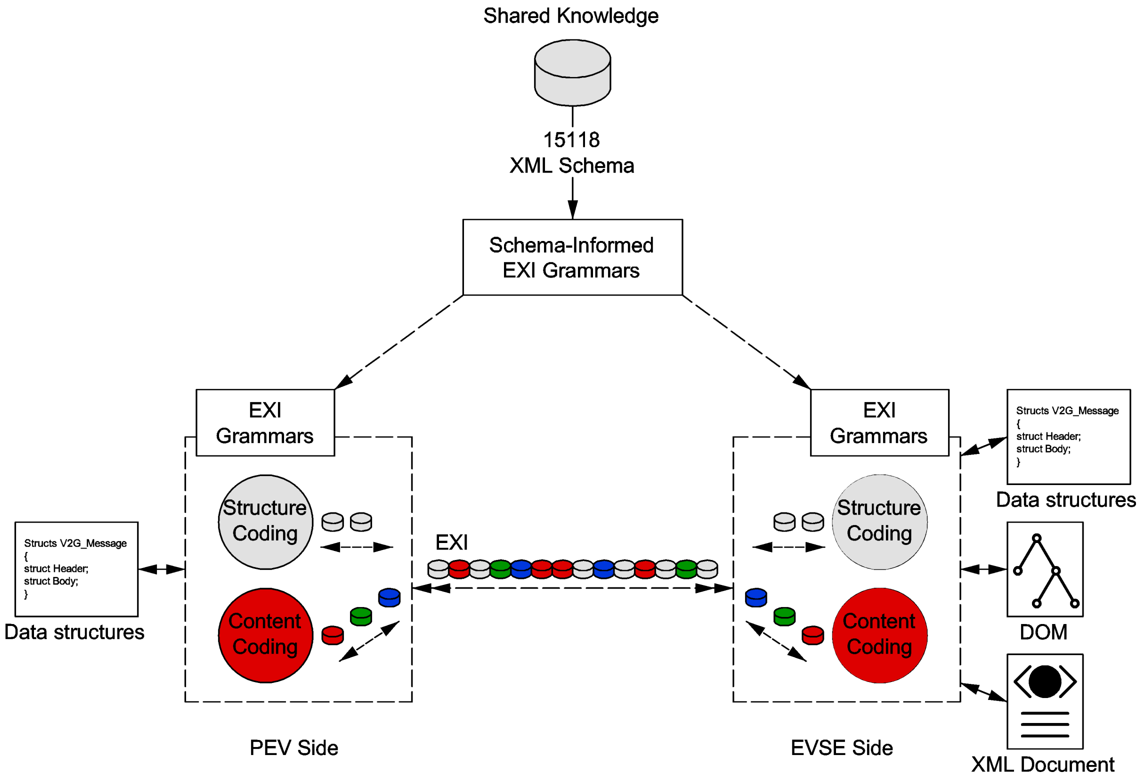


Figure 27 — Basic concept of EXI

If the labelling “XML” or “XML-based” is used in this document, this is equivalent to “binary XML”.

8.8.1.3 EXI Settings for DIN TS/70121 application messages

The following EXI settings shall be used for the EXI-based V2G communication.

[V2G-DC-963] The XML Schema with the namespace "urn:din:70121:2012:MsgDef" that represents this DIN SPEC 70121 version 2.1 (major version "2", minor version "1") shall be used for encoding and decoding EXI streams.

[V2G-DC-176] The EXI coder for encoding and decoding of DIN/TS 70121 communication shall use the default EXI coding options (see http://www.w3.org/TR/exi/#exiOptionsInOptionsField) with the exception of the following options.

Table 17 — Excluded EXI coding options

EXI Option	Description	DIN/TS 70121 value
valuePartitionCapacity	Specifies the total capacity of value partitions in a string table	0

[V2G-DC-177] The EXI header (http://www.w3.org/TR/exi/#header) shall be used in a way that fulfils DIN/TS 70121 needs. That means, the optional *EXI Cookie* ($EXI) shall never be used and the *Presence Bit for EXI Options* shall be always set to 0 (=false) (see Table 17). Consequently, the optional *EXI Options* shall never be part of a DIN/TS 70121 message. Each EXI implementation (on EV or EVSE side) shall discard messages that do not respect DIN/TS 70121 EXI header options.

[V2G-DC-178] An element/attribute which is not defined in "urn:din:70121:2012:MsgDef" namespace shall be encoded and decoded as schema deviation case (also see http://www.w3.org/TR/exi/#addingProductions) and shall follow the EXI Profile settings (see Table 18, also see http://www.w3.org/TR/exi-profile/) provided in the following table:

Table 18 — EXI profile settings

EXI Option	Description	DIN/TS 70121 value
maximumNumberOfBuiltInElementGrammars	This option is the maximum number of built-in element grammars for which dynamically productions other than AT(xsi:type) productions have been added.	0
maximumNumberOfBuiltInProductions	This option is the maximum number of top-level productions that can be dynamically inserted in built-in element grammars excluding AT(xsi:type) productions.	0

[V2G-DC-179] A simple type/value of an element/attribute which is not defined in "urn:din:70121:2012:MsgDef" namespace shall be encoded and decoded type-aware.

8.9 Application layer

8.9.1 Vehicle to grid application layer messages

The vehicle to grid application layer message definitions describes the client-server-based message exchange between EVCC and SECCs for the purpose of initializing and configuring the charging process of an EV. The message set is designed to cover the use cases defined in clause 7 of this specification. The messages and the required message flow (i.e. communication protocol) represent the application layer according to the OSI layered architecture model.

The message set, message flow and the behaviour specific to a certain message are described in clause 9.

8.9.2 Application layer service primitives

8.9.2.1 A-Data.confirmation

Table 19 describes the service primitive A-Data.confirmation and its parameter(s).

Table 19 — A-Data.confirmation service primitive

Primitive name	**A-Data.confirmation**
Entity to support	EVCC
Parameter Name	**Description**
A_Msg	— Session setup — Service discovery — Service detail — Service and payment selection — Contract authentication — Charge parameter discovery — Power delivery — Charging status — Cable check — Pre charging — Current demand — Welding detection — Session stop

A-Data.confirmation (A_Msg = “message name”) indicates the successful reception of a valid response message for the V2G message that is given by A_Msg.

EXAMPLE A-Data.confirmation (A_Msg = SessionSetupRes) indicates the successful reception of a SessionSetupRes Message as defined in 9.4.1.2.3.

8.9.2.2 A-Data.response

Table 20 describes the service primitive A-Data.response and its parameter(s).

Table 20 — A-Data.response service primitive

Primitive name	**A-Data.response**
Entity to support	SECC
Parameter Name	**Description**
A_Msg	— Session setup — Service discovery — Service detail — Service and payment selection — Contract authentication — Charge parameter discovery — Power delivery — Charging status — Cable check — Pre-charging — Current demand — Welding detection — Session stop

A-Data.response (A_Msg = "message name") indicates to the lower layer to send out a V2G response message for the V2G message type that is given by A_Msg.

EXAMPLE A-Data.response (A_Msg = SessionSetupRes) initiates the sending of the SesstionSetupRes message as defined in subclause 9.4.1.2.3.

8.9.2.3 A-Data.request

Table 21 describes the service primitive A-Data.request and its parameter(s).

Table 21 — A-Data.request service primitive

Primitive name	**A-Data.request**
Entity to support	EVCC
Parameter Name	**Description**
A_Msg	— Session setup — Service discovery — Service detail — Service and payment selection — Contract authentication — Charge parameter discovery — Power delivery — Charging status — Cable check — Pre-charging — Current demand — Welding detection — Session stop

A-Data.request (A_Msg= "message name") indicates to the lower layer to send out a V2G request message for the V2G message type that is given by A_Msg.

EXAMPLE A-Data.request (A_Msg = SessionSetupReq) initiates the sending of the SesstionSetupReq message as defined in 9.4.1.2.2.

8.9.2.4 A-Data.indication

Table 22 describes the service primitive A-Data.indication and its parameter(s).

Table 22 — A-Data.indication service primitive

Primitive name	**A-Data.indication**
Entity to support	SECC
Parameter Name	**Description**
A_Msg	— Session setup — Service discovery — Service detail — Service and payment selection — Contract authentication — Charge parameter discovery — Power delivery — Charging status — Cable check — Pre-charging — Current demand — Welding detection — Session stop

A-Data.indication (A_Msg= "message name") indicates the successful reception of a valid request message for the V2G message that is given by A_Msg.

EXAMPLE A-Data.indication (A_Msg = SessionSetupReq) indicates the successful reception of a SessionSetupReq Message as defined in 9.4.1.2.2.

8.9.3 SECC Discovery protocol

8.9.3.1 General information

An EVCC uses the SECC discover protocol (SDP) to get the IP address and port number of the SECC. The SDP client sends out SECC discovery request messages to the local link (multicast) expecting any SDP server to answer its request with a SECC discovery response message containing this information.

After the EVCC received the IP address and the port number of the SECC, it can establish a TCP connection to the SECC (see 8.7, V2G transfer protocol).

[V2G-DC-180] An SDP server shall be accessible in the local link.

As common for internet technologies, an SDP server may be implemented on the same physical device as the SECC and may also interface to the same IP address. If this is not the case, optimistic DAD as specified in RFC 4429 won't lead to a benefit.

8.9.3.2 Supported ports

SDP is a UDP based protocol. The ports listed in Table 14 are used by SDP.

[V2G-DC-473] An SDP client shall support the port V2G_UDP_SDP_CLIENT as defined in Table 14 for sending and receiving SDP messages.

[V2G-DC-474] An SDP server shall support the port V2G_UDP_SDP_SERVER as defined in Table 14 for receiving and sending SDP messages.

Depending on the implementation of the EVCC the dynamically assigned V2G_UDP_SDP_CLIENT port will be assigned once during or before the first transmission of a UDP packet to a SECC or can be dynamically reassigned for each individual UDP request message and response. Also depending on whether messages are repeatedly sent, response messages may arrive asynchronously and may not be associated to the exact corresponding request anymore.

[V2G-DC-475] The SDP client shall be able to handle asynchronously arriving SECC discovery response messages.

8.9.3.3 Protocol data unit

8.9.3.3.1 Structure

An SDP message is based on the V2GTP message format as defined in 8.7.3.1

[V2G-DC-181] An SDP client shall support the definitions in 8.7.3.1 as shown in Figure 24.

[V2G-DC-182] An SDP client shall use a separate UDP packet for each request message.

[V2G-DC-183] An SDP client shall locate the first byte of the request message header as defined in Figure 25 and Table 15 in the first byte of the UDP packet payload.

[V2G-DC-184] An SDP server shall support the definitions in 8.7.3.1 as shown in Figure 24.

[V2G-DC-185] An SDP Server shall use a separate UDP packet for each response message.

[V2G-DC-186] An SDP server shall locate the first byte of the response message header as defined in Figure 25 and Table 15 in the first byte of the UDP packet payload.

Table 15 defines the generic SDP header structure, which is identical to the structure of the V2GTP header as defined in 8.7.3.1

8.9.3.3.2 Header processing

An SDP header processing is based on the V2GTP message header processing as defined in 8.7.3.1

[V2G-DC-187] An SDP client shall apply the header processing as defined in 8.7.3.2 and shown in Figure 26.

[V2G-DC-188] An SDP server shall apply the header processing as defined in 8.7.3.2 and shown in Figure 26.

8.9.3.4 SECC discovery request message

The SDP client uses the SECC discovery request message to request the IP address and the port number of the SECC, as well as the security protocol and the transport protocol to be used throughout the V2G communication session.

[V2G-DC-189] Only SDP client shall send SECC discovery request messages.

[V2G-DC-190] An SDP client shall send SECC discovery request messages with the source IP address on which it expects the SECC Discovery response message.

[V2G-DC-191] An SDP client shall send SDP request messages to destination port V2G_UDP_SDP_SERVER as defined in Table 14.

[V2G-DC-192] An SDP client shall send SDP request messages with source port V2G_UDP_SDP_CLIENT as defined in Table 14 on which it expects the SECC Discovery response message.

[V2G-DC-193] An SDP client shall send SECC discovery request message to the destination local-link multicast address (FF02::1) as defined in IETF RFC 4291.

[V2G-DC-194] The SDP client shall send the SECC discovery request message with payload type value 0x9000 as defined in Table 16.

[V2G-DC-196] The SDP client shall send the SECC discovery request message with the payload length 2.

[V2G-DC-197] The SDP client shall send the SECC discovery request message with the payload as defined in Figure 28.

[V2G-DC-198] An SDP client shall send the payload in the order as shown in Figure 28. A byte with a lower number shall be sent before a byte with a higher number. The payload starts with byte 1 and ends with byte 2.

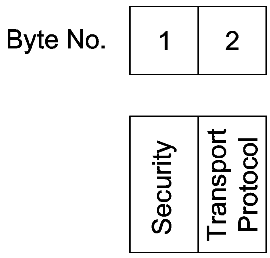


Figure 28 — SECC discovery request message payload

[V2G-DC-195] An SDP client shall use the encoding for the requested security option and the requested transport protocol as defined in Table 23.

Table 23 — Payload type SECC discover request message

	Security	**Transport protocol**
Byte no. SDP request message	1	2
Byte no. SDP response message	19	20
Applicable values	0x00 = secured with TLS 0x01-0x0F = reserved 0x10 = No transport layer security 0x11-0xFF = reserved	0x00= TCP 0x01-0x0F = reserved 0x10 = reserved for UDP 0x11-0xFF = reserved

[V2G-DC-476] An SDP server shall use the encoding for the requested security option and the requested transport protocol as defined in Table 23 to define the supported transmission security and transport protocol for the port provided in the same payload as the security and transport protocol bytes.

[V2G-DC-844] An EVCC that wants to use TCP with no security shall send a SECC discovery request message with transport protocol equal to "TCP" and security equal to "no transport layer security" according to Table 23.

[V2G-DC-845] An EVCC that wants to use TLS shall send a SECC Discovery Request message with transport protocol equal to "TCP" and security equal to "secured with TLS" according to Table 23.

NOTE Safety requirements for the EV are specified in ISO 17409. Safety requirements for the EVSE are specified in IEC 61851-1 and IEC 61851-23.

8.9.3.5 SECC discovery response message

The SDP server uses the SECC Discovery response message to response to an SECC discovery request message and provides the IP-address and the port of the SECC to the client, as well as the security protocol and the transport protocol to be used throughout the V2G communication session.

[V2G-DC-199] The SDP server shall be able to extract the source IP address and source port of a received UDP packet (client IP address and port number) and send a UDP packet to the identified IP address and port number.

[V2G-DC-200] An SDP server shall reply to any SECC Discovery request messages with an SECC discovery response message.

NOTE 1 This requirement ensures that an SDP server serving multiple clients can be reached at any time. This supports charging of multiple EVs at an EVSE with a single SECC.

[V2G-DC-201] An SDP client shall not reply to any SECC discovery request message.

[V2G-DC-202] An SDP server shall only send response messages after an SECC discovery request message has been received.

[V2G-DC-204] If an SDP server has multiple IP addresses, the SDP server shall send an SECC discovery response message with the source IP address on which the SDP server received the SECC request message.

[V2G-DC-205] An SDP server shall send an SDP response with source port V2G_UDP_SDP_SERVER as defined in Table 14.

[V2G-DC-206] An SDP server shall send an SECC discovery response message to the SDP client which sent the SECC Discovery request message.

[V2G-DC-207] An SDP server shall send an SECC discovery response message to the port of the SDP client which sent the SECC discovery request message.

[V2G-DC-208] An SDP server shall send the SECC discovery response message with the payload type value 0x9001 as defined in Table 16.

[V2G-DC-477] An SDP server shall send the SECC discovery response message with payload length 20.

[V2G-DC-209] An SDP server shall send the SECC discovery response message with the payload as defined in Figure 29.

[V2G-DC-210] An SDP server shall send the payload in the order as shown in Figure 29. A byte with a lower number shall be sent before a byte with a higher number. The payload starts with byte 1 and ends with byte 20.

[V2G-DC-211] A SDP server shall send the fields “SECC IP Address” and “SECC Port” in big endian format: The most significant byte is sent first the least significant byte is sent last.

[V2G-DC-846] An SECC that only supports TCP without TLS shall send a SECC Discovery Response message with transport protocol equal to “TCP” and Security equal to “No transport layer security” according to Table 23.

[V2G-DC-847] An SECC that supports the use of TLS shall send the same security and transport protocol in the SECC discovery response message as requested by the EVCC in the SECC discovery request message.

[V2G-DC-848] An SECC shall always send an SDP response with its supported security level according to [V2G-DC-846] or [V2G-DC-847] even if this does not meet the requested security level of the EVCC.

NOTE 2 The mechanism used by the SDP server to determine its own IP address is not in scope of this document.

NOTE 3 The source IP address and the source port of a received UDP packet is usually provided by the TCP/IP stack.

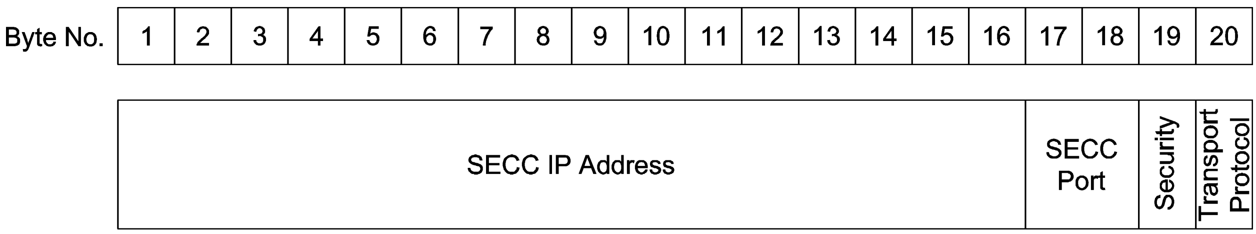


Figure 29 — SECC discovery response message payload

8.9.3.6 Timing and error handling

The process of SECC discovery is based on the application time out definitions. This subclause describes additional timing and error handling for the SECC discovery protocol.

[V2G-DC-212] The SDP client shall count the number of SECC Discovery request messages until a valid SECC discovery response message has been received.

[V2G-DC-213] The SDP client shall reset the counter for sent SECC discovery request messages after a valid SECC discovery response message has been received.

[V2G-DC-214] After sending an SECC discovery request message, the SDP client shall wait for an SECC discovery response message for at least 250 ms.

[V2G-DC-215] After unsuccessfully waiting for an SECC discovery response message the SDP client shall send a new SECC Discovery request message and increment the counter for sent SECC discovery response messages.

[V2G-DC-849] If the SDP client has not received any SECC discovery response message after sending in maximum 50 consecutive SECC discovery request messages it shall stop the SECC discovery.

[V2G-DC-850] The SDP Server shall wait for at least 12.5 s (see [V2G-DC-215] and [V2G-DC-849]) before stopping due to a timeout.

[V2G-DC-478] After stopping the SECC discovery, the SDP client shall go to the same state as defined for an application (refer to [V2G-DC-102] and Figure 22).

8.9.3.7 SECC Discovery service primitives

The N-SECC_Address.indication notifies about the status of the SECC IP address discovery. Table 24 describes the service primitive and its parameter(s).

Table 24 — N-SECC_Address.indication service primitive

Primitive name	**N-SECC_Address.indication**
Entity to support	EVCC
Parameter Name	**Description**
N_SECC_STATUS	— SECC IP-address discovered — Error

N-SECC_Address.indication (N_SECC_STATUS = SECC IP-address discovered) indicates that SDP returned a local or global IP-address for the SECC.

SECC_Address.indication (N_SECC_STATUS = Error) indicates any error during SDP as defined in 8.9.3.

[V2G-DC-217] If SDP returns a global SECC IP-address, the EVCC shall not indicate the discovered SECC IP-address before the EVCC has configured a global IP address as defined in 8.5.3.2 and 8.5.3.3.

[V2G-DC-218] If the SDP returns a global SECC IP-address, the EVCC shall indicate the discovered SECC IP-address after a global address assignment as defined in 8.5.3.2 and 8.5.3.3 is indicated.

[V2G-DC-851] If an EVSE receives a corrupted SDP request it shall disregard the message and shall not send a SDP response.

[V2G-DC-852] If an EV receives a corrupted SDP response, it shall continue as if the SDP response had not been received.

[V2G-DC-853] An EVSE that only supports DIN/TS 70121 shall always reply with transport protocol = TCP.

[V2G-DC-854] If an EVSE that supports more than DIN/TS 70121 receives a payload that indicates "reserved" for the transport protocol field it shall disregard the message.

[V2G-DC-855] If an EVSE that supports more than DIN/TS 70121 receives a payload that indicates "reserved" for the security field it shall disregard the message.

[V2G-DC-933] An EV that only supports DIN/TS 70121 shall always request transport protocol = TCP.

9 Application layer messages

9.1 General information and definitions

A V2G message uses the EXI-based presentation layer as described in 8.8.1. The communication between EVCC and SECC at application layer level is based on a client/server architecture. The EVCC always acts as a client (service requester) during the entire charging process, whereas the SECC always acts as a server (service responder). Hence the EVCC always initiates communication by sending a request message to the SECC which then returns the corresponding response message. All messages exchanged between EVCC and SECC are described with their syntax and their semantics in 9.2, 9.3, 9.4 and 9.5. The entire XML schema definition describing both V2G message sets is included in Annex A.

8.3.4 defines message timing and error handling for the V2G communication message exchange.

An example for a typical message sequence is shown in 9.7.5.

V2G communication consists of two different message sets:

— V2G application layer protocol handshake messages (refer to 9.2);

— V2G application layer messages (refer to 9.3).

9.2 Protocol handshake

9.2.1 Handshake request-response message pair

[V2G-DC-219] Before starting the application layer message exchange, an appropriate application layer protocol including its version shall be negotiated between the EVCC and the SECC.

To negotiate the protocol between the EVCC and the SECC the following application layer protocol handshake is performed.

[V2G-DC-220] The EVCC shall initiate the handshake sending a supportedAppProtocolReq message as depicted in Figure 30 to the SECC. This request message provides a list of charging protocols supported by the EVCC.

[V2G-DC-868] An EV that established a TLS connection shall not send an "AppProtocol" element refering to DIN/TS 70121 in supportedAppProtocolReq.

NOTE 1 This document does not support the use of TLS for its communication if TLS is used refer to other Standards (e.g. ISO 15118-2).

[V2G-DC-870] If the EV supports DIN/TS 70121 and other protocols (e.g. 15118-2), it shall send the "AppProtocol" element referring to DIN/TS 70121 as the first element in the list. Refer also to limitation defined in [V2G-DC-868].

NOTE 2 If a DIN/TS 70121-only charging station can only handle one name space entry although the standard requires to support 20 entries, it can cause a communication break if a DIN/TS 70121 + ISO 15118 EV does not indicate DIN/TS 70121 in the first entry of type AppProtocol in supportedAppProtocolReq.

[V2G-DC-221] Each entry in the list of supported EVCC protocols shall include the ProtocolNamespace, the VersionNumberMajor and VersionNumberMinor, the SchemaID dynamically assigned by the EVCC and the priority of the protocol entry. The priority in the EVCC request message enables the EVCC to announce the preferred application layer protocol where priority equal to 1 indicates the highest priority and priority equal to 20 indicates the lowest priority. The number of protocols included in the request message is limited to 20.

[V2G-DC-222] The SECC shall respond with a supportedAppProtocolRes message as depicted in Figure 31 indicating the protocol to be used for the subsequent message exchange by both the EVCC and the SECC.

[V2G-DC-223] The response message shall include a ResponseCode and the SchemaID of the protocol/schema which is agreed as application protocol for the following V2G communication session. Thereby, the SECC shall select from its own list of supported protocols the protocol with highest priority indicated by the EVCC.

[V2G-DC-224] The SECC shall confirm (positively respond) an EVCC supported protocol even if the values of the VersionNumberMinor in EVCC request message does not match with the VersionNumberMinor of an SECC supported protocol where the VersionNumberMajor matches.

A higher value in the VersionNumberMinor indicates that (in comparison to a lower value) additional data elements will be transmitted from either the EVCC or SECC. Implementations only supporting the lower VersionNumberMinor value may not be able to process the data and may have to ignore this data, however a difference in the VersionNumberMinor value between EVCC and SECC does not lead to an incompatibility. Refer to 9.2.2.2.2 showing examples for successful protocol negotiation.

[V2G-DC-225] All additional data element defined by the respective minor version shall be encoded as schema deviated case by the EXI coder (see also EXI option settings in 8.8.1.3).

[V2G-DC-226] Usually it is expected that the SECC can support the relevant application layer protocols indicated by the EVCC. However, when none of the application layer protocols included the list received from the EVCC is supported by the SECC, the ResponseCode in the response message shall be equal to “Failed_NoNegotiation” indicating that the protocol negotiation was not successful. In this error scenario, the response message shall not include a SchemaID.

[V2G-DC-869] If a TLS connection is established and the EV sends an "AppProtocol" element referring to DIN/TS 70121 only in supportedAppProtocolReq, the EVSE shall send supportedAppProtocolRes with ResponseCode = FAILED_NoNegotiation as specified in [V2G-DC-226].

[V2G-DC-227] If no successful protocol negotiation can be achieved the EVCC shall not initialize a V2G communication session.

[V2G-DC-228] This protocol handshake between EVCC and SECC shall be performed prior the actual V2G application layer message exchange. Only the message set defined in the agreed protocol shall be used in the V2G message flow except for minor deviations.

[V2G-DC-229] In the scope of this document, only the namespace “urn:din:70121:2012:MsgDef” and the version number 2.1 shall be used to indicate DIN/TS 70121.

9.2.2 Message definition supportedAppProtocolReq and supportedAppProtocolRes

[V2G-DC-230] The EVCC shall implement the message and message elements as in defined in Figure 30.

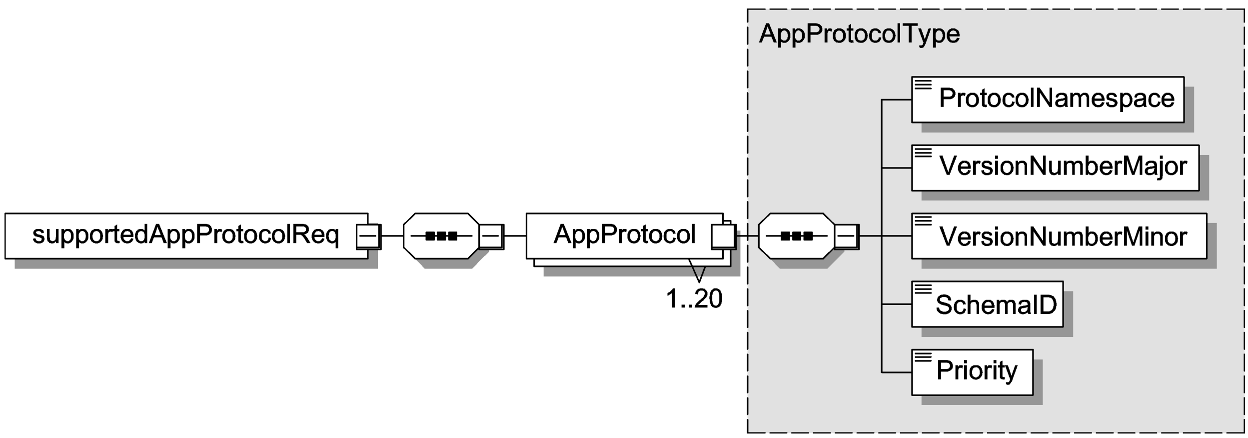


Figure 30 — Schema Diagram — supportedAppProtocolReq

[V2G-DC-231] The SECC shall implement the message and message elements as defined and Figure 31.

[V2G-DC-867] The EVSE shall support at least 20 namespaces according to the XML-Schema.

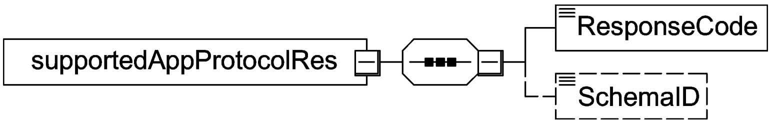


Figure 31 — Schema Diagram — supportedAppProtocolRes

NOTE Refer to Annex A.2 for the XML schema code.

9.2.2.1 Semantics description supportedAppProtocol messages

[V2G-DC-233] The message elements of the messages defined in Figure 30 and Figure 31 shall be used as defined in Table 25.

Table 25 — Semantics and type definition for supportedAppProtocol message elements

Element/Attribute Name	Type	Semantics
AppProtocol	complexType: includes the message elements defined in this table	This message element is used by the EVCC for transmitting the list of supported protocols. Each protocol with a particular version supported by the EVCC is represented by one AppProtocol entry in the request message (maximum number of entries: 20)
ProtocolNamespace	simpleType: protocolNamespaceType string (max length: 100) refer to Annex A.2 for the type definition	This message element is used by the EVCC to uniquely identify the Namespace URI of a specific protocol supported by the EVCC, i.e. this is the protocol name of the related protocol.
VersionNumberMajor	simpleType unsignedInt refer to Annex A.2 for the type definition	This message element is used by the EVCC to indicate the major version number of the protocol indicated in the message element ProtocolNamespace.
VersionNumberMinor	simpleType unsignedInt refer to Annex A.2 for the type definition	This message element is used by the EVCC to indicate the minor version number of the protocol indicated in the message element ProtocolNamespace.
SchemaID	simpleType: unsignedByte refer to Annex A.2 for the type definition	This message element is used by the EVCC to indicate the schemaID assigned by the EVCC to the protocol indicated in the message element ProtocolNamespace, VersionNumberMajor and VersionNumberMinor. This message element is used by the SECC to reference one of the EVCC supported protocols received in the request message. This identifier allows also for referring to a particular protocol later in the communication process (EXI Option schemaID).

Table 25 (*continued*)

Element/Attribute Name	Type	Semantics
Priority	simpleType: priorityType unsignedByte (range 1.20) refer to Annex A.2 for the type definition	This message element is used by the EVCC for indicating the protocol priority of a specific protocol allowing the SECC to select a protocol based on priorities.
ResponseCode	simpleType: responseCodeType enumeration refer to Annex A.2 for the type definition	This message element is used by the SECC for indicating whether the list of protocols received from the EVCC includes at least one protocol matching with the protocols supported by the SECC. Signals a response code with the following meaning: — OK_SuccessfulNegotiation: successful negotiation of a application protocol — OK_SuccessfulNegotiationWithMinorDeviation: successful negotiation of a protocol, with minor deviation — Failed_NoNegotiation: failed negotiation of protocols

9.2.2.2 Message examples

9.2.2.2.1 Protocol prioritization

V2G message example 1 and V2G message example 2 illustrate the exchange of supportedAppProtocol messages between the EVCC and the SECC. In the request message, the EVCC sends a prioritized list of supported application layer protocols (din:70121:2012 with version 2.0, iso:15118:2:2010 with version 1.0) to the SECC. In the response message the SECC confirms one protocol (din:70121:2012 with version 2.0) using a ResponseCode equal to “OK_SuccessfulNegotiation” and a schemaID equal to ten (10).

V2G message example 1 – supportedAppProtocolReq: protocol prioritization

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ns0:supportedAppProtocolReq xmlns:ns0="urn:iso:15118:2:2010:AppProtocol"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<AppProtocol>
<ProtocolNamespace>urn:din:70121:2012:MsgDef</ProtocolNamespace>
<VersionNumberMajor>2</VersionNumberMajor>
<VersionNumberMinor>1</VersionNumberMinor>
<SchemaID>10</SchemaID>
<Priority>1</Priority>
</AppProtocol>
<AppProtocol>
<ProtocolNamespace>urn:iso:15118:2:2010:MsgDef</ProtocolNamespace>
<VersionNumberMajor>1</VersionNumberMajor>
<VersionNumberMinor>0</VersionNumberMinor>
<SchemaID>20</SchemaID>
<Priority>5</Priority>
</AppProtocol>
</ns0:supportedAppProtocolReq>
```


V2G message example 2 – supportedAppProtocolRes: protocol prioritization


```xml
<?xml version="1.0" encoding="UTF-8"?>
<ns0:supportedAppProtocolRes xmlns:ns0="urn:iso:15118:2:2010:AppProtocol"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<ResponseCode>OK_SuccessfulNegotiation</ResponseCode>
<SchemaID>10</SchemaID>
</ns0:supportedAppProtocolRes>
```


9.2.2.2.2 Minor deviation

V2G message example 3 and V2G message example 4 illustrate the exchange of supportedAppProtocol messages between the EVCC and the SECC. In the request message, the EVCC sends just one supported application layer protocol (din:70121:2012 with version 1.0) to the SECC. The SECC supports protocol version 1.1 only. In the response message the SECC confirms protocol (din:70121:2012) with VersionNumberMajor equal to one (1) using a schemaID equal to one (1). However, the ResponseCode is equal to OK_SuccessfulNegotiationWithMinorDeviation signaling that a minor version deviation applies. The EVCC may now expect message elements which are not known by the EVCC but can be ignored.

V2G message example 3 – supportedAppProtocolReq: deviation in minor version

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:supportedAppProtocolReq xmlns:ns0="urn:iso:15118:2:2010:AppProtocol"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<AppProtocol>
<ProtocolNamespace>urn:din:70121:2012:MsgDef</ProtocolNamespace>
<VersionNumberMajor>2</VersionNumberMajor>
<VersionNumberMinor>1</VersionNumberMinor>
<SchemaID>1</SchemaID>
<Priority>1</Priority>
</AppProtocol>
</ns0:supportedAppProtocolReq>
```


V2G message example 4 – supportedAppProtocolRes: deviation in minor version


```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:supportedAppProtocolRes xmlns:ns0="urn:iso:15118:2:2010:AppProtocol"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<ResponseCode>OK_SuccessfulNegotiationWithMinorDeviation</ResponseCode>
<SchemaID>1</SchemaID>
</ns0:supportedAppProtocolRes>
```


9.3 Message structure

9.3.1 Overview

Subclause 9.3 describes the messages of the V2G messages and their contents. It is divided into the following 3 subclauses:

— V2G message definition (refer to 9.3.2);

— V2G message header definition (refer to 9.3.3);

— V2G message body definition (refer to 9.3.4).

NOTE Refer to Annex A for the XML schema code.

The DIN/TS 70121 application layer message set is signalized by the XML schema namespace "din:70121:2012:MsgDef". Refer to the XML schema definition in Annex A for details concerning subnamespace definitions used for the message definition.

9.3.2 Message definition

Figure 32 shows the schema definition of the V2G application layer message.

[V2G-DC-234] The EVCC and the SECC shall implement the V2G message structure as defined in Figure 32.

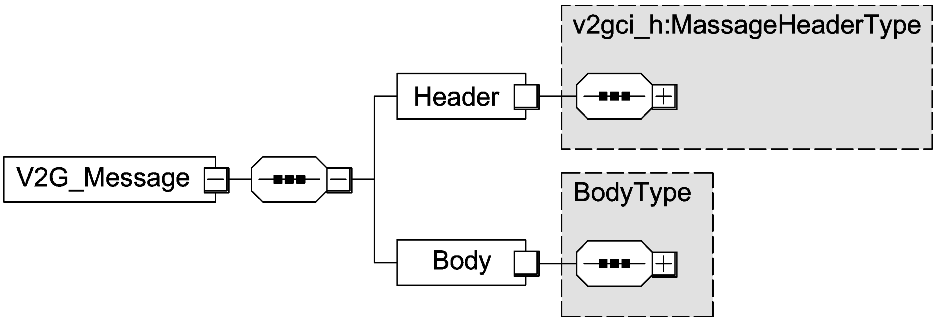


Figure 32 — Schema diagram — V2G message

[V2G-DC-235] The message elements of this message shall be used as defined in Table 26.

Table 26 — Semantics and type definition for a V2G message

Element Name	Type	Semantics
V2G_Message	complexType: Includes the message elements defined in this table	Root element that identifies this XML document as a V2G message. It contains two child elements, a header and body element.
Header	complexType: MessageHeaderType refer to subclause 9.3.3	This element contains the content of the message header. It includes generic information for protocol flow and is not directly related to the semantics of each particular message defined in 9.4.
Body	complexType: BodyType refer to subclause 9.3.4	This element contains the content of the message body. The message body provides the actual semantics of each message defined in section in 9.4.

V2G message example 5 shows an instance of a SessionSetupReq message. The header contains a SessionID equal to zero (0) because a new V2G communication session is about to being stated. The body contains the message specific content. In this case the message contains the message element EVCCID.

V2G message example 5 — Example for a SessionSetupReq message

```
<?xml version="1.0" encoding="UTF-8"?>
<v2gci_d:V2G_Message xmlns:v2gci_b="urn:din:70121:2012:MsgBody"
xmlns:xmlsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:v2gci_d="urn:din:70121:2012:MsgDef"
xmlns:v2gci_t="urn:din:70121:2012:MsgDataTypes"
 xmlns:v2gci_h="urn:din:70121:2012:MsgHeader">
<v2gci_d:Header>
 <v2gci_h:SessionInformation>
 <v2gci_t:SessionID>00</v2gci_t:SessionID>
</v2gci_h:SessionInformation>
</v2gci_d:Header>
<v2gci_d:Body>
<v2gci_b:SessionSetupReq>
<v2gci_b:EVCCID>002412E2B490</v2gci_b:EVCCID>
</v2gci_b:SessionSetupReq>
</v2gci_d:Body>
</v2gci_d:V2G_Message>
```

9.3.3 Message header definition

The message header contains general information that is included in all messages. Figure 33 shows the schema definition of the V2G message header.

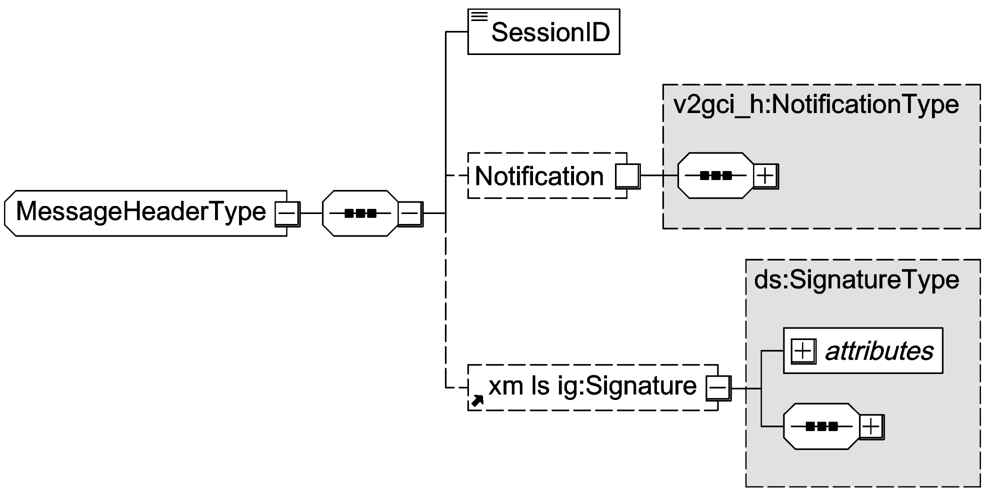


Figure 33 — Schema diagram — message header

[V2G-DC-236] The message elements of the message header shall be used as defined in Table 27.

Table 27 — Semantics and type definition for a V2G message header

Element Name	Type	Semantics
SessionID	simpleType: sessionIDType: hexBinary (max length: 8)	This message element is used by EVCC and SECC for uniquely identifying a V2G communication Session. Refer to 9.4.1.2.1 for requirements concerning this message element.
Notification	complexType: NotificationType,	Optional Element[a]
Signature	—	Optional Element[a]:

[a] In the scope of this document, this optional element is not used.

[V2G-DC-548] In the scope of this document, the element “Signature” shall not be used.

[V2G-DC-619] In the scope of this document, the element “Notification” shall not be used.

NOTE See also [V2G-DC-241] concerning the element “SessionID”.

9.3.4 Message body definition

The message body contains information details related to a specific message. Figure 34 shows the schema definition of the V2G message body. The messages described in the following section are derived from BodyBaseType, which represents the abstract message content (refer to 9.3.2). The different application messages are defined by the BodyElement and described in detail in 9.4.

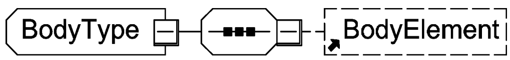


Figure 34 — Schema diagram — message body

[V2G-DC-237] The BodyElement shall be used as defined in Table 28.

Table 28 — Semantics and type definition for a V2G message body

Element Name	Type	Semantics
BodyElement	complexType: BodyBaseType, Refer to 9.4	BodyElement is a head element of a substitution group and does not appear itself in an instance of a message. Instead, one of the body elements defined in the substitution group in 9.4 is instantiated.

9.4 BodyElement Definitions

9.4.1 Common Messages

9.4.1.1 Overview

Messages defined as common messages can be applied to the message sequence in any charging mode defined in ISO 15118-2.

9.4.1.2 Session setup

9.4.1.2.1 Session Setup handling

After a connection has been established between the EVCC and the SECC the V2G communication session is established on application layer.

[V2G-DC-238] A V2G communication session shall be identified by a session identifier (session ID).

[V2G-DC-239] The SessionID shall not change during a V2G communication session.

[V2G-DC-241] The EVCC shall set SessionID equal to zero (0).

[**V2G-DC-993]** When receiving the SessionSetupReq with the parameter SessionID equal to zero (0), the SECC shall generate a new (not stored) SessionID value different from zero (0) and return this value in the SessionSetupRes message header.

[V2G-DC-872] If the SECC receives a SessionSetupReq including a SessionID value which is not equal to zero (0) and not equal to the SessionID value stored from the preceding V2G communication session, it shall send a SessionID value in the SessionSetupRes message that is unequal to "0" and unequal to the SessionID value stored from the preceding V2G communication session and indicate the new V2G communication session with the ResponseCode set to "OK_NewSessionEstablished" (refer also to [V2G-DC-393] for applicability of this response code).

[V2G-DC-934] If the SessionID is checked during the V2G communication session, the EVCC shall first compare the length and then the actual value.

[V2G-DC-935] If the length differs, the EVCC shall interpret the SessionID to be different.

[V2G-DC-873] An EV shall send SessionID = 0 as single hexBinary value: "0x00" (length 1 byte).

[V2G-DC-874] An EVSE shall accept all representation of SessionID = 0. (It shall not do a string comparison).

The EVSE may evaluate the hex value of the received SessionID e.g. "00", .., "0000", "00000000", ... ,"0000000000000000" as 0x00.

NOTE ASCII “0“ (Hex: 0x30) is not allowed.

9.4.1.2.2 Session Setup Request

By using the SessionSetupReq message the EV establishes a V2G communication session.

[V2G-DC-243] The EVCC and the SECC shall implement the mandatory message and message elements as defined in Figure 35.

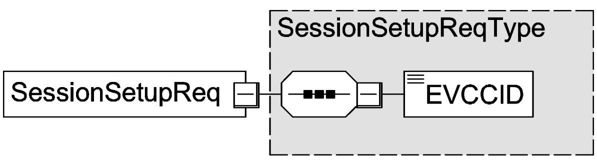


Figure 35 — Schema diagram — SessionSetupReq

[V2G-DC-244] The message elements of this message shall be used as defined in Table 29.

Table 29 — Semantics and type definition for SessionSetupReq

Element Name	Type	Semantics
EVCCID	simpleType: evccIDType hexBinary (max length: 8) refer to A.6 for the type definition	Specifies the EV's identification in a readable format. It contains the MAC address of the EVCC as six hexBinary encoded bytes, i.e. the element shall have a length of six bytes.

9.4.1.2.3 Session Setup Response

By using the SessionSetupRes the SECC responds to an SessionSetupReq. With the SessionSetup response the EVSE notifies the EV with an enclosed response code, whether establishing a new session was successful or not.

[V2G-DC-245] The EVCC and the SECC shall implement the mandatory message and message elements as defined in Figure 36.

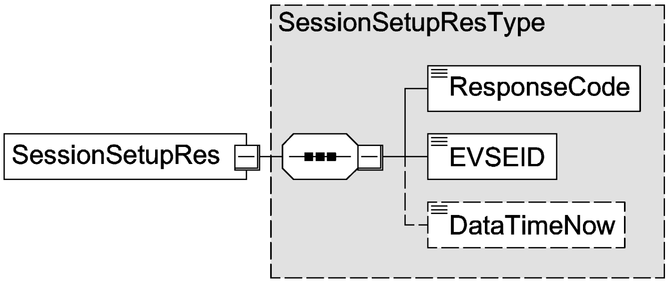


Figure 36 — Schema diagram — SessionSetupRes

[V2G-DC-246] The message elements of this message shall be used as defined in Table 30.

Table 30 — Semantics and type definition for SessionSetupRes

Element Name	Type	Semantics
ResponseCode	simpleType: responseCodeType enumeration refer to A.6 for the type definition	Response code indicating the acknowledgment status of any of the V2G messages received by the SECC.
EVSEID	simpleType: evseIDType hexBinary (max length: 32) refer to A.6 for the type definition	Any ID that uniquely identifies the EVSE. The format of this message element is defined in DIN SPEC 91286. If an SECC cannot provide such ID data, the value of the EVSEID is set to zero (00hex).
DateTimeNow	simpleType long refer to Annex A for the type definition	Optional Element: Timestamp of the current SECC time using to the unix time stamp format. The EV

[V2G-DC-247] The SECC and the EVCC may use the format for EVSEID as defined in DIN SPEC 91286.

[V2G-DC-620] The Power Outlet ID of DIN SPEC 91286 shall use the following pattern:
<Power Outlet ID> = DIGIT * 31 (DIGIT / "*");

[V2G-DC-621] To transform a string-based DIN SPEC 91286 EVSE ID to hexBinary representation and vice versa, the following conversion rules shall be used for each character and hex digit: ’0’ <–> 0x0, ’1’ <–> 0x1, ’2’ <–> 0x2, ’3’ <–> 0x3, ’4’ <–> 0x4, ’5’ <–> 0x5, ’6’ <–> 0x6, ’7’ <–> 0x7, ’8’ <–> 0x8, ’9’ <–> 0x9, ’*’ <–> 0xA, Unused <–> 0xB .. 0xF.

EXAMPLE The DIN SPEC 91286 EVSE ID “49*89*6360” is represented as “0x49 0xA8 0x9A 0x63 0x60”.

NOTE 1 In case a charger supports DIN and ISO at the same time it needs different EVSEIDs according to the protocol used.

[V2G-DC-876] If the SECC wants to send zero, it shall send the EVSEID as single hexBinary value: "0x00".

[V2G-DC-877] If the EVCC receives EVSEID="0x00", it shall ignore the EVSEID.

NOTE 2 The type of the EVSEID was changed from DIN/TS 70121 to ISO 15118-2.

[V2G-DC-878] An EVCC shall not expect a transmitted timestamp to be correct or a timestamp to be sent at all.

An EVSE may not send the timestamp.

9.4.1.3 Service discovery

9.4.1.3.1 Service discovery handling

The service discovery enables the EV to find all services provided by the EVSE. This document only describes relevant aspects of the V2G CI with regards to *charging* of the EV. Nevertheless, the basis for discovery of future value-added services is already considered and offers means for extensibility. Therefore, the service discovery differentiates between various service types and scopes.

9.4.1.3.2 Service discovery request

By sending the ServiceDiscoveryReq message, the EVCC triggers the SECC to send information about all services offered by the SECC. Furthermore, the EVCC can limit for particular services by using the service scope and service type elements.

[V2G-DC-248] The EVCC and the SECC shall implement the mandatory message and message elements as defined in Figure 37.

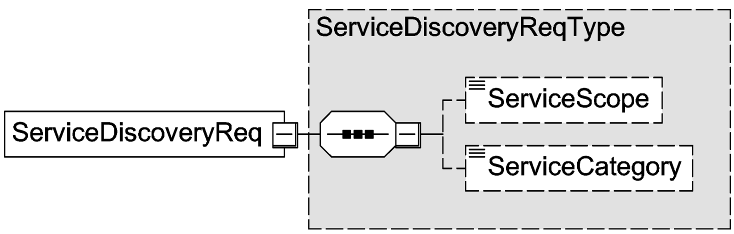


Figure 37 — Schema diagram — ServiceDiscoveryReq

[V2G-DC-249] The message elements of this message shall be used as defined in Table 31.

Table 31 — Semantics and type definition for ServiceDiscoveryReq

Element Name	Type	Semantics
ServiceScope	simpleType: serviceScopeType string (max length: 32) refer to A.6 for the type definition	Optional element: Defines the scope of the service discovery. A scope is defined by a unique URI which corresponds to a service provider (e.g. mobility provider, value added service provider etc.). The Service Discovery can respect multiple scopes in one request. By applying a scope to the service discovery, the resulting list of services returned in the ServiceDiscoveryRes can be limited to a certain scope, thus enables pre-filtering. The SECC always returns all supported services for alle scopes if no specific ServiceScope has been indicated in request message.
ServiceCategory	simpleType: serviceCategoryType enumeration refer to A.6 for the type definition	Optional element: Defines the service category for the service discovery (e.g. EV charging, internet access etc.). By applying a category to the service discovery, the resulting list of services returned in the ServiceDiscoveryRes can be limited to a certain category of services, thus enables pre-filtering. The SECC always returns all supported services for all categories if no specific category has been indicated in request message by using the ServicCategory message element. In the scope of this document, if this optional element is used, it shall always contain the value "EVcharging".

[V2G-DC-557] In the scope of this document, if the optional element ServiceCategory is used, it shall always contain the value "EVcharging".

[V2G-DC-879] If the EV sends a value ServiceCategory that is not "EVcharging", the EVSE shall ignore the parameter ServiceCategory.

[V2G-DC-622] In the scope of this document, the optional element ServiceScope shall not be used.

9.4.1.3.3 Service discovery response

After receiving the ServiceDiscoveryReq message of the EVCC, the SECC sends the ServiceDiscoveryRes message. In case of a successful service discovery, the response lists all available services of the SECC for the defined criteria. In case the service discovery failed, the service list is empty, and the response code indicates potential reasons.

[V2G-DC-250] The EVCC and the SECC shall implement the mandatory message and message elements as defined in Figure 38.

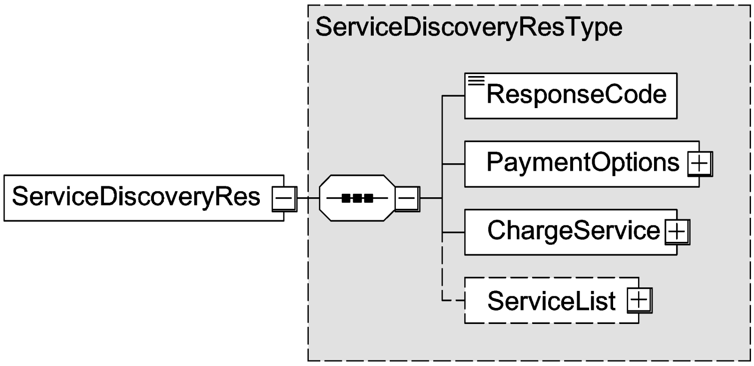


Figure 38 — Schema diagram — ServiceDiscoveryRes

[V2G-DC-251] The message elements of this message shall be used as defined in Table 32.

Table 32 — Semantics and type definition for ServiceDiscoveryRes

Element Name	Type	Semantics
ResponseCode	simpleType: responseCodeType enumeration refer to A.6 for the type definition	Response code indicating the acknowledgment status of any of the V2G messages received by the SECC.
PaymentOptions	complexType: paymentOptionsType refer to 9.5.2.5	This element includes the list of payment options an SECC offers to the EVCC indicating what method could be chosen to pay for the services. The EVCC can only select one payment method for all services used by the EVCC.
ChargeService	complexType: ServiceChargeType refer to 9.5.2.3	Available charging services supported by the EVSE.
ServiceList	—	Optional Element[a]
[a] In the scope of this document, this optional element is not used.		

[V2G-DC-549] In the scope of this document, the element “ServiceList” shall not be used.

9.4.1.4 Service and payment selection

9.4.1.4.1 Service and payment selection handling

Based on the provided services and the corresponding payment options by the SECC this message pair allows the transmission of the selected PaymentOption, SelectedServices and related ParameterSets.

[V2G-DC-252] Only the PaymentOption “ExternalPayment” shall be used since detailed payment options are not defined.

9.4.1.4.2 Service and payment selection request

This request message transports the information on the selected services and on how the all the selected services are paid.

[V2G-DC-253] The EVCC and the SECC shall implement the mandatory message and message elements as defined in Figure 39.

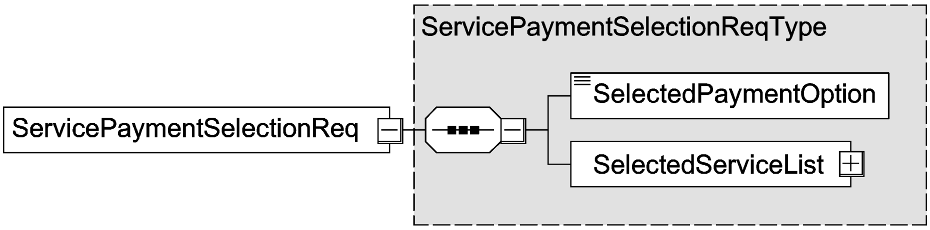


Figure 39 — Schema diagram — ServicePaymentSelectionReq

[V2G-DC-254] The message elements of this message shall be used as defined in Table 33.

Table 33 — Semantics and type definition for ServicePaymentSelectionReq

Element Name	Type	Semantics
SelectedPaymentOption	complexType: PaymentOptionType refer to A.6	This element is used for indicating the payment type selected for the use of all selected services in the selectedServiceList.
SelectedServiceList	complexType: SelectedServiceListType refer to 9.5.2.13	List contains all selected ServiceIDs and the optional parameterSetID for applicable for the respective serviceID.

9.4.1.4.3 Service and Payment Selection Response

With this message, the SECC informs the EVCC whether the selected services and payment option were accepted (see Figure 40).

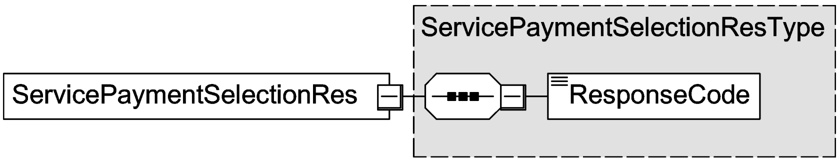


Figure 40 — Schema diagram — ServicePaymentSelectionRes

[V2G-DC-255] The message elements of this message shall be used as defined in Table 34.

Table 34 — Semantics and type definition for ServicePaymentSelectionRes

Element Name	Type	Semantics
ResponseCode	simpleType: responseCodeType enumeration refer to A.6 for the type definition	Response code indicating the acknowledgment status of any of the V2G messages received by the SECC.

9.4.1.5 Contract authentication

9.4.1.5.1 Contract authentication request

[V2G-DC-481] The EVCC and the SECC shall implement the mandatory messages and message elements as defined in Figure 41.

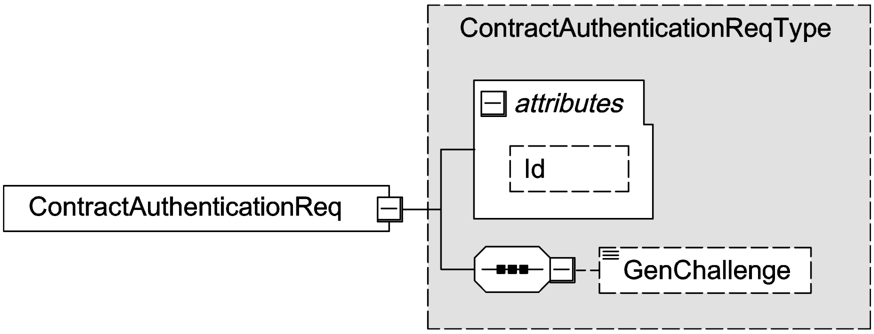


Figure 41 — Schema diagram — ContractAuthenticationReq

[V2G-DC-482] The message elements of this message shall be used as defined in Table 35.

Table 35 — Semantics and type definition for ContractAuthenticationReq

Element Name	Type	Semantics
GenChallenge	simpleType genChallengeType string refer to A.6 for the type definition	Optional Element: NOTE In the scope of this document, this optional element is not used.
Id	simpleType IDREF refer to A.6 for the type definition	Optional Element[a]
[a] In the scope of this document, this optional element is not used.		

[V2G-DC-550] In the scope of this document, the element “GenChallenge” shall not be used.

[V2G-DC-545] In the scope of this document, the element “Id” shall not be used.

9.4.1.5.2 Contract authentication response

[V2G-DC-483] The EVCC and the SECC shall implement the mandatory messages and message elements as defined in Figure 42.

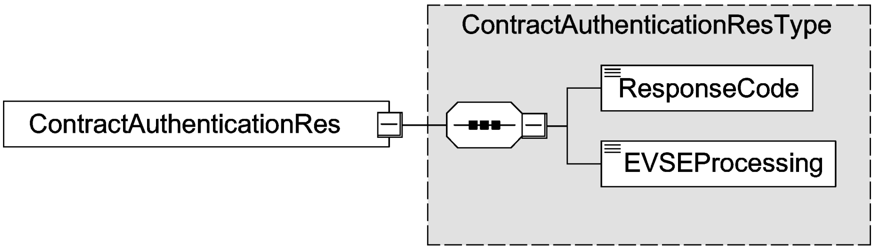


Figure 42 — Schema diagram — ContractAuthenticationRes

[V2G-DC-484] The message elements of this message shall be used as defined in Table 36.

Table 36 — Semantics and type definition for ContractAuthenticationRes

Element Name	Type	Semantics
ResponseCode	simpleType: responseCodeType enumeration refer to A.6 for the type definition	Response Code indicating the acknowledgment status of any of the V2G messages received by the SECC.
EVSEProcessing	simpleType: EVSEProcessingType enumeration refer to A.6 for the type definition	Parameter indicating that the EVSE has finished the processing that was initiated after the ContractAuthenticationReq or that the EVSE is still processing at the time the response message was sent.

NOTE By using the EVSEProcessing parameter, the EVSE can indicate to the EVCC that the processing has not finished but a response message was sent to fulfil the timeout and performance requirements defined in 0. This allows delaying the V2G communication Session while fulfilling the performance and timeout requirements.

9.4.1.6 Charge parameter discovery

9.4.1.6.1 Charge parameter discovery handling

After being authorized for charging at the EVSE the EV sends a charge parameter discovery request message to the EVSE. This message provides status information about the EV and furthermore provides parameters like estimated energy amounts for recharge and the point in time for the end of charge. In turn, the EVSE responds with its status information and provides currently applicable power output limits.

Handling of maximum values in ChargeParameterDiscovery is specified in IEC61851-23.

9.4.1.6.2 Charge parameter discovery request

By sending the Charge Parameter Discovery Request message the EVCC provides its charging parameters to the SECC. This message provides status information about the EV and additional charging parameters, like estimated energy amounts for recharge and the point in time for the end of charge.

[V2G-DC-256] The EVCC and the SECC shall implement the mandatory messages and message elements as defined in Figure 43.

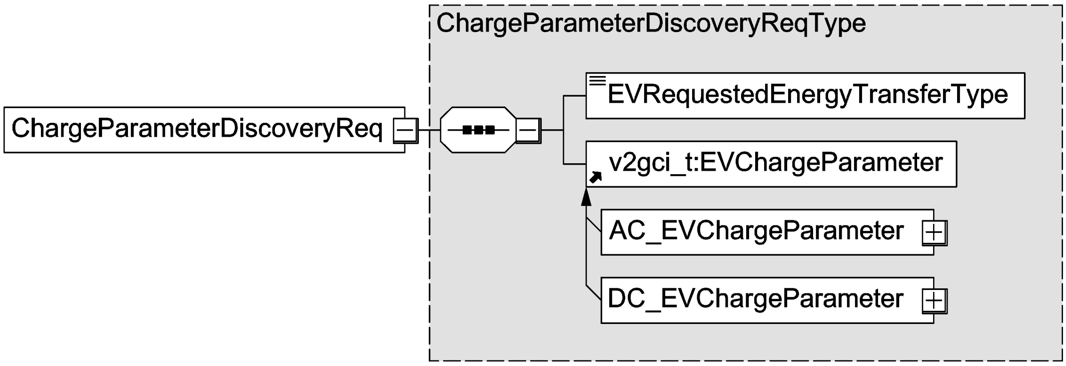


Figure 43 — Schema diagram — ChargeParameterDiscoveryReq

[V2G-DC-258] The message elements of this message shall be used as defined in Table 37.

Table 37 — Semantics and type definition for ChargeParameterDiscoveryReq

Element Name	Type	Semantics
EVRequestedEnergyTransfer-Type	simpleType: EVRequestedEnergyTransferType enumeration refer to A.6 for the type definition and the Table 38.	Selected energy transfer type for charging that is requested by the EVCC. Refer to Table 38 for details.
AC_EVChargeParameter	—	See [V2G-DC-551].
DC_EVChargeParameter	complexType: DC_EVChargeParameterType substitutes abstract type EV_ChargeParameterType refer to 9.5.3.3	This element is used the by the EVCC for initiating the target setting process for DC charging.

[V2G-DC-551] In the scope of this document, the element “AC_EVChargeParameter” shall not be used.

The definition of EVRequestedEnergyTransferType supports connectors according to IEC 62196-3 configuration CC (corresponding to IEC 62196-2 Type 1), IEC 62196-3 configuration DD (corresponding to IEC 62196-2 Type 2), IEC 62196-3 Configuration EE, and IEC 62196-3 configuration FF. Based on the supported connectors, the EVCC can choose the charging services as defined in Table 38.

[V2G-DC-259] The EVCC shall use the EVRequestedEnergyTransferType as described in Table 38.

Table 38 — Semantics for EVRequestedEnergyTransferType

EnergyTransferType	Offered charging service
AC_single_phase_core	Not used in the scope of this document.
AC_three_phase_core	Not used in the scope of this document.
DC_core	EV asks for DC charging using the core pins of an connector.
DC_extended	EV asks for DC charging using the extended pins of an IEC 62196-3 configuration EE or configuration FF connector

Table 38 (*continued*)

EnergyTransferType	Offered charging service
DC_combo_core	Not used in the scope of this document.
DC_unique	Not used in the scope of this document.

[V2G-DC-623] If the EV requests DC charging using the extended pins of a vehicle connector according to IEC 62196-3 Configuration EE or configuration FF, the EVCC shall transmit “DC_extended” in EVRequestedEnergyTransferType.

[V2G-DC-624] If the EV requests DC charging using the core pins of a connector the EVCC shall transmit “DC_core” in EVRequestedEnergyTransferType.

[V2G-DC-625] In the scope of this document, the EVCC shall not transmit other values than “DC_extended” and “DC_core” in EVRequestedEnergyTransferType.

9.4.1.6.3 Charge parameter discovery response

With the charge parameter discovery response message, the SECC provides applicable charge parameters from the grid’s perspective. Next to general charge, parameters of the EVSE this optionally includes further information on cost over time, cost over demand, cost over consumption or a combination of these. The term cost refers to any kind of cost specified in this document and is not limited to monetary costs. Based on this cost information the EV may optimize its charge for the requested amount of energy.

[V2G-DC-260] The EVCC and the SECC shall implement the mandatory message and message elements as defined in Figure 44.

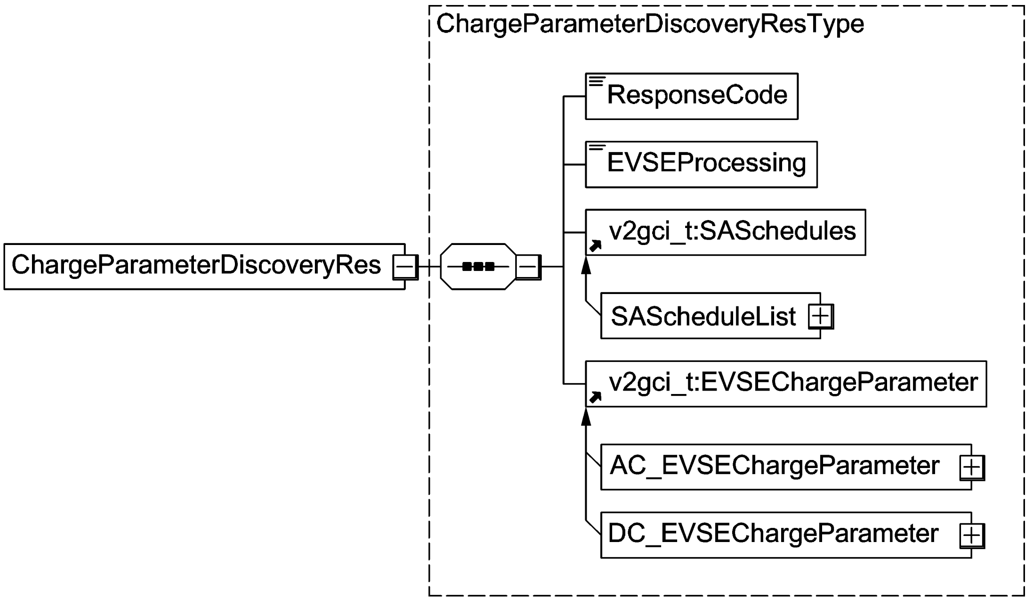


Figure 44 — Schema diagram — ChargeParameterDiscoveryRes

[V2G-DC-262] The message elements of this message shall be used as defined in Table 39.

Table 39 — Semantics and type definition for ChargeParameterDiscoveryRes

Element Name	Type	Semantics
ResponseCode	simpleType: responseCodeType enumeration refer to A.6 for the type definition	Response Code indicating the acknowledgment status of any of the V2G messages received by the SECC.
EVSEProcessing	simpleType: EVSEProcessingType enumeration refer to A.6 for the type definition	Parameter indicating that the EVSE has finished the processing that was initiated after the ChargeParameterDiscoveryReq or that the EVSE is still processing at the time the response message was sent.
SAScheduleList	complexType: SAScheduleListType substitutes abstract type SASchedulesType refer to 9.5.2.8	Includes several tuples of schedules from secondary actors
AC_EVSEChargeParameter	—	See [V2G-DC-552].
DC_EVSEChargeParameter	complexType: DC_EVSEChargeParameterType substitutes abstract type EVSE_ChargeParameterType refer to 9.5.3.4	This element is used the by the SECC for initiating the target setting process for DC charging.

NOTE By using the EVSEProcessing parameter, the EVSE can indicate to the EVCC that the processing has not finished but a response message is sent to fulfil the timeout and performance requirements defined in IEC 61851-23. This allows delaying the V2G communication session while fulfilling the performance and timeout requirements.

[V2G-DC-552] In the scope of this document, the element “AC_EVSEChargeParameter” shall not be used.

[V2G-DC-882] The EV shall ignore the SASchedule received in ChargeParameterDiscoveryRes.

[V2G-DC-884] The EVSE shall send an SASchedule with the same values as sent in EVSEMaximumPowerLimit for a power smaller or equal to +32.767W.

[V2G-DC-885] The EVSE shall send an SASchedule with the value equal to +32.767W for a power bigger than +32.767W.

9.4.1.6.4 Power delivery handling

The Power Delivery message exchange marks the point in time when the EVSE provides voltage to its connector and the EV can start to charge its battery.

9.4.1.6.5 Power delivery request

By sending the power delivery request the EVCC requests the EVSE to switch power on and transmits the charging profile it will follow during the charging process. The point in time this message is sent does not necessarily correlate with the start of the charging process. The onboard charger of the EV may decide based on its schedule when the charging process starts.

[V2G-DC-263] The EVCC and the SECC shall implement the mandatory message and message elements as defined in Figure 45.

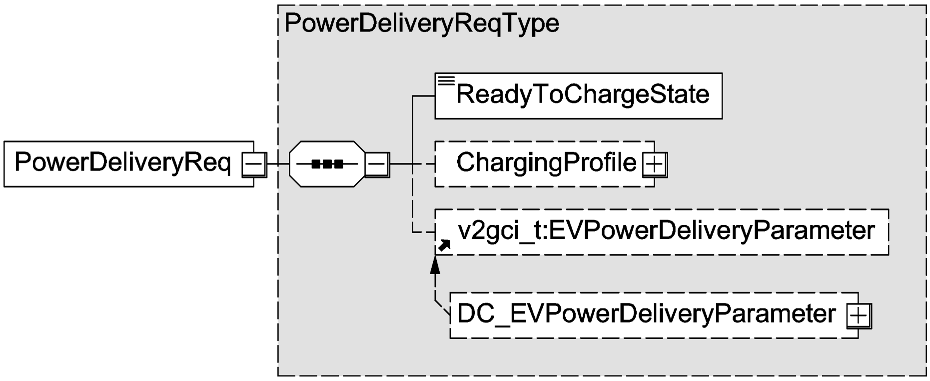


Figure 45 — Schema diagram — PowerDeliveryReq

[V2G-DC-264] The message elements of this message shall be used as defined in Table 40.

Table 40 — Semantics and type definition for PowerDeliveryReq

Element Name	Type	Semantics
ReadyToChargeState	simpleType: boolean refer to Annex A for the type definition	This message element is used to request the EVSE to fulfil all conditions that the energy transfer can start as soon as the EV onboard system begins to retrieve energy without any further action to be taken (i.e. the EVSE is requested to close its contactors successfully). If ReadyToCharge is equal to TRUE the EVSE is requested to prepare the energy flow for an immediate start, if ReadyToCharge is equal to FALSE the EVSE is requested to stop the energy flow.
ChargingProfile	complexType: ChargingProfileType refer to 9.5.2.6	Optional Element: Charge reservation of the EV as maximum amount of power drawn over time for this charging session.
DC_EVPowerDeliveryParameter	complexType: DC_EVPowerDeliveryParameter-Type substitutes abstract type EVPowerDeliveryParameter refer to 9.5.3.5	Optional Element[a] This element is used the by the EVCC for transmitting the parameters for power delivery.
[a] In the scope of this document, this optional element shall be used.		

[V2G-DC-558] In the scope of this document, the optional element “DC_EVPowerDeliveryParameter” shall be used.

[V2G-DC-991] The EVCC shall not send the parameter ChargingProfile in PowerDeliveryReq.

[V2G-DC-992] The SECC shall ignore the optional Element ChargingProfile in PowerDeliveryReq message if included.

9.4.1.6.6 Power delivery response

After receiving the power delivery request message of the EVCC, the SECC sends the power delivery response message including information if power will be available.

[V2G-DC-265] The EVCC and the SECC shall implement the mandatory messages and message elements as defined in Figure 46.

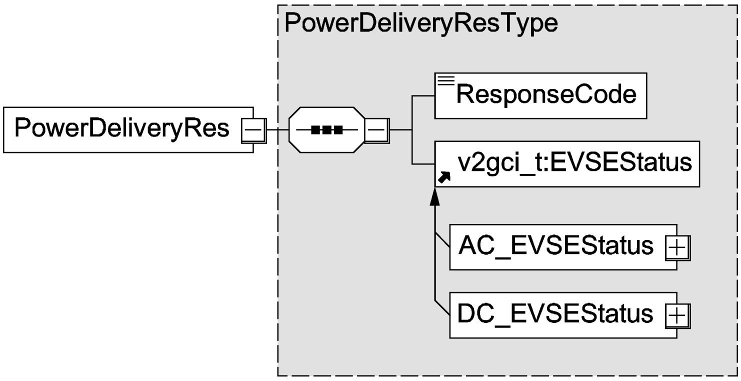


Figure 46 — Schema diagram — PowerDeliveryRes

[V2G-DC-266] The SECC shall always accept the ChargingProfile of the EVCC (see 9.5.2.6) if it does not exceed the PMax values of all PMaxScheduleEntry elements (see 9.5.2.10) according to the chosen SAScheduleTuple element (see 9.5.2.8) in the charge parameter discovery response message (see 9.4.1.6.3).

[V2G-DC-267] The SECC shall send the negative response code FAILED_ChargingProfileInvalid in the PowerDelivery response message if the EVCC sends a ChargingProfile (see 9.5.2.6) which is not adhering to the PMax values of all PMaxScheduleEntry elements (see 9.5.2.10) according to the chosen SAScheduleTuple element (see 9.5.2.8) in the charge parameter discovery response message (see 9.4.1.6.3).

[V2G-DC-268] The message elements of this message shall be used as defined in Table 41.

Table 41 — Semantics and type definition for PowerDeliveryRes

Element Name	Type	Semantics
ResponseCode	simpleType: responseCodeType enumeration refer to A.6 for the type definition	Response code indicating the acknowledgment status of any of the V2G messages received by the SECC.
AC_EVSEStatus	—	See [V2G-DC-553].
DC_EVSEStatus	complexType: DC_EVSEStatusType substitutes abstract type EVSE_StatusType refer to 9.5.3.1	This element is used the by the SECC for indicating the SECC status.

[V2G-DC-553] In the scope of this document, the element “AC_EVSEStatus” shall not be used.

9.4.1.7 Session stop

9.4.1.7.1 Session stop handling

This V2G message pair shall be used for terminating a V2G communication session initiated by preceding SessionSetupReq message.

9.4.1.7.2 Session stop request

By sending the session stop request the EV requests termination of the V2G communication session.

[V2G-DC-287] The EVCC and the SECC shall implement the mandatory message and message elements as defined in Figure 47.

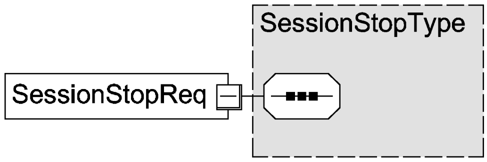


Figure 47 — Schema diagram — SessionStopReq

9.4.1.7.3 Session stop response

After receiving the session stop request message of the EVCC, the SECC sends the session stop response message informing the EV if terminating the V2G communication session was successful.

[V2G-DC-288] The EVCC and the SECC shall implement the mandatory message and message elements as defined in Figure 48.

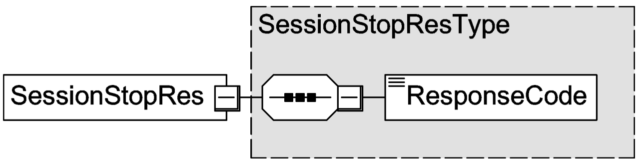


Figure 48 — Schema diagram — SessionStopRes

[V2G-DC-289] The message elements of this message shall be used as defined in Table 42.

Table 42 — Semantics and type definition for SessionStopRes

Element Name	Type	Semantics
ResponseCode	simpleType: responseCodeType enumeration refer to A.6 for the type definition	Response code indicating the acknowledgment status of any of the V2G messages received by the SECC.

9.4.2 DC messages

9.4.2.1 Overview

Messages defined as DC messages shall be applied to DC charging message sequence only.

9.4.2.2 Cable check

9.4.2.2.1 Cable check handling

For a safe DC charging a cable check shall be performed prior to charging.

[V2G-DC-269] For cable check details see IEC 61851-23

9.4.2.2.2 Cable check request

With the Cable Check Request, the EV asks the EVSE to perform a cable check, which includes an isolation test, before charging.

[V2G-DC-270] The EVCC and the SECC shall implement the mandatory messages and message elements as defined in Figure 49.

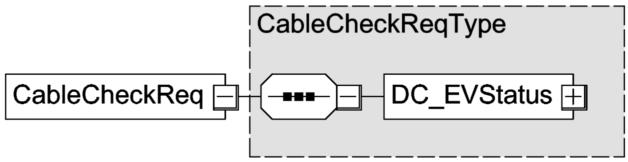


Figure 49 — Schema diagram — CableCheckReq

[V2G-DC-271] The message elements of this message shall be used as defined in Table 43.

Table 43 — Semantics and type definition for CableCheckReq

Element Name	Type	Semantics
DC_EVStatus	complexType: DC_EVStatusType refer to 9.5.3.2	Current status of the EV

9.4.2.2.3 Cable check response

After receiving the cable check request message of the EVCC, the SECC sends the cable check response message informing the EV about result of cable check and EVSE status.

[V2G-DC-272] The EVCC and the SECC shall implement the mandatory message and message elements as defined in Figure 50.

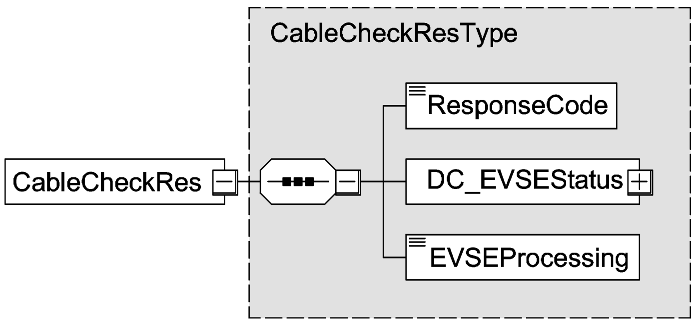


Figure 50 — Schema diagram — CableCheckRes

[V2G-DC-274] The message elements of this message shall be used as defined in Table 44.

Table 44 — Semantics and type definition for CableCheckRes

Element Name	Type	Semantics
ResponseCode	simpleType: responseCodeType enumeration refer to A.6 for the type definition	Response code indicating the acknowledgment status of any of the V2G messages received by the SECC.
DC_EVSEStatus	complexType: DC_EVSEStatusType refer to 9.5.3.1	Current status of the EVSE
EVSEProcessing	simpleType: EVSEProcessingType enumeration refer to A.6 for the type definition	Parameter indicating that the EVSE has finished the processing that was initiated after the CableCheckReq or that the EVSE is still processing at the time the response message was sent.

NOTE By using the EVSEProcessing parameter, the EVSE can indicate to the EVCC that the processing has not finished but a response message is sent to fulfil the timeout and performance requirements defined in IEC 61851-23. This allows to delay the V2G communication session while fulfilling the performance and timeout requirements.

9.4.2.3 Pre charge

9.4.2.3.1 Pre charge handling

Pre charge is used for adjusting the EVSE output voltage to the EV RESS voltage to minimize the inrush current when the contactors of the EV are being closed.

9.4.2.3.2 Pre-charge request

With the pre-charging request the EV asks the EVSE to apply certain values for output voltage and output current. Since the contactors of the EV are open during pre-charging, the actual current flow from the EVSE to the EV will be very small, i. e. in most cases smaller than the requested output current. The EV can use several pre-charging request/response message pairs to precisely adjust the EVSE output voltage to the EV RESS voltage measured inside the EV.

[V2G-DC-275] The EVCC and the SECC shall implement the mandatory messages and message elements as defined in Figure 51.

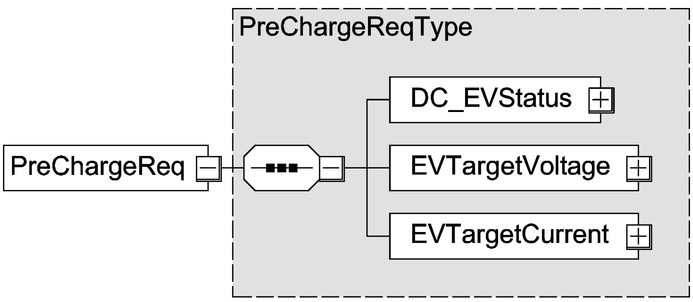


Figure 51 — Schema diagram — PreChargeReq

[V2G-DC-276] The message elements of this message shall be used as defined in Table 45.

Table 45 — Semantics and type definition for PreChargeReq

Element Name	Type	Semantics
DC_EVStatus	complexType: DC_EVStatusType refer to 9.5.3.2	Current status of the EV
EVTargetVoltage	complexType PhysicalValueType refer to 9.5.2.4	Target Voltage requested by EV
EVTargetCurrent	complexType PhysicalValueType refer to 9.5.2.4	Current demanded by EV

[V2G-DC-909] The EV shall not rely on the parameter EVSEPresentVoltage but it shall use its own measurement device for measuring the inlet voltage (pre charge voltage).

9.4.2.3.3 Pre charge response

After receiving the pre-charge request message of the EVCC, the SECC sends the pre-charge response message informing the EV about EVSE status and present EVSE output voltage.

[V2G-DC-277] The EVCC and the SECC shall implement the mandatory messages and message elements as defined in Figure 52.

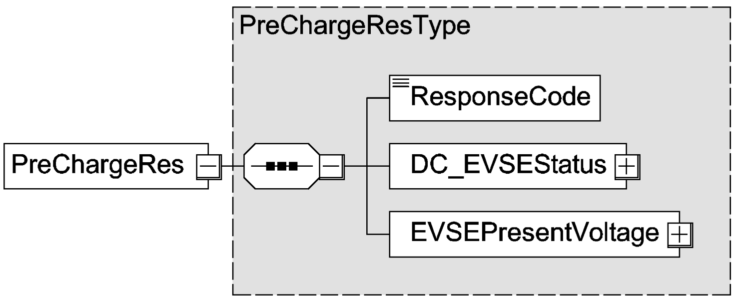


Figure 52 — Schema diagram — PreChargeRes

[V2G-DC-278] The message elements of this message shall be used as defined in Table 46.

Table 46 — Semantics and type definition for PreChargeRes

Element Name	Type	Semantics
ResponseCode	simpleType: responseCodeType enumeration refer to A.6 for the type definition	Response code indicating the acknowledgment status of any of the V2G messages received by the SECC.
DC_EVSEStatus	complexType: DC_EVSEStatusType refer to 9.5.3.1	Current status of the EVSE
EVSEPresentVoltage	complexType PhysicalValueType refer to 9.5.2.4	Present voltage of EVSE

[V2G-DC-910] The EVSE shall use EVSEPresentVoltage to send the real measurement value of its outlet voltage in PreChargeRes and CurrentDemandRes.

9.4.2.4 Current demand

9.4.2.4.1 Current demand handling

For DC charging control cyclic exchange of requested current and voltage from EV side is necessary.

9.4.2.4.2 Current demand request

By sending the current demand request the EV requests a certain current from EVSE. Also, the target voltage and current are transferred.

[V2G-DC-279] The EVCC and the SECC shall implement the mandatory message and message elements as defined in Figure 53.

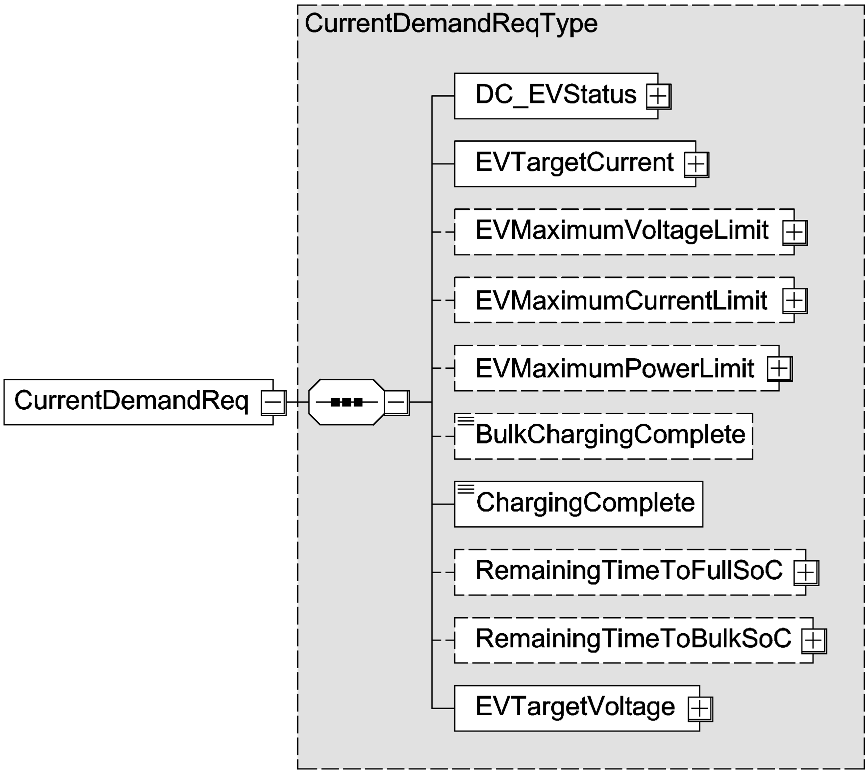


Figure 53 — Schema diagram — CurrentDemandReq

[V2G-DC-280] The message elements of this message shall be used as defined in Table 47.

Table 47 — Semantics and type definition for CurrentDemandReq

Element Name	Type	Semantics
DC_EVStatus	complexType: DC_EVStatusType refer to 9.5.3.2	Current status of the EV
EVTargetCurrent	complexType PhysicalValueType refer to 9.5.2.4	Instantaneous current requested by the EV
EVMaximumVoltageLimit	complexType PhysicalValueType refer to 9.5.2.4	Optional element: Maximum voltage allowed by the EV
EVMaximumCurrentLimit	complexType PhysicalValueType refer to 9.5.2.4	Optional element: Maximum current allowed by the EV
EVMaximumPowerLimit	complexType PhysicalValueType refer to 9.5.2.4	Optional element: Maximum power allowed by the EV

Table 47 (*continued*)

Element Name	Type	Semantics
BulkChargingComplete	simpleType boolean refer to Annex A for the type definition	Optional element: If set to TRUE, the EV indicates that bulk charge (approx. 80 % SOC) is complete.
ChargingComplete	simpleType boolean refer to Annex A for the type definition	If set to TRUE, the EV indicates that full charge (100 % SOC) is complete.
RemainingTimeToFullSoC	complexType PhysicalValueType refer to 9.5.2.4	Optional element: Estimated or calculated time until full charge (100 % SOC) is complete
RemainingTimeToBulkSoC	complexType PhysicalValueType refer to 9.5.2.4	Optional element: Estimated or calculated time until bulk charge (approx. 80 % SOC) is complete
EVTargetVoltage	complexType PhysicalValueType refer to 9.5.2.4	Target voltage requested by the EV.

NOTE 1 Values for EVSEMaximumVoltageLimit, CurrentLimit and Power Limit are specified in IEC 61851-23.

NOTE 2 For details on the processing of “EVMaximumCurrentLimit” and “EVMaximumVoltageLimit” by the EVSE refer to IEC 61851-23.

[V2G-DC-972] In the scope of this document, the optional element “EVMaximumPowerLimit” shall be used in all messages.

9.4.2.4.3 Current demand response

After receiving the Current Demand Request message of the EVCC, the SECC sends the current demand response message informing the EV about EVSE status and present EVSE output voltage and current.

[V2G-DC-281] The EVCC and the SECC shall implement the mandatory message and message elements as defined in Figure 54.

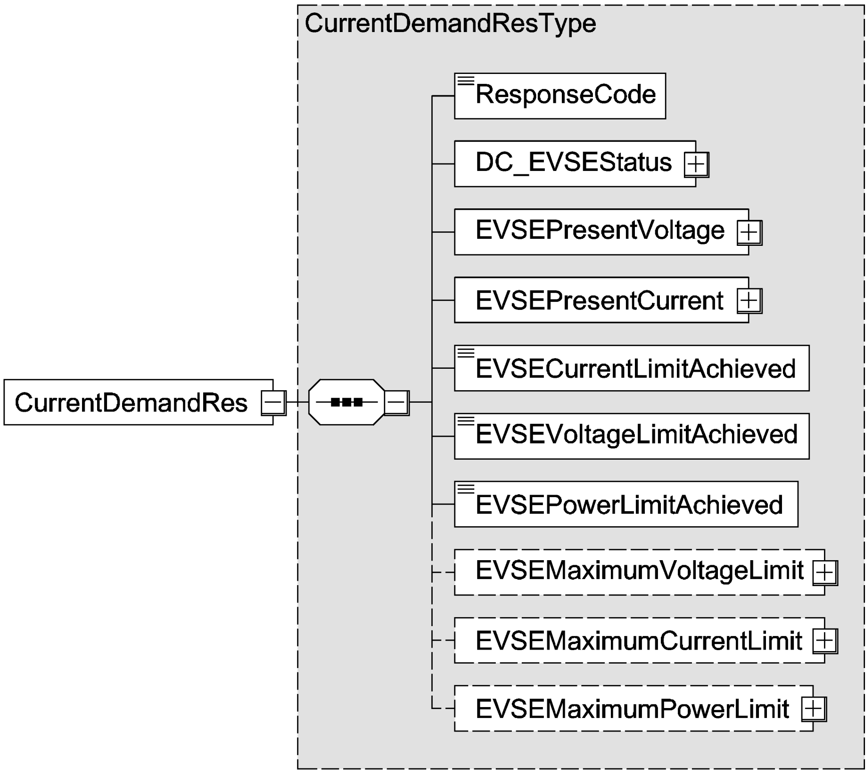


Figure 54 — Schema diagram — CurrentDemandRes

[V2G-DC-282] The message elements of this message shall be used as defined in Table 48.

Table 48 — Semantics and type definition for CurrentDemandRes

Element Name	Type	Semantics
ResponseCode	simpleType: responseCodeType enumeration refer to A.6 for the type definition	Response Code indicating the acknowledgment status of any of the V2G messages received by the SECC.
DC_EVSEStatus	complexType: DC_EVSEStatusType refer to 9.5.3.1	Current status of the EVSE
EVSEPresentVoltage	complexType PhysicalValueType refer to 9.5.2.4	Present output voltage of the EVSE.
EVSEPresentCurrent	complexType PhysicalValueType refer to 9.5.2.4	Present output current of the EVSE.
EVSECurrentLimitAchieved	simpleType boolean refer to Annex A for the type definition	If set to TRUE, the EVSE has reached its current limit.

Table 48 (*continued*)

Element Name	Type	Semantics
EVSEVoltageLimitAchieved	simpleType boolean refer to Annex A for the type definition	If set to TRUE, the EVSE has reached its voltage limit
EVSEPowerLimitAchieved	simpleType boolean refer to Annex A for the type definition	If set to TRUE, the EVSE has reached its power limit
EVSEMaximumVoltageLimit	complexType PhysicalValueType refer to 9.5.2.4	See [V2G-DC-948].
EVSEMaximumCurrentLimit	complexType PhysicalValueType refer to 9.5.2.4	See [V2G-DC-949].
EVSEMaximumPowerLimit	complexType PhysicalValueType refer to 9.5.2.4	See [V2G-DC-950].

[V2G-DC-948] In the scope of this document, the optional element “EVSEMaximumCurrentLimit” shall be used.

[V2G-DC-949] In the scope of this document, the optional element “EVSEMaximumVoltageLimit” shall be used.

[V2G-DC-950] In the scope of this document, the optional element “EVSEMaximumPowerLimit” shall be used.

9.4.2.5 Welding detection

9.4.2.5.1 Welding detection handling

Since welded contactors impose a safety risk, the EV may optionally perform a welding detection at the end of the charging sequence, prior to unlocking the connector. To detect welded contactors, the EV steps through a sequence of independently opening and closing its two contactors while evaluating the voltage on the inlet side of the contactors in each step. It may use its own voltage measurement at the inlet or the voltage measurement of the EVSE at its output, or both.

9.4.2.5.2 Welding detection request

The EV sends the welding detection request to obtain from the EVSE the voltage value measured by the EVSE at its output.

[V2G-DC-283] The SECC shall implement the mandatory message and message elements as defined in Figure 55. For the EVCC it is optional to implement this message.

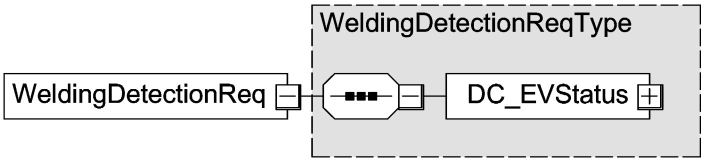


Figure 55 — Schema diagram — WeldingDetectionReq

[V2G-DC-284] The message elements of this message, if implemented, shall be used as defined in Table 49.

Table 49 — Semantics and type definition for WeldingDetectionReq

Element Name	Type	Semantics
DC_EVStatus	complexType: DC_EVStatusType refer to 9.5.3.2	Current status of the EV

9.4.2.5.3 Welding detection response

After receiving the welding detection request message of the EVCC, the SECC sends the welding detection response message informing the EV about the EVSE status and the present EVSE output voltage.

[V2G-DC-285] The SECC shall implement the mandatory message and message elements as defined in Figure 56. For the EVCC it is optional to implement this message.

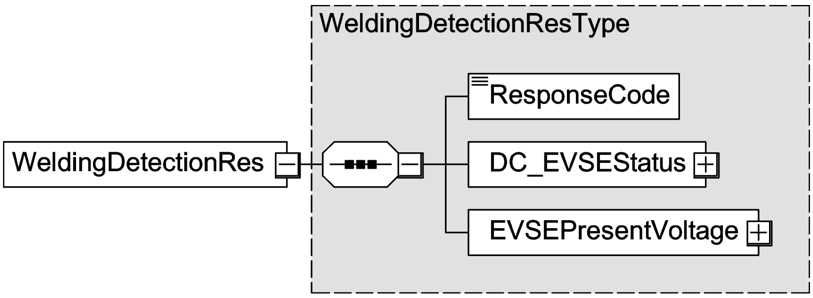


Figure 56 — Schema diagram — WeldingDetectionRes

[V2G-DC-286] The message elements of this message, if implemented, shall be used as defined in Table 50.

Table 50 — Semantics and type definition for WeldingDetectionRes

Element Name	Type	Semantics
ResponseCode	simpleType: responseCodeType enumeration refer to A.6 for the type definition	Response code indicating the acknowledgment status of any of the V2G messages received by the SECC.
DC_EVSEStatus	complexType: DC_EVSEStatusType refer to 9.5.3.1	Current status of the EVSE
EVSEPresentVoltage	complexType PhysicalValueType refer to 9.5.2.4	Present voltage of EVSE

9.5 Complex data types

9.5.1 Overview

In this section complex data types are defined which are used in the messages. Complex data types are composed of several elements which themselves are based on simple data types.

9.5.2 Common types

9.5.2.1 Service tagtype

This type represents a tag for a specific service. It gives a short definition and identification of a specific service.

[V2G-DC-290] The SECC and the EVCC shall implement this type as defined in Figure 57.

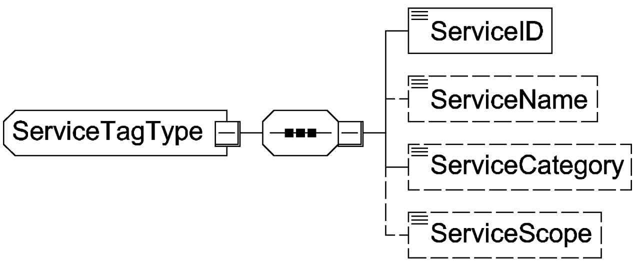


Figure 57 — Schema diagram — ServiceTagType

[V2G-DC-291] The message element shall be used as defined in Table 51.

Table 51 — Semantics and type definition for ServiceTagType

Element Name	Type	Semantics
ServiceID	simpleType: unsignedShort refer to A.6 for the type definition	Unique identifier of the service
ServiceName	simpleType: serviceNameType string (max length: 32) refer to A.6 for the type definition	See [V2G-DC-628],
ServiceCategory	simpleType: serviceCategoryType enumeration refer to A.6 for the type definition	Category of a service, corresponds to the defined services, derived from the base service
ServiceScope	simpleType: serviceScopeType string (max length: 32) refer to A.6 for the type definition	See [V2G-DC-629].

[V2G-DC-628] In the scope of this document, the optional element “ServiceName” shall not be used.

[V2G-DC-629] In the scope of this document, the optional element “ServiceScope” shall not be used.

9.5.2.2 Service Type

[V2G-DC-292] The SECC and the EVCC shall implement this type as defined in Figure 58.

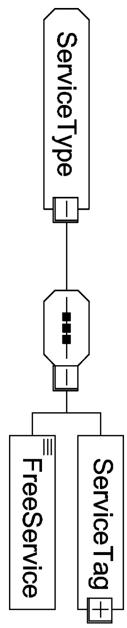


Figure 58 — Schema diagram — ServiceType

[V2G-DC-293] The message element shall be used as defined in Table 52.

Table 52 — Semantics and type definition for ServiceType

Element Name	Type	Semantics
ServiceTag	complexType: serviceTagType refer to 9.5.2.1	Includes the data elements to uniquely identify a service.
FreeService	simpleType: boolean refer to A.6 for the type definition	This element is used by the SECC to indicate if a service can be used by the EVCC free of charge or not. If FreeService is equal to true, the EV can use the offered service without payment. If FreeService is equal to false the service, if used by the EV, will be billed using the payment method negotiated using the payment option message element.

9.5.2.3 ServiceChargeType

EV charging specific service derived from ServiceType (refer to 9.5.2.2) which contains additional information about EVSESSupportedEngeryTransferType(s) offered by the EVSE.

[V2G-DC-294] The SECC and the EVCC shall implement this type as defined in Figure 59.

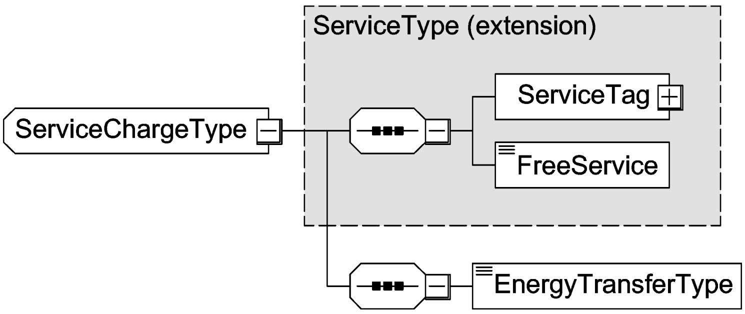


Figure 59 — Schema diagram — ServiceChargeType

[V2G-DC-295] The message element shall be used as defined in Table 53

Table 53 — Semantics and type definition for ServiceChargeType

Element Name	Type	Semantics
EnergyTransferType	simpleType: EVSESupportedEnergyTransferType enumeration refer to A.6 for the type definition	Available charging types or methods, supported by the EVSE.

The definition of the EVSESupportedEnergyTransferType supports connectors according to IEC 62196-3 configuration CC (corresponding to IEC 62196-2 Type 1), IEC 62196-3 configuration DD (corresponding to IEC 62196-2 Type 2), IEC 62196-3 Configuration EE, and IEC 62196-3 configuration FF. Based on the supported connectors, the SECC can offer charging services as defined in Table 54.

[V2G-DC-296] The SECC shall use the EVSESupportedEnergyTransferType as described in Table 54.

Table 54 — Semantics for EVSESupportedEnergyTransferType

EnergyTransferType	Offered charging service
AC_single_phase_core	Not used in the scope of this document.
AC_three_phase_core	Not used in the scope of this document.
DC_core	EVSE supports DC charging using the core pins of a connector
DC_extended	EVSE supports DC charging using the extended pins of an IEC 62196-3 configuration EE or configuration FF connector
DC_combo_core	Not used in the scope of this document.
DC_dual	Not used in the scope of this document.
AC_core1p_DC_extended	Not used in the scope of this document.
AC_single_DC_core	Not used in the scope of this document.
AC_single_phase_three_phase_core_DC_extended	Not used in the scope of this document.
AC_core3p_DC_extended	Not used in the scope of this document.

[V2G-DC-630] If the EVSE supports DC charging using the extended pins of a vehicle connector according to IEC 62196-3 configuration EE or configuration FF, the SECC shall transmit "DC_extended" in EnergyTransferType.

[V2G-DC-631] If the EVSE supports DC charging using the core pins of a connector the SECC shall transmit "DC_core" in EnergyTransferType.

[V2G-DC-632] In the scope of this document, the SECC shall not transmit other values than "DC_extended" and "DC_core" in EnergyTransferType.

9.5.2.4 PhysicalValueType

[V2G-DC-297] The SECC and the EVCC shall implement this type as defined in Figure 60.

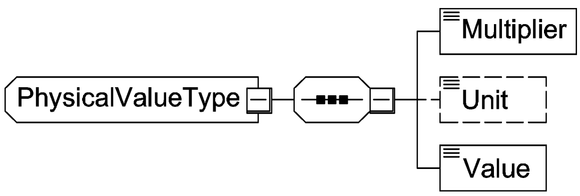


Figure 60 — Schema diagram — PhysicalValueType

[V2G-DC-298] The message element shall be used as defined in Table 55.

Table 55 — Semantics and type definition for PhysicalValueType

Element Name	Type	Semantics
Multiplier	simpleType: unitMultiplierType byte (range: -3 to +3) refer to A.6 for the type definition	The multiplier defines the exponent to base 10 (dec). The final physical value is determined by: value * 10 ^ multiplier [Unit].
Unit	simpleType: unitSymbolType enumeration refer to A.6 for the type definition	Optional element: Unit of the value
Value	simpleType: short refer to A.6 for the type definition	Value which shall be multiplied

NOTE 1 The maximum and minimum transmittable values are calculated using the boundaries of the PhysicalValueType. They are given through the signed 16-bit Integer (Range: -32.768 to +32.767) of the value and the exponent which ranges from -3 to +3. This results in a transmittable value range from -32.768.000 to 32.767.000 in the PhysicalValueType.

NOTE 2 The EVCC and SECC assure keeping sufficient accuracy and resolution of parameters (e.g. EVSEMaximumVoltageLimit) when converting PhysicalValueType to another datatype.

[V2G-DC-952] The EVCC and SECC shall send the optional Element "Unit" in PhysicalValueType.

9.5.2.5 PaymentOptionsType

[V2G-DC-301] The EVCC and the SECC shall implement this type as defined in Figure 61.

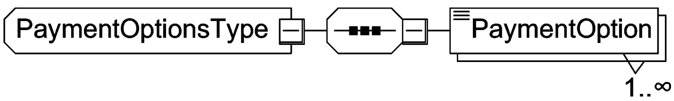


Figure 61 — Schema diagram — PaymentOptionsType

[V2G-DC-302] The message element shall be used as defined in Table 56.

Table 56 — Semantics and type definition for PaymentOptionsType

Element Name	Type	Semantics
PaymentOption	simpleType: paymentOptionType enumeration refer to A.6 for the type definition	This type includes the list of payment options an SECC offers to the EVCC indicating what method could be chosen to pay for the services. The EVCC can only select one payment method for all services used by the EVCC.

[V2G-DC-633] In the scope of this document, only the PaymentOption “ExternalPayment” shall be used.

[V2G-DC-634] The number of PaymentOption elements in the PaymentOptionsType shall be limited to 2.

9.5.2.6 ChargingProfileType

[V2G-DC-303] The SECC and the EVCC shall implement this type as defined in Figure 62.

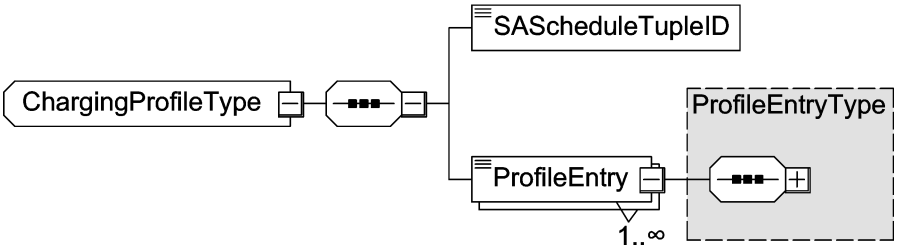


Figure 62 — Schema diagram — XML Schema Definition for the ChargingProfileType

[V2G-DC-304] The message element shall be used as defined in Table 57.

Table 57 — Semantics and type definition for ChargingProfileType

Element Name	Type	Semantics
SAScheduleTupleID	simpleType: SAIDType short refer to A.6 for the type definition	Unique identifier within a charging session referring to the selected SAScheduleTuple element from the SAScheduleListType (see 9.5.2.7).
ProfileEntry	complexType: ProfileEntryType, refer to 9.5.2.7	Element used for encapsulating an individual charging profile entry of the charge schedule

[V2G-DC-305] The value of the SAScheduleTupleID element shall be equal to one of the values of the SAScheduleTupleID elements in the list of SAScheduleTuple elements (see 9.5.2.8) provided with the Charge Parameter Discovery Response message (see 9.4.1.6.3).

[V2G-DC-306] The SAScheduleTupleID element shall identify the selected SAScheduleTuple element (see 9.5.2.8) in the list of SAScheduleTuple elements (see 9.5.2.7) provided in the Charge Parameter Discovery Response message (see 9.4.1.6.3).

[V2G-DC-307] The number of ProfileEntry elements in the ChargingProfileType shall be limited to 24.

9.5.2.7 ProfileEntryType

[V2G-DC-308] The SECC and the EVCC shall implement this type as defined in Figure 63.

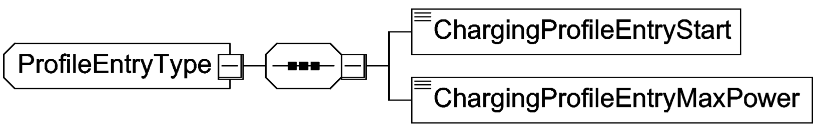


Figure 63 — Schema diagram — ProfileEntryType

[V2G-DC-309] The message element shall be used as defined in Table 58.

Table 58 — Semantics and type definition for ProfileEntryType

Element Name	Type	Semantics
ChargingProfileEntryStart	simpleType: unsignedInt refer to A.6 for the type definition	Time when chargingProfileEntry starts to be valid. Offset in seconds from NOW.
ChargingProfileEntryMaxPower	simpleType: PMaxType short refer to A.6 for the type definition	Maximum power in Watt consumed by the EV within the current charging profile entry (beginning from ChargingProfileEntryStart)

[V2G-DC-310] The value of the ChargingProfileEntryStart element shall be defined as point in time when this element of ProfileEntryType starts to be active.

[V2G-DC-311] The value of the next ChargingProfileEntryStart element in the list of elements of ProfileEntryType (see 9.5.2.7) shall be defined as point in time when this element of ProfileEntryType becomes inactive.

NOTE [V2G-DC-310] and [V2G-DC-311] define the period of time an element of ProfileEntryType is active.

[V2G-DC-312] The last element in the list of elements of type ProfileEntryType is active until the list is updated.

[V2G-DC-313] The value of the ChargingProfileEntryMaxPower element shall be defined as maximum power in Watt consumed by the EV within the active period of an element of type ProfileEntryType.

[V2G-DC-314] The values of the ChargingProfileEntryMaxPower element shall be equal to or smaller than the limits in respective elements of the PMaxScheduleType (see 9.5.2.10) provided in the charge parameter discovery response message.

9.5.2.8 SAScheduleListType

[V2G-DC-315] The SECC and the EVCC shall implement this type as defined in Figure 64.

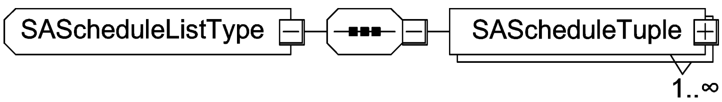


Figure 64 — Schema diagram — SAScheduleListType

[V2G-DC-316] The message element shall be used as defined in Table 59.

Table 59 — Semantics and type definition for SAScheduleListType

Element Name	Type	Semantics
SAScheduleTuple	complexType: SAScheduleTupleType refer to 9.5.2.9	Includes several tuples of schedules from secondary actors

[V2G-DC-317] The number of SAScheduleTuple elements in the SAScheduleListType shall be limited to three (3).

[V2G-DC-318] The EVCC may implement a mechanism to compare different SAScheduleTuple elements to optimize the charge schedule considering any given kind of cost.

[V2G-DC-319] The first SAScheduleTuple element in the SAScheduleListType shall be defined as default SASchedule.

[V2G-DC-320] If the EVCC is not capable of comparing different SAScheduleTuple elements or comparison fails, the EVCC shall choose the default SAScheduleTuple according to [V2G-DC-319].

9.5.2.9 SAScheduleTupleType

[V2G-DC-321] The SECC and the EVCC shall implement this type as defined in Figure 65.

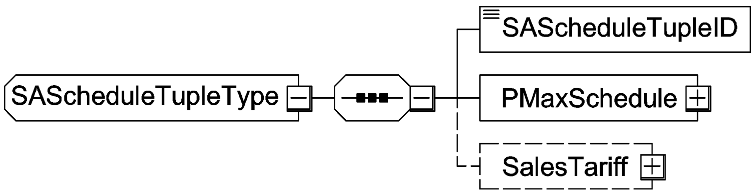


Figure 65 — Schema diagram — SAScheduleTupleType

[V2G-DC-322] The message element shall be used as defined in Table 60.

Table 60 — Semantics and type definition for SAScheduleTupleType

Element Name	Type	Semantics
SAScheduleTupleID	simpleType: SAIDType short refer to A.6 for the type definition	Unique identifier within a charging session for a SAScheduleTuple element
PMaxSchedule	complexType: PMaxScheduleType refer to 9.5.2.10	Encapsulating element describing all relevant details for one PMaxSchedule from the secondary actor
SalesTariff	—	See [V2G-DC-554].

[V2G-DC-554] In the scope of this document, the element “SalesTariff” shall not be used.

[V2G-DC-323] The SAScheduleTupleID element shall be unique within all SAScheduleTuple elements in the SAScheduleListType and uniquely identifies a tuple of PMaxSchedule and SalesTariff (this optional element is not used here) elements during the entire charging session.

[V2G-DC-324] The SECC shall provide a PMaxSchedule element based upon the limits of the local installation if no secondary actor provides a grid schedule.

[V2G-DC-559] Since for DC charging according to this document, the EVCC is not able to provide a planned departure time, SAScheduleList shall provide PMaxSchedule (refer to 9.5.2.10) covering at least 24 hours.

9.5.2.10 PMaxScheduleType

[V2G-DC-326] The SECC and the EVCC shall implement this type as defined in Figure 66.

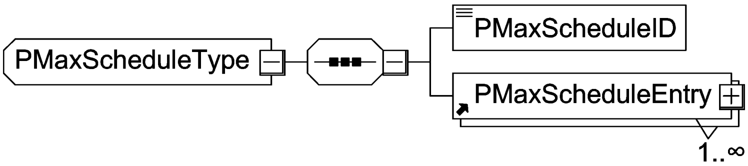


Figure 66 — Schema diagram — PMaxScheduleType

[V2G-DC-327] The message element shall be used as defined in Table 61.

Table 61 — Semantics and type definition for PMaxScheduleType

Element Name	**Type**	**Semantics**
PMaxScheduleID	simpleType: SAIDType short refer to A.6 for the type definition	Unique identifier for an element of type PMaxScheduleType across a charging session.
PMaxScheduleEntry	complexType: PMaxScheduleEntryType refer to 9.5.2.11	List of PMaxScheduleEntry elements

[V2G-DC-328] The value of the PMaxScheduleID element shall uniquely identify an element of type PMaxScheduleType during the entire charging session.

[V2G-DC-329] The number of PMaxScheduleEntry elements in the PMaxScheduleType shall be limited to twelve (12).

9.5.2.11 PMaxScheduleEntryType

[V2G-DC-330] The SECC and the EVCC shall implement this type as defined in Figure 67.

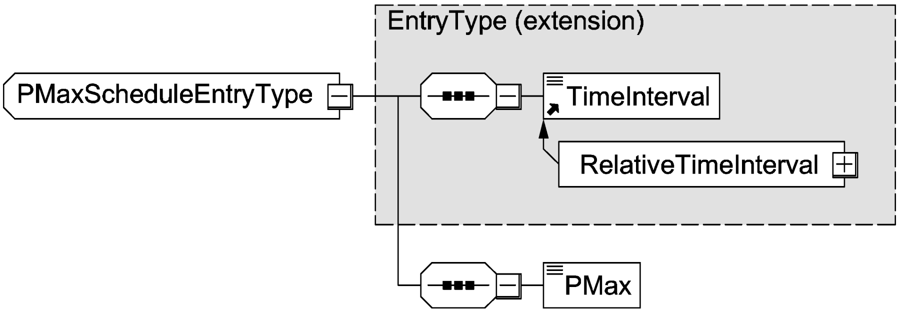


Figure 67 — Schema diagram — PMaxScheduleEntryType

[V2G-DC-331] The message element shall be used as defined in Table 62.

Table 62 — Semantics and type definition for PMaxScheduleEntryType

Element Name	Type	Semantics
RelativeTimeInterval	complexType: RelativeTimeInterval substitutes abstract element TimeInterval. refer to 9.5.2.12	Extends the TimeIntervalType and defines the time interval the PMaxScheduleEntry is valid for based upon relative times.
PMax	simpleType: PMaxType short refer to A.6 for the type definition	Defines maximum amount of power to be drawn from the EVSE outlet the vehicle is connected to.

[V2G-DC-332] Any extension of the TimeInterval element shall define the active period of time for the respective parent element of type PMaxScheduleEntryType.

[V2G-DC-333] The PMax element shall define the maximum amount of power to be drawn from the EVSE outlet when the element of type PMaxScheduleEntryType is active.

9.5.2.12 RelativeTimeIntervalType

[V2G-DC-334] The SECC and the EVCC shall implement this type as defined in Figure 68.

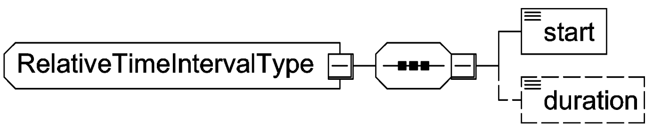


Figure 68 — Schema diagram — RelativeTimeIntervalType

[V2G-DC-335] The message element shall be used as defined in Table 63.

Table 63 — Semantics and type definition for RelativeTimeIntervalType

Element Name	Type	Semantics
duration	simpleType unsignedInt refer to A.6 for the type definition	Optional Element: Duration of the interval, in seconds.
start	simpleType unsignedInt refer to A.6 for the type definition	Start of the interval, in seconds from NOW.

[V2G-DC-336] The value of the start element shall be defined in seconds from NOW.

[V2G-DC-337] The value of the start element shall simultaneously define the start time of this interval and the stop time of the previous interval.

[V2G-DC-338] The value of the duration element shall be defined as period in seconds.

[V2G-DC-339] The duration element shall only be used for the last interval of the PMaxSchedule (see 9.5.2.10).

NOTE It indicates the end of the coverage time of the delivered PMaxSchedule information.

9.5.2.13 SelectedServiceListType

[V2G-DC-340] The EVCC and the SECC shall implement this type as defined in Figure 69.

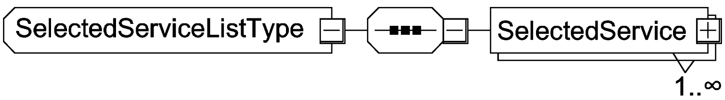


Figure 69 — Schema diagram — SelectedServiceListType

[V2G-DC-341] The message element shall be used as defined in Table 64.

Table 64 — Semantics and type definition for SelectedServiceListType

Element Name	Type	Semantics
SelectedService	complexType: SelectedServiceType refer to 9.5.2.14	This element is used to indicate the selected ServiceID and the associated parameterSet.

[V2G-DC-635] The number of SelectedService elements in the SelectedServiceListType shall be limited to 1.

9.5.2.14 SelectedServiceType

[V2G-DC-342] The EVCC and the SECC shall implement this type as defined in Figure 70.

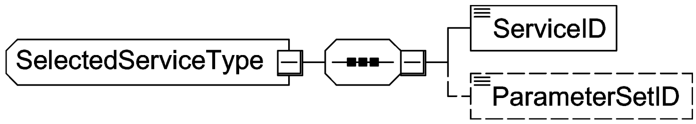


Figure 70 — Schema Diagram — SelectedServiceType

[V2G-DC-343] The message element shall be used as defined in Table 65.

Table 65 — Semantics and type definition for SelectedServiceType

Element Name	Type	Semantics
ServiceID	simpleType: unsignedShort refer to A.6 for the type definition	Unique identifier of the service
ParameterSetID	—	See [V2G-DC-555]

[V2G-DC-555] In the scope of this document, the element “ParameterSetID” shall not be used.

9.5.3 DC specific types

9.5.3.1 DC_EVSEStatusType

[V2G-DC-344] The EVCC and the SECC shall implement this type as defined in Figure 71.

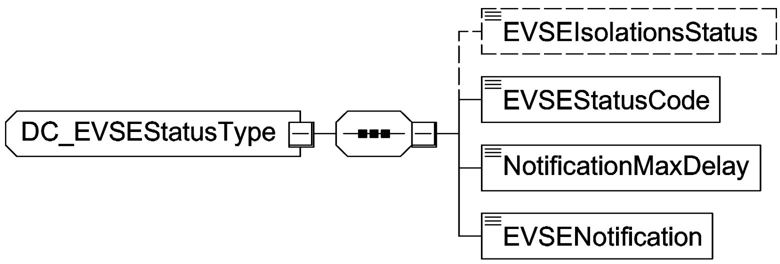


Figure 71 — Schema diagram — DC_EVSEStatusType

[V2G-DC-345] The message element shall be used as defined in Table 66.

Table 66 — Semantics and type definition for DC_EVSEStatusType

Element Name	Type	Semantics
EVSEIsolationStatus	simpleType: isolationLevelType enumeration refer to A.6 for the type definition	Optional Element: Indicates the isolation condition (result of the isolation test/monitoring). Refer to Table 67 for details.
EVSEStatusCode	simpleType: DC_EVSEStatusCodeType enumeration refer to A.6 for the type definition	Indicates if the internal state of the EVSE. Refer to Table 68 for details.

Table 66 (*continued*)

Element Name	Type	Semantics
NotificationMaxDelay	simpleType: unsignedInt refer to A.6 for the type definition	The SECC uses the NotificationMaxDelay element in the EVSEStatus to indicate the time until it expects the EVCC to react on the action request indicated in EVSENotification. If the target time is not in the future, the EVCC is expected to perform the action immediately.
EVSENotification	simpleType: EVSENotificationType enumeration refer to A.6 for the type definition	This value is used by the SECC to influence the behaviour of the EVCC. The EVSENotification contains an action that the SECC wants the EVCC to perform. The requested action is expected by the EVCC until the time provided in NotificationMaxDelay. If the target time is not in the future, the EVCC is expected to perform the action immediately. During normal operation the value of EVSENotification is set to “None”.

[V2G-DC-346] The SECC shall use the EVSEIsolationStatus as described in Table 67.

NOTE For further information on the Isolation Status, refer to IEC 61851-23.

[V2G-DC-500] For DC charging according to this document, the value of EVSENotification shall always be set to “None”.

[V2G-DC-636] For DC charging according to this document, the EVCC shall ignore the value of the parameter NotificationMaxDelay in the EVSEStatus.

Table 67 — Semantics & Types of Elements in the isolationLevelType

Element Name	Semantics
Invalid	An isolation test has not been carried out.
Valid	The isolation test has been carried out successfully and did not result in an isolation warning or fault.
Warning	The measured isolation resistance is below the warning level defined in IEC 61851-23.
Fault	The measured isolation resistance is below the fault level defined in IEC 61851-23.

[V2G-DC-347] The SECC shall use the EVSEStatusCode as described in Table 68.

Table 68 — Semantics and type definition for DC_EVSEStatusCodeType

Element Name	Semantics
EVSE_NotReady	Not authorized, StandBy, on maintenance, …
EVSE_Ready	Standard value used during normal operation
EVSE_Shutdown	Charger shutdown, customer-initiated shutdown

Table 68 (*continued*)

Element Name	Semantics
EVSE_UtilityInterruptEvent	Utility interrupt event, utility or equipment operator has requested a temporary reduction in load.
EVSE_IsolationMonitoringActive	After the charging station has confirmed HV isolation internally, it will remain in this state until the cable isolation integrity is checked
EVSE_EmergencyShutdown	Charging system incompatibility, emergency shutdown or “E-Stop” button pressed at charging station.
EVSE_Malfunction	A non-recoverable charger fault has occurred (isolation failure, etc.)
Reserved 8-C	Reserved for future use.

[V2G-DC-637] All EVSEStatusCodes in Table 68 for which there are no explicit requirements in this document are for information purpose only. They shall not influence the EV charging process.

[V2G-DC-638] EVSEStatusCode equal to “EVSE_Ready” shall be used unless a different value shall be used according to another requirement in this document.

In the context of this document, “EVSE_Shutdown” is used by the SECC to trigger a termination of the charging process in situations other than an emergency shutdown, see 9.7.4.1.6. “EVSE_EmergencyShutdown” may be used by the SECC to inform the EVCC that the EVSE is in the process of, or has just executed, an emergency shutdown. However, to force the EV to participate in the emergency shutdown, the SECC does not use this value, but instead uses the CPL in accordance with IEC 61851-23. In all other cases, the expected value for EVSEStatusCode is “EVSE_Ready” or “EVSE_IsolationMonitoringActive”.

9.5.3.2 DC_EVStatusType

[V2G-DC-348] The EVCC and the SECC shall implement this type as defined in Figure 72.

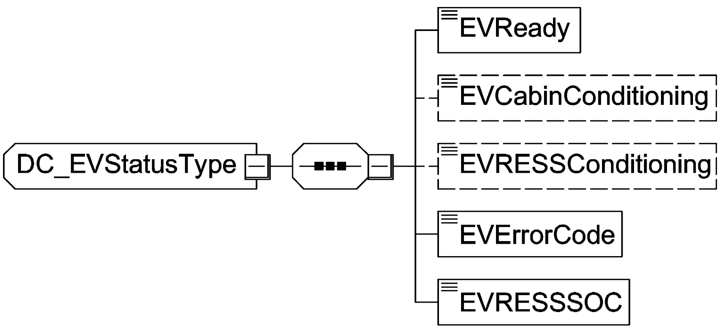


Figure 72 — Schema diagram — DC_EVStatusType

[V2G-DC-349] The message element shall be used as defined in Table 69.

Table 69 — Semantics and type definition for DC_EVStatusType

Element Name	Type	Semantics
EVReady	simpleType: boolean refer to A.6 for the type definition	If set to TRUE, the EV is ready to charge.
EVCabinConditioning	simpleType: boolean refer to A.6 for the type definition	Optional Element: Vehicle Cabin Conditioning, the EV is using energy from the DC supply to heat or cool the passenger compartment.
EVRESSConiditioning	simpleType: boolean refer to A.6 for the type definition	Optional Element: Vehicle RESS Conditioning, the vehicle is using energy from the DC charger to condition the RESS to a target temperature.
EVErrorCode	simpleType: DC_EVErrorCodeType enumeration refer to A.6 for the type definition	Indicates the EV internal status. Refer to Table 70 for details.
EVRESSSOC	simpleType: percentValueType byte (range: 0-100) refer to A.6 for the type definition	State of charge of the EV’s battery (RESS)

[V2G-DC-639] For DC charging according to this document, the elements EVReady, EVCabinConditioning, and EVRESSConiditioning shall not affect the charging session. However, they may be used for customer information.

[V2G-DC-903] For DC_EVStatus as defined in Table 69, the EVCC should not send:

— EVCabinConditioning (optional) [V2G-DC-639]

— EVRESSConditioning (optional) [V2G-DC-639]

[V2G-DC-831] The EV may send any value in EVReady according [V2G-DC-639].

[V2G-DC-832] The EVSE shall not use the parameter EVReady to have any impact on the charging process according to [V2G-DC-639].

[V2G-DC-904] The SECC shall not use the parameters EVErrorCode, EVCabinConditioning and EVRESSConditioning to have any impact on the charging process.

[V2G-DC-350] The EVCC shall use the EVErrorCodes as described in Table 70.

Table 70 — Semantics and type definition for EVErrorCodeType

Element Name	Semantics
NO_ERROR	Default value when EV has no error detected
FAILED_RESSTemperatureInhibit	Battery temperature inhibit, battery too hot/cold to accept charge

Table 70 (*continued*)

Element Name	Semantics
FAILED_EVShiftPosition	Vehicle shift position, vehicle is not in Park
FAILED_ChargerConnectorLockFault	Charger connector lock fault, vehicle has not detected the charge cord connector locked into the inlet or failure where connector cannot be unlocked from the charging inlet.
FAILED_EVRESSMalfunction	Vehicle RESS malfunction, any non-recoverable fault or error condition of the vehicle RESS.
FAILED_ChargingCurrentdifferential	Charging current differential, indication that vehicle has stopped the communication session after detecting that the charging station is not able to maintain an output current that fulfils the current request.
FAILED_ChargingVoltageOutOfRange	Charging voltage out of range, Indication that vehicle has stopped the communication session after detecting that the RESS is either under or above normal operating voltage range.
Reserved A-C	Reserved for future use.
FAILED_ChargingSystemIncompatibility	Charging system incompatibility if the vehicle determines that the charging station is incompatible. Using this value is optional; as an alternative, the vehicle can use EVReady in DC_EVStatusType equal to “FALSE”
NoData	No data. Only used when vehicle has not yet determined its operating state.

[V2G-DC-640] All EVErrorCodes in Table 70 for which there are no explicit requirements in this document are for information purpose only. They may be used for information to the customer, but they shall not influence the EVSE charging process.

NOTE In the context of this document this means that the EVSE does not change its behavior based on the value of EVErrorCode. If the EV detects a situation that requires a termination of the charging process, the EV will use other means to execute this shutdown, e.g. ramp down the current request, or, in case an emergency shutdown is required, change to CP State B in accordance with IEC 61851-1 and IEC 61851-23. In all other cases, the expected value for EVErrorCode is “NO_ERROR”.

9.5.3.3 DC_EVChargeParameterType

[V2G-DC-351] The EVCC and the SECC shall implement this type as defined in Figure 73.

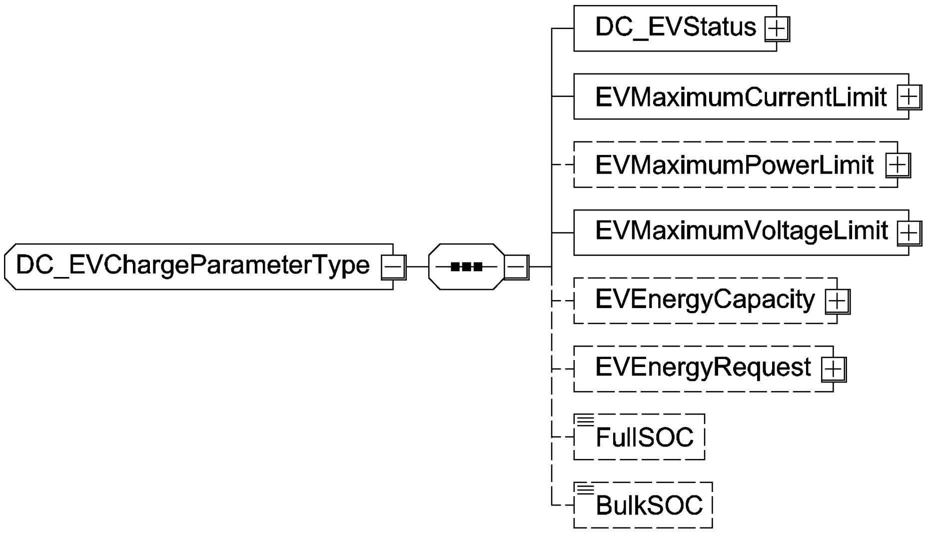


Figure 73 — Schema diagram — DC_EVChargeParameterType

[V2G-DC-352] The message element shall be used as defined in Table 71.

Table 71 — Semantics and type definition for DC_EVChargeParameterType

Element Name	Type	Semantics
DC_EVStatus	complexType DC_EVStatusType refer to 9.5.3.2	Current status of the EV
EVMaximumCurrentLimit	complexType PhysicalValueType refer to 9.5.2.4	Maximum current supported by the EV
EVMaximumPowerLimit	complexType PhysicalValueType refer to 9.5.2.4	Optional Element: Maximum power supported by the EV
EVMaximumVoltageLimit	complexType PhysicalValueType refer to 9.5.2.4	Maximum voltage supported by the EV
EVEnergyCapacity	complexType PhysicalValueType refer to 9.5.2.4	Optional Element: Maximum energy capacity supported by the EV
EVEnergyRequest	complexType PhysicalValueType refer to 9.5.2.4	Optional Element: Amount of energy the EV requests from the EVSE

Table 71 (*continued*)

Element Name	Type	Semantics
FullSOC	simpleType: percentValueType byte (range: 0-100) refer to A.6 for the type definition	Optional Element: SOC at which the EV considers the battery to be fully charged
BulkSOC	simpleType: percentValueType byte (range: 0-100) refer to A.6 for the type definition	Optional Element: SOC at which the EV considers a fast-charging process to end

9.5.3.4 DC_EVSEChargeParameterType

[V2G-DC-353] The EVCC and the SECC shall implement this type as defined in Figure 74.

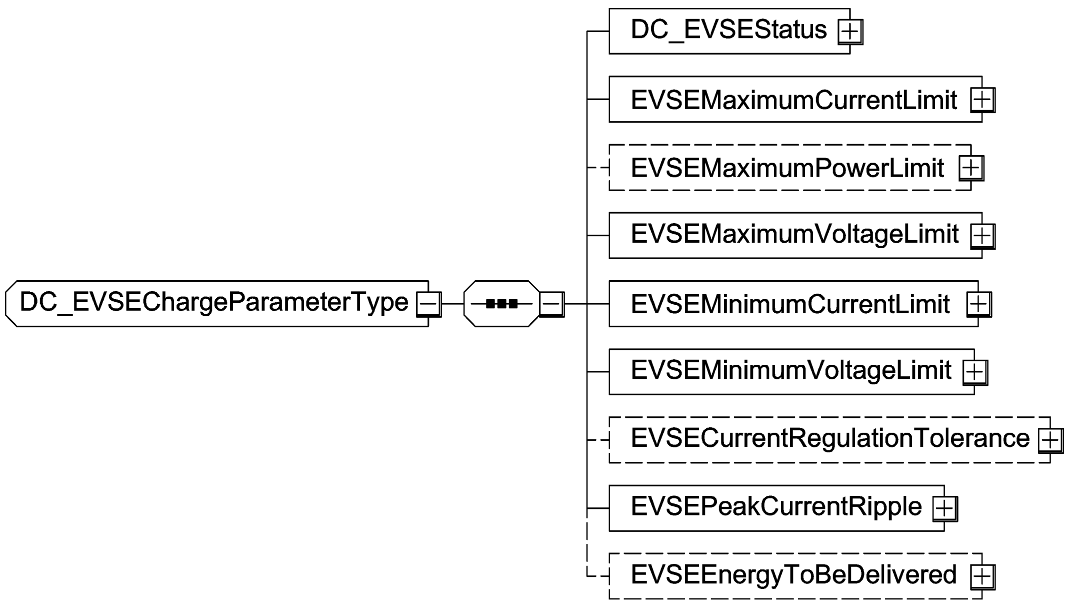


Figure 74 — Schema diagram — DC_EVSEChargeParameterType

[V2G-DC-354] The message element shall be used as defined in Table 72.

Table 72 — Semantics and type definition for DC_EVSEChargeParameterType

Element Name	Type	Semantics
DC_EVSEStatus	complexType DC_EVSEStatusType refer to 9.5.3.1	Current status of the EVSE
EVSEMaximumCurrentLimit	complexType PhysicalValueType refer to 9.5.2.4	Maximum current the EVSE can deliver

Table 72 (*continued*)

Element Name	Type	Semantics
EVSEMaximumPowerLimit	complexType PhysicalValueType refer to 9.5.2.4	Optional Element: Maximum power the EVSE can deliver
EVSEMaximumVoltageLimit	complexType PhysicalValueType refer to 9.5.2.4	Maximum voltage the EVSE can deliver
EVSEMinimumCurrentLimit	complexType PhysicalValueType refer to 9.5.2.4	Minimum current the EVSE can deliver with the expected accuracy
EVSEMinimumVoltageLimit	complexType PhysicalValueType refer to 9.5.2.4	Minimum voltage the EVSE can deliver with the expected accuracy
EVSECurrentRegulationTolerance	complexType PhysicalValueType refer to 9.5.2.4	Optional Element: Absolute magnitude of the regulation tolerance of the EVSE
EVSEPeakCurrentRipple	complexType PhysicalValueType refer to 9.5.2.4	Peak-to-peak magnitude of the current ripple of the EVSE
EVSEEnergyToBeDelivered	complexType PhysicalValueType refer to 9.5.2.4	Optional Element: Amount of energy to be delivered by the EVSE

[V2G-DC-643] In the scope of this document, if the element “EVSEEnergyToBeDelivered” is contained in the message ChargeParameterDiscoveryRes, it shall represent the amount of energy that the EVSE will be able to provide to the EV throughout the entire charging process.

9.5.3.5 DC_EVPowerDeliveryParameterType

[V2G-DC-355] The EVCC and the SECC shall implement this type as defined in Figure 75.

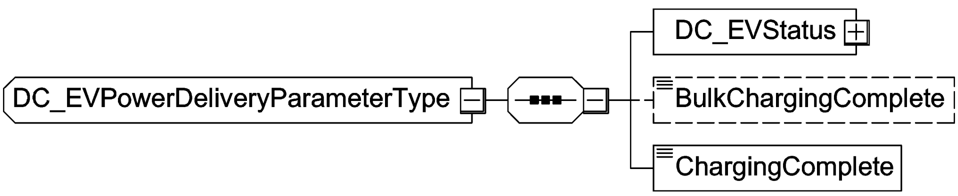


Figure 75 — Schema diagram — DC_EVPowerDeliveryParameterType

[V2G-DC-356] The message element shall be used as defined in Table 73.

Table 73 — Semantics and type definition for DC_EVPowerDeliveryParameterType

Element Name	Type	Semantics
DC_EVStatus	complexType DC_EVStatusType refer to 9.5.3.2	Current status of the EV.
BulkChargingComplete	simpleType: Boolean refer to A.6 for the type definition	Optional Element: If set to TRUE, the EV indicates that bulk charge (approx. 80 % SOC) is complete.
ChargingComplete	simpleType: Boolean refer to A.6 for the type definition	If set to TRUE, the EV indicates that full charge (100 % SOC) is complete.

9.5.4 Use of optional schema elements

Table 74 lists XML elements that are optional themselves or that contain optional sub elements as per the schema definition. The marker in the columns EVCC and EVSE, respectively, indicate as follows whether the use of the corresponding schema element is optional or mandatory within this document:

M means that the respective schema element is mandatory as per the schema definition, or that it is optional as per the schema definition but the use of it is mandatory in the corresponding message in the scope of this document.

O means that the respective schema element is optional as per the schema definition, and the use of it is optional in the corresponding message in the scope of this document.

N/A means that the respective schema element is optional as per the schema definition, but that it shall not be used in the corresponding message in the scope of this document.

MP means that the respective schema element is optional as per the schema definition; however, if it is used by the sender of the message, the receiver of the message shall process this element.

Table 74 — Optional schema elements

V2G Message								
Name	**Parameter Level**						**DC Charging**	
	1	**2**	**3**	**4**	**5**	**6**	**EVCC**	**SECC**
Supported App Protocol Res							M	M
	SchemaID						MP	O
Session Setup Res							M	M
	DateTime-Now						O	O

Table 74 (*continued*)

V2G Message								
Name	**Parameter Level**						**DC Charging**	
	1	**2**	**3**	**4**	**5**	**6**	**EVCC**	**SECC**
V2G_Message							M	M
	Header						M	M
		Notification					N/A	N/A
		Signature					N/A	N/A
	Body						M	M
		BodyElement (Substituted by correspon-ding message)					M	M
Service Discovery Req							M	M
	Service Scope						N/A	N/A
	Service Category						O	MP
Service Discovery Res							M	M
	Charge Service						M	M
		ServiceTag					M	M
			ServiceName				N/A	N/A
			ServiceCate-gory				M	M
			ServiceScope				N/A	N/A
	ServiceList						N/A	N/A
Service Payment Selection Req							M	M
	SelectedSer-viceList						M	M
		SelectedSer-vice					M	M
			ServiceTag				M	M
				Parameter-SetID			N/A	N/A
Contract Authentication Req							M	M
	GenChal-lenge						N/A	N/A
	Id						N/A	N/A

Table 74 (*continued*)

V2G Message								
Name	**Parameter Level**						**DC Charging**	
	1	**2**	**3**	**4**	**5**	**6**	**EVCC**	**SECC**
Charge Parameter DiscoveryReq							M	M
	AC_EV Charge Parameter						N/A	N/A
	DC_EV Charge Parameter						M	M
		DC_EVStatus					M	M
			EVCabinCon-ditioning				O	O
			EVRESSConi-ditioning				O	O
		EVMaximum-PowerLimit					O	O
		EVEnergyCa-pacity					O	O
		EVEnergyRe-quest					O	O
		FullSOC					O	O
		BulkSOC					O	O

Table 74 (*continued*)

V2G Message								
Name	**Parameter Level**						**DC Charging**	
	1	**2**	**3**	**4**	**5**	**6**	**EVCC**	**SECC**
Charge Parameter Discovery Res							M	M
	SASchedule-List						M	M
		SASchedu-leTuple					M	M
			SASchedu-leTupleID				M	M
			PMaxSche-dule				M	M
				PMAaxSche-duleEntry			M	M
					Relative-TimeInterval		M	M
						duration	MP	O
					Pmax		M	M
			SalesTariff				N/A	N/A
	AC_EVSE Charge Parameter						N/A	N/A
	DC_EVSE Charge Parameter						M	M
		DC_EVSESta-tus					M	M
			EVSEIsolati-onStatus				O	O
		EVSECurrent-RegulationTo-lerance					O	O
		EVSEMaxi-mumPowerLi-mit					MP	M
		EVSEEnergy-ToBeDelive-red					O	O

Table 74 (*continued*)

V2G Message								
Name	**Parameter Level**						**DC Charging**	
	1	**2**	**3**	**4**	**5**	**6**	**EVCC**	**SECC**
Power Delivery Req							M	M
	Charging-Profile						O	MP
	DC_EV Power Delivery Parameter						M	M
		DC_EVStatus					M	M
			EVCabinCon-ditioning				O	O
			EVRESSConi-ditioning				O	O
		BulkCharging-Complete					O	O
Power Delivery Res							M	M
	AC_EVSE-Status						N/A	N/A
	DC_EVSE-Status						M	M
		EVSEIsolati-onStatus					O	O
Cable Check Req							M	M
	DC_EVSta-tus						M	M
		EVCabinCon-ditioning					O	O
		EVRESSConi-ditioning					O	O
Cable Check Res							M	M
	DC_EVSE-Status						M	M
		EVSEIsolati-onStatus					O	O
PreChargeReq							M	M
	DC_EVSta-tus						M	M
		EVCabinCon-ditioning					O	O
		EVRESSConi-ditioning					O	O

Table 74 (*continued*)

V2G Message								
Name	**Parameter Level**						**DC Charging**	
	1	**2**	**3**	**4**	**5**	**6**	**EVCC**	**SECC**
PreChargeRes							M	M
	DC_EVSE-Status						M	M
		EVSEIsolati-onStatus					O	O
Current Demand Req							M	M
	DC_EVSta-tus						M	M
		EVCabinCon-ditioning					O	O
		EVRESSConi-ditioning					O	O
	EVMaxi-mumVolta-geLimit						O	MP
	EVMaxi-mumCur-rentLimit						O	MP
	EVMaxi-mumPower-Limit						O	O
	BulkChar-gingCom-plete						O	O
	Remai-ningTimeT-oFullSoC						O	O
	Remai-ningTime-ToBulkSoC						O	O
Current Demand Res							M	M
	DC_EVSE-Status						M	M
		EVSEIsolati-onStatus					O	O
	EVSEMaxi-mumVolta-geLimit						MP	O
	EVSEMaxi-mumCur-rentLimit						MP	O
	EVSEMaxi-mumPower-Limit						MP	O

Table 74 (*continued*)

V2G Message								
Name	**Parameter Level**						**DC Charging**	
	1	**2**	**3**	**4**	**5**	**6**	**EVCC**	**SECC**
Welding Detection Req							O	MP
	DC_EVStatus						M	M
		EVCabinConditioning					O	O
		EVRESSConiditioning					O	O
Welding Detection Res							O	MP
	DC_EVSEStatus						M	M
		EVSEIsolationStatus					o	O

9.6 Session timing and error handling

9.6.1 Overview

This subclause describes the timing and error handling for the V2G communication session. The error handling is, beside other tasks, based on timers enabling the EVCC and the SECC to monitor the V2G message exchange. For the detection of missing or delayed messages, the EVCC and the SECC use predefined timeout values as error criteria. Whenever a timer is equal of larger than the related timeout, the related error handling is processed accordingly.

It is important to understand that all timers, timeouts, and performance times specified in this subclause are merely means to describe the temporal behaviour of the EVCC and the SECC required by this standard. While the EVCC and the SECC are required to exhibit a temporal behavior and an error handling behavior as described in this subclause by means of timers, timeouts, and performance times, it is not necessary that all timers and timeouts be implemented exactly as described herein, or be implemented at all, resp.

A timer counts the duration from the last time it was reset and then started. The instantaneous value of a timer is the duration from its most recent reset and start time to the present time. The monitoring of the V2G communication message exchange is based on two-timer categories:

— Message timer: Monitors the exchange of a request message and the corresponding response message (request-response-pair);

— Sequence timer: Monitors a response-request message sequence.

To enable error handling for a V2G communication session setup, the EVCC monitors the time between plug-in and the reception of the SessionSetupRes message and the PowerDeliveryRes message, respectively. This allows the EVCC to decide about a successful or failed communication session setup after the defined timeouts.

The monitoring of a V2G communication session is based on two-timer categories:

— Communication setup timer: Monitors the time from plug-in until reception of the SessionSetupRes message. It allows deciding if the communication setup was successful;

— Ready-to-charge timer: Monitors the time from plug-in until reception of the first PowerDeliveryRes message. It allows deciding if the request for power of the EV was successful.

  The monitoring of the Cable Check is carried out by the EV using the V2G_EVCC_CableCheck_Timer. It is started when the EV requests the EVSE to start the cable check, and ended when the EVSE has finished the cable check, or when the V2G_EVCC_CableCheck_Timer expires.

  The monitoring of the pre-charge is carried out by the EV using the V2G_EVCC_PreCharge_Timer. It is started when the EV starts the pre-charging by sending the first PreChargeReq message, and ended when the pre-charging has finished, indicated by the EV determining that the EVSE output voltage, as measured inside the EV, has sufficiently been adjusted to the EV RESS voltage, or when the V2G_EVCC_PreCharge_Timer expires.

The timers are compared to predefined time values as decision criterion. The EVCC and the SECC distinguish between two types of criteria:

— Timeout: If the specified time is exceeded, the related error handling is initiated;

— Performance time: If the specified time is exceeded, the performance requirement is not fulfilled.

While exceeding a timeout always causes an error handling, exceeding the specified performance time does not necessarily cause error handling. Depending on the system behaviour (e.g. transmission time) no error may occur if the corresponding communication partner does not detect a timeout, but the probability for causing a timeout is high.

[V2G-DC-833] EV and EVSE implementations shall consider the time out requirements in chapter 9.6 (V2G_EVCC_Msg_Timeout (MessageType))

[V2G-DC-834] EV and EVSE implementations may consider the performance time in chapter 9.6 for further internal decisions (V2G_SECC_Msg_ Performance_Time) and error handling.

NOTE For Details see Table 76.

9.6.2 Message sequence and performance timing definitions

Message timers, sequence timers, timeouts, and performance times are defined for EVCC and SECC separately and summarized in Table 75. Timeouts and performance times are parameterized for messages separately to describe different processing times. Table 76 defines the values for each V2G message type.

Table 75 — EVCC and SECC timers, timeouts, performance times

Name	**Type**	**Applicable for**	
		EVCC	SECC
V2G_EVCC_Msg_Timer	Message timer in the EVCC	X	
V2G_SECC_Msg_Timer	Message timer in the SECC		X
V2G_EVCC_Sequence_Timer	Sequence timer in the EVCC	X	
V2G_SECC_Sequence_Timer	Sequence timer in the SECC		X
V2G_SECC_Sequence_TimerCR	Sequence timer in the SECC for CurrentDemand		X

Table 75 (*continued*)

Name	Type	Applicable for	
		EVCC	SECC
V2G_EVCC_Msg_Timeout(MessageType)	Timeout for the message timer The value is defined depending on the parameter MessageType as defined in Table 76.	X	
V2G_SECC_Msg_Performance_Time(MessageType)	Performance time for the message timer The value is defined depending on the parameter MessageType as defined in Table 76.		X
V2G_EVCC_Sequence_Performance_Time	Performance time for the sequence timer as defined in Table 76.	X	
V2G_SECC_Sequence_Timeout	Timeout for the sequence timer as defined in Table 76.		X
V2G_EVCC_Ongoing_Timer	Ongoing Timer	X	
V2G_EVCC_Ongoing_Timeout	Timeout for ongoing timer	X	
V2G_SECC_Ongoing_Performance_Time	Performance time for ongoing timer		X

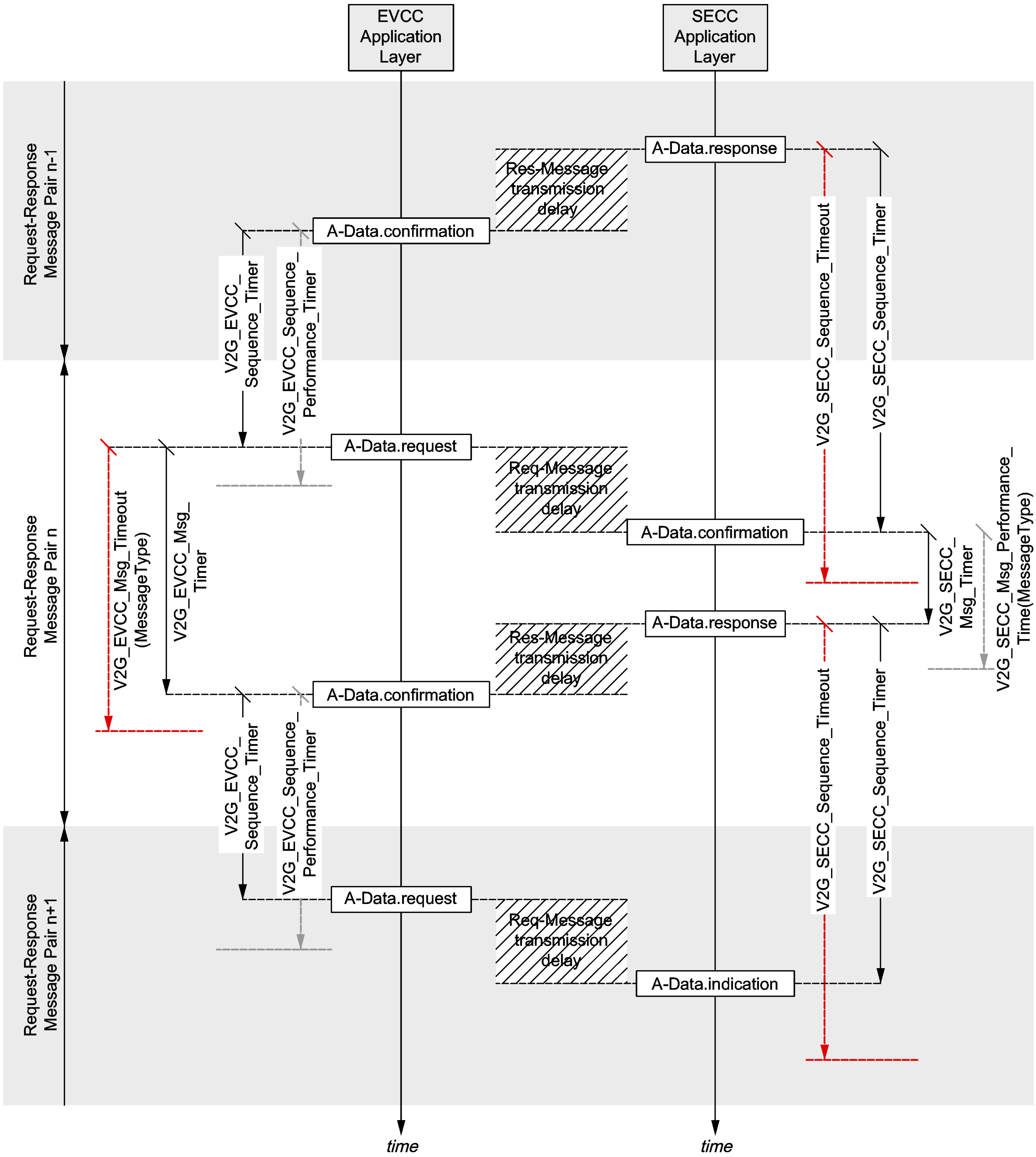
Table 76 — EVCC and SECC message timeouts and performance times

Name	Message type	Value [s]
V2G_EVCC_Msg_Timeout(MessageType)	supportedAppProtocolReq	2
	SessionSetupReq	2
	ServiceDiscoveryReq	2
	ServicePaymentSelectionReq	2
	ContractAuthenticationReq	2
	ChargeParameterDiscoveryReq	2
	PowerDeliveryReq	2
	CableCheckReq	2
	PreChargeReq	2
	CurrentDemandReq	0,5
	WeldingDetectionReq	2
	SessionStopReq	2

Table 76 (*continued*)

Name	**Message type**	**Value [s]**
V2G_SECC_Msg_Performance_Time (MessageType)	SupportedAppProtocolRes	1,5
	SessionSetupRes	1,5
	ServiceDiscoveryRes	1,5
	ServicePaymentSelectionRes	1,5
	ContractAuthenticationRes	1,5
	ChargeParameterDiscoveryRes	1,5
	PowerDeliveryRes	1,5
	CableCheckRes	1,5
	PreChargeRes	1,5
	CurrentDemandRes	0,25
	WeldingDetectionRes	1,5
	SessionStopRes	1,5
V2G_SECC_Sequence_Timeout	(all messages except CurrentDemand)	60
V2G_EVCC_Sequence_Performance_Time	(all messages except CurrentDemand)	59
V2G_SECC_Sequence_TimeoutCR	CurrentDemand	5
V2G_EVCC_Sequence_Performance_TimeCR	CurrentDemand	1
V2G_EVCC_Ongoing_Timeout	Response messages with parameter EVSEProcessing equal to ’Ongoing	60
V2G_SECC_Ongoing_Performance_Time	Response messages with parameter EVSEProcessing equal to ’Ongoing’	55

Figure 76 illustrates how the message timers, sequence timers, timeouts and performance times are applied in the EVCC and the SECC.



Key

Threshold: Timeout (error criteria)
Threshold: Performance time (performance criteria)
Timer

Figure 76 — Message sequence and performance timing

[V2G-DC-357] The EVCC shall implement an EVCC specific timeout supervision according to Table 75 that satisfy the values defined in Table 76.

This EVCC should exhibit a temporal behavior with respect to the so-called “performance times” that satisfy the values defined in Table 76.

[V2G-DC-358] The SECC shall implement a SECC specific timeout supervision according to Table 75 that satisfy the values defined in Table 76.

This SECC should exhibit a temporal behavior with respect to the so-called “performance times” that satisfy the values defined in Table 76.

9.6.3 EVCC timing and error handling for request-response message pairs

[V2G-DC-359] When the EVCC sends a request message, it shall set the timeout V2G_EVCC_Msg_Timeout to the value defined in Table 76 for the respective message. It shall then reset and start the V2G_EVCC_Msg_Timer, and it shall start to continuously monitor the V2G_EVCC_Msg_Timer for reaching or exceeding V2G_EVCC_Msg_Timeout.

NOTE 1 In this document, sending a request message is described by A-DATA.request.

[V2G-DC-975] At any point in time in the V2G communication session while the EVCC monitors the

V2G_EVCC_Msg_Timer, if V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout, the EVCC shall proceed as if the EVCC itself wished to stop the Communication Session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-673].

[V2G-DC-362] When the EVCC receives a response message while V2G_EVCC_Msg_Timer is smaller than V2G_EVCC_Msg_Timeout, it shall stop the V2G_EVCC_Msg_Timer and end monitoring it.

NOTE 2 In this document, receiving a response message is described by A-DATA.confirmation.

[V2G-DC-863] If the EVCC receives a V2G response message with parameter EVSEProcessing equal to ’Ongoing’ for the first time in a response message it shall start the timer V2G_EVCC_Ongoing_Timer and wait for parameter EVSEProcessing equal to ’Finished’.

[V2G-DC-864] If [V2G-DC-863] applies, the EVCC shall stop the V2G communication session when V2G_EVCC_Ongoing_Timer is equal or larger than V2G_EVCC_Ongoing_Timeout and no parameter EVSEProcessing equal to ’Finished’ has been received.

[V2G-DC-1004] If [V2G-DC-863] applies, when the EVCC receives parameter EVSEProcessing equal to ’Finished’, it shall stop the V2G_EVCC_Ongoing_Timer and end monitoring it.

9.6.4 SECC timing and error handling for response-request message sequence

[V2G-DC-957] When the SECC sends a CurrentDemandRes message, it shall set the timeout V2G_SECC_Sequence_TimeoutCR to the value defined in Table 76. It shall then reset and start the V2G_SECC_Sequence_TimerCR, and it shall start to continuously monitor the V2G_SECC_Sequence_TimerCR for reaching or exceeding V2G_SECC_Sequence_TimeoutCR.

NOTE 1 In this document, sending a response message is described by A-DATA.response.

[V2G-DC-958] At any point in time in the V2G communication session while the SECC monitors the V2G_SECC_Sequence_TimerCR, if V2G_SECC_Sequence_TimerCR is equal or larger than V2G_SECC_Sequence_TimeoutCR, the SECC shall perform a normal shutdown according to [V2G-DC-650].

[V2G-DC-364] When the SECC sends a response message, it shall set the timeout V2G_SECC_Sequence_Timeout to the value defined in Table 76 for the respective message. It shall then reset and start the V2G_SECC_Sequence_Timer, and it shall start to continuously monitor the V2G_SECC_Sequence_Timer for reaching or exceeding V2G_SECC_Sequence_Timeout.

NOTE 2 In this document, sending a response message is described by A-DATA.response.

[V2G-DC-985] At any point in time in the V2G communication session while the SECC monitors the V2G_SECC_Sequence_Timer, if V2G_SECC_Sequence_Timer is equal or larger than V2G_SECC_Sequence_Timeout, the SECC shall proceed as if the SECC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-663].

[V2G-DC-367] When the SECC receives a request message while V2G_SECC_Sequence_Timer is smaller than V2G_SECC_Sequence_Timeout, it shall stop the V2G_SECC_Sequence_Timer and end monitoring it.

NOTE 3 In this document, receiving a request message is described by A-DATA.indication.

[V2G-DC-865] If the SECC sends a V2G response message with parameter EVSEProcessing equal to ’Ongoing’ for the first time in a response message it shall start the timer V2G_SECC_Ongoing_Timer.

[V2G-DC-866] If [V2G-DC-865] applies, the SECC shall try to send ResponseCode equal to "FAILED" when V2G_SECC_Ongoing_Timer is equal or larger than V2G_SECC_Ongoing_Performance_Time and no parameter EVSEProcessing equal to ’Finished’ has been sent. The SECC shall stop the V2G communication session.

9.6.5 V2G communication session timing definitions

Table 77 shows timing parameters applicable to several phases of the V2G communication session as defined in this document. Table 78 defines the values for the related performance times and the timeouts.

9.6.5.1 EVCC timing and error handling for V2G communication session setup timeout

[V2G-DC-369] When CP State B with 5% duty cycle is indicated at the beginning of a charging session, the EVCC shall set the timeout V2G_EVCC_CommunicationSetup_Timeout to the value defined in Table 78. It shall then reset and start the V2G_EVCC_CommunicationSetup_Timer, and it shall start to continuously monitor the V2G_EVCC_CommunicationSetup_Timer for reaching or exceeding V2G_EVCC_CommunicationSetup_Timeout.

[V2G-DC-976] At any point in time in the V2G communication session while the EVCC monitors the V2G_EVCC_CommuncationSetup_Timer, if V2G_EVCC_CommuncationSetup_Timer is equal or larger than V2G_EVCC_CommunicationSetup_Timeout, the EVCC shall proceed as if the EVCC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-673].

[V2G-DC-372] When waiting for a SessionSetupRes message according to the message sequence specified in 9.7.4.1, if the EVCC receives a SessionSetupRes message, it shall stop the V2G_EVCC_CommunicationSetup_Timer and end monitoring it.

NOTE In this document, receiving the SessionSetupRes message is described by A-DATA.confirmation(SessionSetupRes).

9.6.5.2 EVCC timing and error handling for ready to charge timeout

[V2G-DC-373] When CP state B with 5 % duty cycle is indicated at the beginning of a charging session, the EVCC shall set the timeout V2G_EVCC_ReadyToCharge_Timeout to the value defined in Table 78. It shall then reset and start the V2G_EVCC_ReadyToCharge_Timer, and it shall start to continuously monitor the V2G_EVCC_ReadyToCharge_Timer for reaching or exceeding V2G_EVCC_ReadyToCharge_Timeout.

[V2G-DC-977] At any point in time in the V2G communication session while the EVCC monitors the V2G_EVCC_ReadyToCharge_Timer, if V2G_EVCC_ReadyToCharge_Timer is equal or larger than V2G_EVCC_ReadyToCharge_Timeout, the EVCC shall proceed as if the EVCC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-673].

[V2G-DC-376] When waiting for a PowerDeliveryRes message according to the message sequence specified in chapter 9.7.4.1, if the EVCC receives a PowerDeliveryRes message, it shall stop the V2G_EVCC_ReadyToCharge_Timer and end monitoring it.

NOTE In this document, receiving the PowerDeliveryRes message is described by A-DATA.confirmation(PowerDeliveryRes).

9.6.5.3 EVCC timing and error handling for cable check timeout

[V2G-DC-377] When the EVCC sends the CableCheckReq message for the first time in a communication session, it shall set the timeout V2G_EVCC_CableCheck_Timeout to the value defined in Table 78. It shall then reset and start the V2G_EVCC_CableCheck_Timer, and it shall start to continuously monitor the V2G_EVCC_CableCheck_Timer for reaching or exceeding V2G_EVCC_CableCheck_Timeout.

NOTE 1 In this document, sending a CableCheckReq message is described by A-DATA.request(CableCheckReq).

[V2G-DC-978] At any point in time in the V2G communication session while the EVCC monitors the V2G_EVCC_CableCheck_Timer, if V2G_EVCC_CableCheck_Timer is equal or larger than V2G_EVCC_CableCheck_Timeout, the EVCC shall proceed as if the EVCC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-673].

[V2G-DC-381] When waiting for a CableCheckRes message according to the message sequence specified in chapter 9.7.4.1, if the EVCC receives a CableCheckRes message with parameter ResponseCode equal to “OK” and parameter EVSEProcessing equal to “Finished”, the EVCC shall stop the V2G_EVCC_CableCheck_Timer and end monitoring it.

NOTE 2 In this document, receiving a CableCheckRes message is described by A-DATA.confirmation(CableCheckRes).

9.6.5.4 EVCC timing and error handling for pre-charging timeout

[V2G-DC-382] When the EVCC sends the PreChargeReq message for the first time in a communication session, it shall set the timeout V2G_EVCC_PreCharge_Timeout to the value defined in Table 78. It shall then reset and start the V2G_EVCC_PreCharge_Timer, and it shall start to continuously monitor the V2G_EVCC_PreCharge_Timer for reaching or exceeding V2G_EVCC_PreCharge_Timeout.

NOTE 1 In this document, sending a PreChargeReq message is described by A-DATA.request(PreChargeReq).

[V2G-DC-979] At any point in time in the V2G communication session while the EVCC monitors the V2G_EVCC_PreCharge_Timer, if V2G_EVCC_PreCharge_Timer is equal or larger than V2G_EVCC_PreCharge_Timeout, the EVCC shall proceed as if the EVCC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-673].

[V2G-DC-386] When waiting for a PreChargeRes message, if the EVCC receives a PreChargeRes message and the EV determines that the EVSE output voltage, as measured inside the EV, has sufficiently been adjusted to the EV RESS voltage, the EVCC shall stop the V2G_EVCC_PreCharge_Timer and end monitoring it.

NOTE 2 In this document, receiving a PreChargeRes message is described by A-DATA.confirmation(PreChargeRes).

9.6.5.5 SECC timing and error handling for power delivery timeout

[V2G-DC-969] When the SECC receives a PreChargeReq message for the first time in a communication session, it shall set the timeout V2G_SECC_PowerDelivery_Timeout to the value defined in Table 78. It shall then reset and start the V2G_SECC_PowerDelivery_Timer and it shall start to continuously monitor the V2G_SECC_PowerDelivery_Timer for reaching or exceeding V2G_SECC_PowerDelivery_Timeout.

[V2G-DC-989] At any point in time in the V2G communication session while the SECC monitors the V2G_SECC_PowerDelivery_Timer, if V2G_SECC_PowerDelivery_Timer is equal or larger than V2G_SECC_PowerDelivery_Timeout, the SECC may proceed as if the SECC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-663].

NOTE According to IEC 61851-23, when the SECC carries out an “EVSE emergency shutdown”, it also performs a fast reduction of the output current. For details, refer to IEC 61851-23.

[V2G-DC-971] When waiting for another PreChargeReq message, a PowerDeliveryReq message, or a SessionStopReq message, if the SECC receives a PowerDeliveryReq message, the SECC shall stop the V2G_SECC_PowerDelivery_Timer and end monitoring it.

The EVSE may shutdown when first PreChargeReq message to PowerDeliveryReq message takes too long (see IEC61851-23 Ed2).

9.6.5.6 Summary of V2G communication session timing

Table 77 — EVCC and SECC V2G communication session timing parameters

Parameter name	**Definition**	**Implementation**	
		EVCC	SECC
V2G_EVCC_CommunicationSetup_Timer	Communication setup timer in the EVCC	X	
V2G_SECC_CommunicationSetup_Timer	Communication setup timer in the SECC		X
V2G_EVCC_ReadyToCharge_Timer	Ready-to-charge timer in the EVCC	X	
V2G_SECC_ReadyToCharge_Timer	Ready-to-charge timer in the SECC		X
V2G_EVCC_CableCheck_Timer	Cable check timer in the EVCC	X	
V2G_SECC_CableCheck_Timer	Cable check timer in the SECC		X
V2G_EVCC_PreCharge_Timer	PreCharge timer in the EVCC	X	
V2G_SECC_PreCharge_Timer	PreCharge timer in the SECC		X
V2G_SECC_PowerDelivery_Timer	PowerDelivery timer in the SECC		X
V2G_SECC_CPState_Detection_Timer	CPState detection timer in the SECC		X
V2G_EVCC_CommunicationSetup_Timeout	Timeout for the communication setup timer in the EVCC as defined in Table 78.	X	
V2G_SECC_CommunicationSetup_ Performance_Time	Performance time for the communication setup timer in the SECC as defined in Table 78.		X

Table 77 (*continued*)

Parameter name	Definition	Implementation	
		EVCC	SECC
V2G_EVCC_ReadyToCharge_Timeout	Timeout for the ready-to-charge timer in the EVCC as defined in Table 78.	X	
V2G_SECC_ReadyToCharge_Performance_Time	Performance time for the ready-to-charge timer in the SECC as defined in Table 78.		X
V2G_EVCC_CableCheck_Timeout	Timeout for the CableCheck timer in the EVCC as defined in Table 78.	X	
V2G_SECC_CableCheck_Performance_Time	Performance time for the CableCheck timer in the SECC as defined in Table 78.		X
V2G_EVCC_PreCharge_Timeout	Timeout for the PreCharge timer in the EVCC as defined in Table 78.	X	
V2G_SECC_PreCharge_Performance_Time	Performance time for the PreCharge timer in the SECC as defined in Table 78.		X
V2G_SECC_PowerDelivery_Timeout	Timeout for the PowerDelivery timer in the SECC as defined in Table 78.		X

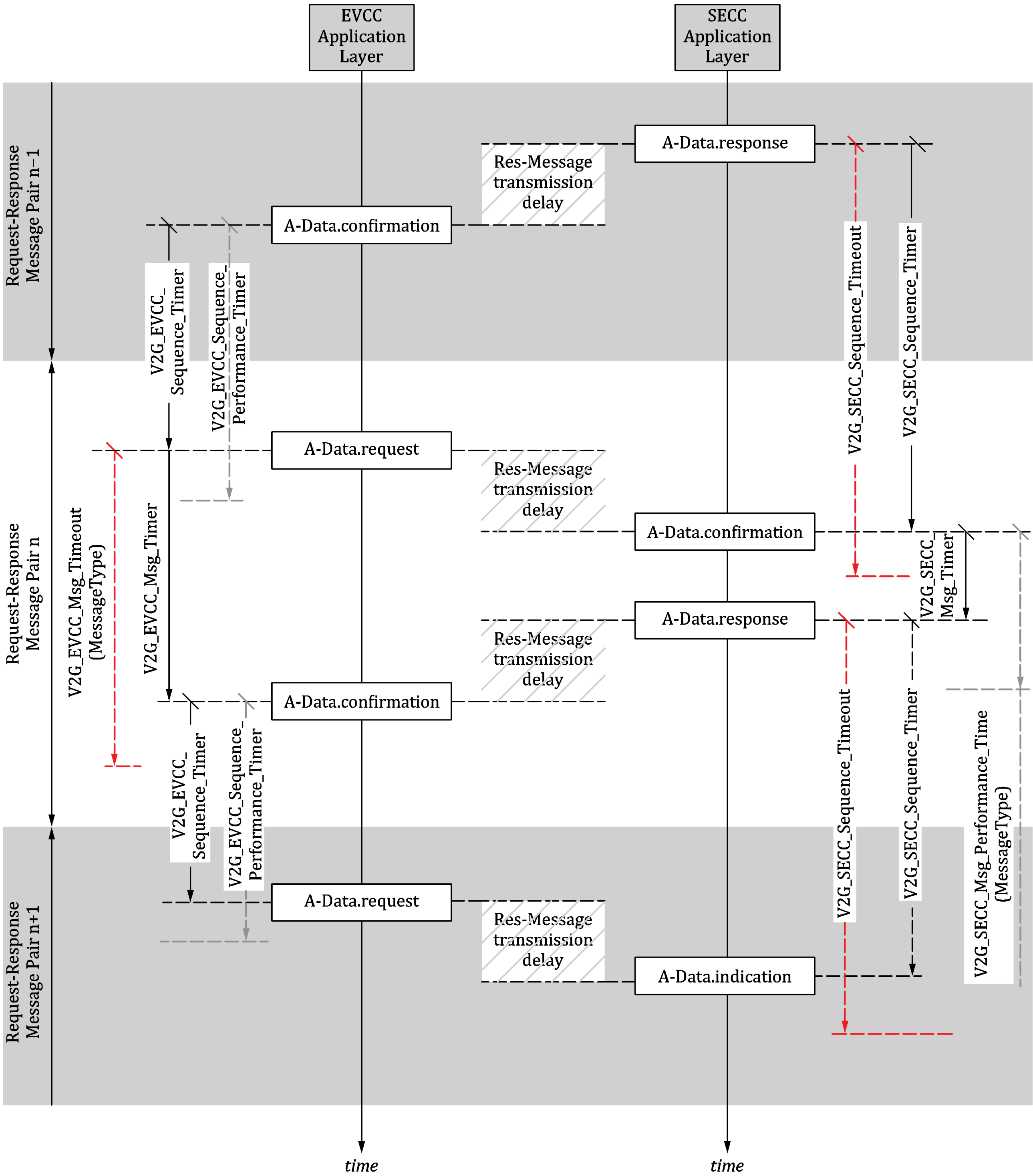
[V2G-DC-644] The EVCC shall implement an EVCC specific timeout supervision according to Table 77 that satisfies the values defined in Table 78.

The SECC should exhibit a temporal behavior with respect to the so-called “performance times” that satisfies the values defined in Table 78.

Table 78 — EVCC and SECC V2G communication session timing parameter values

Parameter name	Value [s]	Implementation	
		EVCC	SECC
V2G_SECC_ReadyToCharge_Performance_Time	148		x
V2G_EVCC_ReadyToCharge_Timeout	150	x	
V2G_SECC_CommunicationSetup_Performance_Time	18		x
V2G_EVCC_CommunicationSetup_Timeout	20	x	
V2G_SECC_CableCheck_Performance_Time	38		x
V2G_EVCC_CableCheck_Timeout	40	x	
V2G_SECC_PreCharge_Performance_Time	8		x
V2G_EVCC_PreCharge_Timeout	10	x	
V2G_SECC_PowerDelivery_Timeout	20		x
V2G_SECC_CPState_Detection_Timeout	1,5		x
V2G_SECC_CPOscillator_Retain_Time	1,5		x

— Figure 77 illustrates how some of the timing parameters defined in Table 77 are applied.



Key

Threshold: Timeout (error critera)

Threshold: Performance time (performance criteria)

Timer

Figure 77 — V2G communication session timing

9.7 Message sequences

9.7.1 Protocol flow stages and associated messages

After connecting EV and EVSE, a network connection is established. Then, initialization of this V2G communication session shall be done. After getting all services offered by the EVSE, charging the battery can start. At the end of the charging process, power is switched off. The following list describes the sequence of messages which is used to control the flow:

Initialization of V2G communication session: Check protocol compatibility, establish V2G communication session, exchange client-/server IDs. Messages for this activity:

— supportedAppProtocolRequest/response;

— Session setup request/response.

Service Discovery: Discover the services offered by the EVSE, agreement on billing parameters. Messages for this activity:

— Service discovery request/response;

— Service and payment selection request/response;

— Contract authentication request/response.

Charge vehicle: Charging the EV is one possible service offered by the EVSE. It is divided into three phases:

Set up charging process: Agreement on technical charging parameters, agreement on billing parameters, lock charge cord, start charging process. Messages for this activity:

— Charge parameter discovery request/response;

— Cable check request/response;

— PreCharge request/response;

— Power delivery request/response.

Charging process: energy transfer

— Current demand request/response.

Finalise charging process: Stop charging process, check welding, unlock charge cord:

— Power delivery request/response;

— Welding detection request/response;

— Stop session request/response.

After finalising the charging process, the charge cord can be disconnected from the EV.

9.7.2 Basic definitions for error handling

The basic error handling for a request-response message pair is based on the response code included in the response message of the SECC. Depending on the value in the response code, the EVCC decides if it can proceed with the standard request-response message sequence or whether it handles an error.

In this document, the response code as defined in Annex A.6 is interpreted by the EVCC as follows:

OK: Any value starting with "OK" or "OK_" indicates a positive response. Detailed information may be provided by OK_<additional info>. This information may be used to differentiate the reaction to the positive response.

FAILED: Any value starting with "FAILED" or "FAILED_" indicates a negative response. Detailed information may be provided by FAILED_<additional info>. This information may be used to differentiate the reaction to the negative response.

9.7.3 Response code usage by the SECC

Each response message the EVSE sends to the EV contains a ResponseCode that informs the EV whether the previous request sent by the EV has been processed successfully or has resulted in an error. Successful processing is indicated by the ResponseCode value "OK" or "OK_<additional info>". Errors are indicated either by the general error value "FAILED", or by a more specific error code "FAILED_<additional info>" where "<additional info>" contains a short description of the cause of the error.

[V2G-DC-388] A response message shall contain the ResponseCode "OK" if the processing of the request message was successful. If at some point in this document, a specific positive ResponseCode is defined for a specific situation, that ResponseCode shall be used.

[V2G-DC-389] A response message shall contain the ResponseCode "FAILED" if the processing of the request message is not successful and no specific ResponseCode is defined for the concrete error case.

[V2G-DC-390] The response message shall contain the ResponseCode "FAILED_SequenceError" if the SECC has received an unexpected request message.

[V2G-DC-391] The response message shall contain the ResponseCode "FAILED_UnknownSession" if the SessionID in a request message does not match the SessionID provided by the SECC in the SessionSetupRes message.

[V2G-DC-393] The SessionSetupRes message shall contain the specific ResponseCode "OK_NewSessionEstablished" if processing of the SessionSetupReq message was successful and a different SessionID is contained in the response message than the SessionID in the request message.

[V2G-DC-395] The ServicePaymentSelectionRes message shall contain the ResponseCode "FAILED_PaymentSelectionInvalid" if the SelectedPaymentOption contained in the ServicePaymentSelectionReq message was not part of the offered PaymentOptions in the ServiceDiscoveryRes message.

[V2G-DC-396] The ServicePaymentSelectionRes message shall contain the ResponseCode "FAILED_ServiceSelectionInvalid" if the SelectedServiceList contained in the ServicePaymentSelectionReq message contains a ServiceID which was not contained in the offered ServiceList in the ServiceDiscoveryRes message.

[V2G-DC-397] The ChargeParameterDiscoveryRes message shall contain the ResponseCode "FAILED_WrongEnergyTransferType" if the content of element EVRequestedEnergyTransferType" in the ChargeParameterDiscoveryReq message is not valid or does not match the content of the element EVChargeParameter.

[V2G-DC-398] The ChargeParameterDiscoveryRes message shall contain the ResponseCode "FAILED_WrongChargeParameter" if the content of element EVChargeParameter in the ChargeParameterDiscoveryReq message is not valid, e.g. a wrong parameter set is provided, one or multiple parameters cannot be interpreted, etc.

[V2G-DC-399] The PowerDeliveryRes message shall contain the ResponseCode “FAILED_ChargingProfileInvalid” if the content of element ChargingProfile in the PowerDeliveryReq message violates a power limitation provided in the ChargeParameterDiscoveryRes message.

[V2G-DC-400] The PowerDeliveryRes message shall contain the ResponseCode “FAILED_TariffSelectionInvalid” if the content of element ChargingProfile in the PowerDeliveryReq message contains a SAScheduleTupleID which was not contained in the SASchedules attribute provided in the ChargeParameterDiscoveryRes message.

[V2G-DC-401] The PowerDeliveryRes message shall contain the ResponseCode “FAILED_PowerDeliveryNotApplied” if the EVSE is not able to deliver energy.

NOTE Response codes that are not defined in this subclause can be used implementation specific.

Table 79 shows an overview of the response codes and the messages as defined before.

Table 79 — Overview on the use of ResponseCodes

Response Code	Response Messages																
	SessionSetupRes	ServicerDiscoveryRes	ServiceDetailRes	ServiceandPaymentSelectionRes	PaymentDetailsRes	ContractAuthenticationRes	ChargeParameterDiscoveryRes	PowerDeliveryRes	ChargingStatusRes	MetreringReceiptRes	CertificateupdateRes	CertificateinstallationRes	CableCheckRes	PreChargingRes	CurrentDemandRes	WeldingDetectionRes	SessionStopRes
OK	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
OK_CertificateExpiresSoon																	
OK_NewSessionEstablished	X																
OK_OldSessionJoined	X																
FAILED	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
FAILED_SequenceError	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
FAILED_SignatureError	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	x
FAILED_UnknownSession		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
FAILED_ServiceIDInvalid			X														
FAILED_PaymentSelectionInvalid				X													
FAILED_ServiceSelectionInvalid				X													
FAILED_CertificateExpired					X						X	X					
FAILED_NoCertificateAvailable												X					
FAILED_CertChainError											X						
FAILED_ContractCanceled											X						
FAILED_ChallengeInvalid						X											
FAILED_WrongEnergyTransfer-Type							X										
FAILED_WrongChargeParameter							X										
FAILED_ChargingProfileInvalid								X									
FAILED_TariffSelectionInvalid								X									

Table 79 (*continued*)

Response Code	Response Messages																
FAILED_EVSEPresentVoltageTo-Low																	
FAILED_PowerDeliveryNotApp-lied								X									
FAILED_MeteringSignatureNotVa-lid										X							
X: response message contains response code.																	

9.7.4 Request-response message sequence requirements

9.7.4.1 EVCC message sequence requirements

This chapter describes the EVCC Communication states for the DC V2G messaging.

9.7.4.1.1 General EVCC sequence requirement

[V2G-DC-645] Each request messages of the message sequence as specified in 9.7.4.1.4 shall only be sent by the EVCC at that/those position(s) in the message sequence that is/are explicitly specified in requirements within this subclause, and under that/those condition(s) explicitly specified in the respective requirement(s), if any condition(s) is/are specified.

Exception: When, according to a requirement in this specification, the EVCC terminates the V2G communication session (e.g. due to an error condition), the EVCC may still send one or more request messages, e.g. trying to properly terminate the bidirectional communication with the EVSE, but the EVCC shall not rely on any response messages from the EVSE.

[V2G-DC-646] A change of the CP state shall only be caused by the EVCC at that/those position(s) in the message sequence that is/are explicitly specified in requirements within this chapter, and under that/those condition(s) explicitly specified in the respective requirement(s), if any condition(s) is/are specified.

Exception: CP State changes that are specified in standards referenced by this document, e.g. ISO/IEC 17409, are also permissible.

[V2G-DC-647] During the charging session, the EVCC shall continuously monitor the CP duty cycle for any change.

9.7.4.1.2 EVCC requirements for normal shutdown

[V2G-DC-648] If the EVCC wishes to stop the communication session for a non-critical reason (e.g. user interaction) after having sent a SessionSetupReq message and before having sent a PowerDeliveryReq message with ReadyToChargeState equal to "TRUE", the EVCC should not immediately change the CP state, unless this is required by another requirement in this document, but the EVCC shall send a SessionStopReq message as the next request message and then wait for a SessionStopRes message. It shall then proceed according to [V2G-DC-413], [V2G-DC-750], [V2G-DC-657], and [V2G-DC-658].

[V2G-DC-673] If the EVCC wishes to stop the communication session for a non-critical reason (e.g. user interaction) after having sent a CurrentDemandReq message and before having sent a PowerDeliveryReq message with ReadyToChargeState equal to "FALSE", the EVCC shall not immediately change the CP State, unless this is required by another requirement in this document, but the EVCC shall send a PowerDeliveryReq message with ReadyToCharge-

State equal to “FALSE” as the next request message and then wait for a PowerDeliveryRes message. It shall then proceed according to [V2G-DC-747], [V2G-DC- 502], [V2G-DC-427], [V2G-DC-428], and so on.

NOTE 1 If the EVCC wishes to stop the communication session for a non-critical reason (e.g. user interaction) after having sent a PowerDeliveryReq message with ReadyToChargeState equal to “TRUE” and before having sent a CurrentDemandReq message, the EVCC, in accordance with [V2G-DC-424], sends a CurrentDemandReq message, e. g. with EVTargetCurrent equal to “0 A” (zero ampere), and wait for a CurrentDemandRes message before continuing according to [V2G-DC-673].

NOTE 2 It is recommended for the EVSE to reduce current before initiating shutdown with EVSE_Shutdown. (e.g. stop pressed at EVSE).

[V2G-DC-649] Before having sent a ChargeParameterDiscoveryReq message, if the EVCC receives a response message that it expects in response to the latest request message sent, and this response message contains ResponseCode equal to “FAILED”, and the EVCC detects that the CP oscillator is not turned off, the EVCC shall proceed as if the EVCC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648].

NOTE 3 Refer to 9.7.2 concerning values of ResponseCode starting with “FAILED” or “FAILED_”.

[V2G-DC-650] If the EVCC receives a response message that it expects in response to the latest request message sent, and this response message contains EVSEStatusCode equal to “EVSE_Shutdown“, the EVCC shall proceed as if the EVCC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] and [V2G-DC-673].

If the parameter EVSEStatusCode equal to “EVSE_Shutdown“ is contained in a PowerDeliveryRes message sent in response to a PowerDeliveryReq message with ReadyToChargeState equal to “TRUE”, the EVCC, in accordance with [V2G-DC-424], shall send a CurrentDemandReq message, e. g. with EVTargetCurrent equal to “0 A” (zero ampere), and wait for a CurrentDemandRes message before continuing according to [V2G-DC-673].

It is recommended for the EV to reduce current before initiating shutdown with PowerDeliveryReq (ReadyToChargeState=false), (e.g. stop pressed at EV).

9.7.4.1.3 EVCC requirements for normal shutdown

[V2G-DC-980] If the EVCC receives a response message that it expects in response to the latest request message sent, and if the EVCC cannot process this response message, e.g. due to errors in the message parameters or due to impeding conditions in the EV, the EVCC shall proceed as if the EVCC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-673].

[V2G-DC-981] Before having sent a ChargeParameterDiscoveryReq message, if the EVCC receives a response message that it expects in response to the latest request message sent, and this response message contains ResponseCode equal to “FAILED”, and the EVCC detects that the CP oscillator is turned off, the EVCC shall proceed as if the EVCC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-673].

NOTE 1 Refer to 9.7.2 concerning values of ResponseCode starting with “FAILED” or “FAILED_”.

[V2G-DC-982] After having sent a ChargeParameterDiscoveryReq message, if the EVCC receives a response message that it expects in response to the latest request message sent, and this response message contains ResponseCode equal to “FAILED”, the EVCC shall proceed as if the EVCC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-673].

NOTE 2 Refer to 9.7.2 concerning values of ResponseCode starting with "FAILED" or "FAILED_".

NOTE 3 According to IEC 61851-23, if the SECC detects an unexpected CP State B, the SECC carries out, without any delay, an "EVSE-initiated emergency shutdown", which includes turning off the CP oscillator, if it is turned on, and a fast reduction of the output current. For details, refer to IEC 61851-23.

[V2G-DC-983] At any point in time in the V2G communication session, if the EVCC receives a response message that it does not expect in response to the latest request message sent, the EVCC shall proceed as if the EVCC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-673].

[V2G-DC-990] If the EVCC receives a response message that it expects in response to the latest request message sent, and this response message contains ResponseCode equal to "FAILED_SequenceError", the EVCC shall proceed as if the EVCC itself wished to stop the Communication Session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-673].

[V2G-DC-655] At any point in time in the V2G communication session, if the EVCC detects CP State E or CP State F, the EVCC shall close the TCP connection according to

[V2G-DC-984] At any time in the communication session after having received a ChargeParameterDiscoveryRes message, if the EVCC detects that the CP duty cycle is not 5 %, the EVCC shall proceed as if the EVCC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-673].

[V2G-DC-871] At any Point after the supportedAppProtocolReq/Res sequence has ended and DIN/TS 70121 has been selected as the charging protocol, the EV measures a PWM which is not in range of 3 % <= duty cycle <= 7 %, the EVCC shall:

1. change to CP State B, if the EVCC is not in State B,
2. terminate the V2G communication session, and
3. close the TCP connection according to
4. [V2G-DC-107].

NOTE 4 For further details (e.g. timings) see IEC61851-23.

NOTE 5 Until supportedAppProtocol, the EV does not know if the EVSE is AC or DC station and if it supports DIN/TS 70121 and/or ISO 15118. For backward compatibility, the EV is capable to handle these situations.

If the EVSE signals 8 % - 97 % it is an AC station, but it may still support ISO 15118.

9.7.4.1.4 Normal EVCC message sequence

[V2G-DC-738] Before initiating the V2G communication session, the EVCC shall be in CP state B.

[V2G-DC-403] After the TCP connection is established, in order to initiate the V2G communication session according to [V2G-DC-106], the EVCC shall send a supportedAppProtocolReq message as the first application layer message of a V2G communication session and shall then wait for a supportedAppProtocolRes message.

[V2G-DC-739] When waiting for a supportedAppProtocolRes message, if the EVCC receives a supportedAppProtocolRes message, the EVCC shall process this response message.

[V2G-DC-405] After the EVCC has successfully processed a received supportedAppProtocolRes message with ResponseCode equal to “OK_SuccessfulNegotiation” or “OK_SuccessfulNegotiationWithMinorDeviation”, the EVCC shall send a SessionSetupReq message and shall then wait for a SessionSetupRes message.

[V2G-DC-740] When waiting for a SessionSetupRes message, if the EVCC receives a SessionSetupRes message, the EVCC shall process this response message.

[V2G-DC-407] After the EVCC has successfully processed a received SessionSetupRes message with ResponseCode equal to “OK”, the EVCC shall send a ServiceDiscoveryReq message and shall then wait for a ServiceDiscoveryRes message.

[V2G-DC-741] When waiting for a ServiceDiscoveryRes message, if the EVCC receives a ServiceDiscoveryRes message, the EVCC shall process this response message.

[V2G-DC-409] After the EVCC has successfully processed a received ServiceDiscoveryRes message with ResponseCode equal to “OK”, the EVCC shall send a ServicePaymentSelectionReq message and shall then wait for a ServicePaymentSelectionRes message.

NOTE 1 There are no service details needed within the scope of this document.

[V2G-DC-742] When waiting for a ServicePaymentSelectionRes message, if the EVCC receives a ServicePaymentSelectionRes message, the EVCC shall process this response message.

[V2G-DC-414] After the EVCC has successfully processed a received ServicePaymentSelectionRes message with ResponseCode equal to “OK”, the EVCC shall send a ContractAuthenticationReq message and shall then wait for a ContractAuthenticationRes message.

[V2G-DC-743] When waiting for a ContractAuthenticationRes message, if the EVCC receives a ContractAuthenticationRes message, the EVCC shall process this response message.

[V2G-DC-492] After the EVCC has successfully processed a received ContractAuthenticationRes message with ResponseCode equal to “OK” and EVSEProcessing equal to “Ongoing”, the EVCC shall resend the identical ContractAuthenticationReq message as before and shall then wait for a ContractAuthenticationRes message.

To avoid a timeout by V2G_EVCC_Msg_Timer reaching or exceeding V2G_EVCC_Msg_Timeout, the SECC may use EVSEProcessing equal to “Ongoing” in ContractAuthenticationRes to inform the EVCC that the SECC is still processing the requested contract authentication.

[V2G-DC-490] After the EVCC has successfully processed a received ContractAuthenticationRes message with ResponseCode equal to “OK” and EVSEProcessing equal to “Finished”, the EVCC shall send a ChargeParameterDiscoveryReq message and shall then wait for a ChargeParameterDiscoveryRes message.

NOTE 2 The EVSE can perform authorization before providing energy. Some EVSE manufactures are confused how to use the ContractAuthorizationReq/Res in case of the following both authorization situations:

— Authorization before plugging;

— Authorization after plugging;

— Authorization before plugging:

  — If authorization is done before plugging, at least one ContractAuthorizationReq/Res shall be sent with ContractAuthorizationRes and EVSEProcessing = Finished

— Authorization after plugging:

  — If authorization is done after plugging and not until the ContractAuthorizationReq/Res, the ContractAuthorizationRes message shall contain EVSEProcessing = Ongoing until authorization is done. If the authorization is done, the ContractAuthorizationRes message shall contain EVSEProcessing = Finished.

[V2G-DC-744] When waiting for a ChargeParameterDiscoveryRes message, if the EVCC receives a ChargeParameterDiscoveryRes message, the EVCC shall process this response message.

[V2G-DC-493] After the EVCC has successfully processed a received ChargeParameterDiscoveryRes message with ResponseCode equal to “OK” and EVSEProcessing equal to “Ongoing”, the EVCC shall ignore the values of SAScheduleList and DC_EVSEChargeParameter contained in this ChargeParameterDiscoveryRes message, and shall send another ChargeParameterDiscoveryReq message and shall then wait for a ChargeParameterDiscoveryRes message. This following ChargeParameterDiscoveryReq message, if any, may contain different values for the contained parameters than the preceding ChargeParameterDiscoveryReq message.

To avoid a timeout by V2G_EVCC_Msg_Timer reaching or exceeding V2G_EVCC_Msg_Timeout, the SECC may use EVSEProcessing equal to “Ongoing” in the ChargeParameterDiscoveryRes message to inform the EVCC that the SECC is still processing the requested charge parameter discovery.

[V2G-DC-881] If the EVSE has received a ChargeParameterDiscoveryReq and the timeout as defined [V2G-DC-357] has not been met, the EVSE shall send a ChargeParameterDiscoveryRes as defined in [V2G-DC-493]:

— EVSEStatusCode = EVSE_IsolationMonitoringActive [V2G-DC-499]

— EVSEIsolationStatus = Invalid

— EVSENotification = none

[V2G-DC-880] After receiving a ChargeParameterDiscoveryRes message with EVSEProcessing=Finished and before sending a CableCheckReq message, the EVCC shall change from CP state B to CP state C or D as defined in IEC 61851-1.

NOTE 3 No simplified control pilot is used for interoperability (no state C after plug-in).

[V2G-DC-415] After the EVCC has successfully processed a received ChargeParameterDiscoveryRes message with ResponseCode equal to “OK” and EVSEProcessing equal to “Finished”, the EVCC shall send a CableCheckReq message and shall then wait for a CableCheckRes message.

[V2G-DC-745] When waiting for a CableCheckRes message, if the EVCC receives a CableCheckRes message, the EVCC shall process this response message.

[V2G-DC-418] After the EVCC has successfully processed a received CableCheckRes message with ResponseCode equal to “OK” and EVSEProcessing equal to “Ongoing” while V2G_EVCC_CableCheck_Timer is smaller than V2G_EVCC_CableCheck_Timeout, the EVCC shall resend the identical CableCheckReq message as before and shall then wait for a CableCheckRes message.

To avoid a timeout by V2G_EVCC_Msg_Timer reaching or exceeding V2G_EVCC_Msg_Timeout, the SECC may use EVSEProcessing equal to “Ongoing” in the CableCheckRes message to inform the EVCC that the SECC is still processing the requested cable check.

[V2G-DC-417] After the EVCC has successfully processed a received CableCheckRes message with ResponseCode equal to "OK" and EVSEProcessing equal to "Finished", the EVCC shall send a PreChargeReq message and shall then wait for a PreChargeRes message.

[V2G-DC-892] In case [V2G-DC-889] applies, when EVSEProcessing is equal to "Finished" and ResponseCode is equal to "OK", the EVCC shall send a PreChargeReq.

[V2G-DC-893] In case [V2G-DC-889] applies, when EVSEProcessing is equal to 'Finished', the EVCC shall disregard all values of EVSEStatusCode except for EVSE_Shutdown and EVSE_EmergencyShutdown for sending the next PreChargeReq message.

[V2G-DC-894] In case [V2G-DC-889] applies when EVSEProcessing is equal to 'Finished', the EVCC shall disregard all values of EVSEIsolationStatus for sending the next PreChargeReq message.

[V2G-DC-895] In case [V2G-DC-886] applies, when EVSEProcessing is equal to "Finished" and ResponseCode is equal to "OK", the EVCC shall send a PreChargeReq.

[V2G-DC-896] In case [V2G-DC-888] applies, the EVCC shall disregard all values of EVSEStatusCode except for EVSE_Shutdown and EVSE_EmergencyShutdown for sending the next PreChargeReq message.

[V2G-DC-897] In case [V2G-DC-888] applies, the EVCC shall disregard all values of EVSEIsolationStatus for sending the next PreChargeReq message.

[V2G-DC-898] While EVSEProcessing is equal to Ongoing, the EVCC shall continue sending CableCheckReq until EVSEProcessing in CableCheckRes is Finished.

[V2G-DC-899] While EVSEProcessing is equal to Ongoing in CableCheckRes, the EVCC shall disregard all values of EVSEStatusCode except for EVSE_Shutdown and EVSE_EmergencyShutdown for sending the next CableCheckReq message.

[V2G-DC-900] While EVSEProcessing is equal to Ongoing in CableCheckRes, the EVCC shall disregard all values of EVSEIsolationStatus for sending the next CableCheckReq message.

[V2G-DC-901] In case [V2G-DC-891] applies, when EVSEProcessing is equal to "Finished" and ResponseCode is equal to "FAILED", the EVCC shall stop the charging session for a non-critical reason as defined in [V2G-DC-650], [V2G-DC-648] and [V2G-DC-673].

[V2G-DC-902] In case [V2G-DC-891] applies, if [V2G-DC-418] and [V2G-DC-650] apply at the same time, [V2G-DC-650] shall be processed, even if EVSEProcessing is equal to "Finished".

[V2G-DC-746] When waiting for a PreChargeRes message, if the EVCC receives a PreChargeRes message, the EVCC shall process this response message.

[V2G-DC-421] After the EVCC has successfully processed a received PreChargeRes message with ResponseCode equal to "OK", if the EV determines that the EVSE output voltage, as measured inside the EV, has sufficiently been adjusted to the EV RESS voltage, the EVCC shall send a PowerDeliveryReq message with ReadyToChargeState equal to "TRUE" and shall then wait for a PowerDeliveryRes message.

[V2G-DC-422] After the EVCC has successfully processed a received PreChargeRes message with ResponseCode equal to "OK", if the EV determines that the EVSE output voltage, as measured inside the EV, has not sufficiently been adjusted to the EV RESS voltage, the EVCC shall send another PreChargeReq message and shall then wait for a PreChargeRes message.

NOTE 4 See IEC 61851-23 for pre-charge details.

[V2G-DC-747] When waiting for a PowerDeliveryRes message, if the EVCC receives a PowerDeliveryRes message, the EVCC shall process this response message.

[V2G-DC-424] After the EVCC has successfully processed a received PowerDeliveryRes message with ResponseCode equal to “OK” as a response to a previous PowerDeliveryReq message with ReadyToChargeState equal to “TRUE”, the EVCC shall send a CurrentDemandReq message and shall then wait for a CurrentDemandRes message.

[V2G-DC-748] When waiting for a CurrentDemandRes message, if the EVCC receives a CurrentDemandRes message, the EVCC shall process this response message.

[V2G-DC-425] After the EVCC has successfully processed a received CurrentDemandRes message with ResponseCode equal to “OK”, if the charging process shall be continued, the EVCC shall send another CurrentDemandReq message and shall then wait for a CurrentDemandRes message.

[V2G-DC-915] The EVCC shall transmit CurrentDemandReq or PowerDeliveryReq (ReadytoChargeState = False) no later than 1 second from the time the last CurrentDemandRes was received.

The EVSE should not to stop if the timing above it not met to support legacy EVs.

[V2G-DC-916] After the EVCC has received a CurrentDemandRes message it shall wait at least 100 ms before sending the next CurrentDemandRequest message.

NOTE 5 Some EVSEs are not able to handle a huge load of incoming V2G request messages in short time. (e.g. CurrentDemandRes after CurrentDemandReq).

[V2G-DC-420] After the EVCC has successfully processed a received CurrentDemandRes message with ResponseCode equal to “OK”, if the charging process shall be stopped, the EVCC shall send a PowerDeliveryReq message with ReadyToChargeState equal to “FALSE” and shall then wait for a PowerDeliveryRes message.

[V2G-DC-502] After sending a PowerDeliveryReq message with ReadyToChargeState equal to “FALSE” and receiving the corresponding PowerDeliveryRes message, the EVCC shall change to CP State B as defined in IEC 61851-1 before sending the next request message.

[V2G-DC-427] After the EVCC has successfully processed a received PowerDeliveryRes message with ResponseCode equal to “OK” as a response to a previous PowerDeliveryReq message with ReadyToChargeState equal to “FALSE”, if the EV wishes to perform a Welding Detection, the EVCC shall send a WeldingDetectionReq message and shall then wait for a WeldingDetectionRes message.

[V2G-DC-428] After the EVCC has successfully processed a received PowerDeliveryRes message with ResponseCode equal to “OK” as a response to a previous PowerDeliveryReq message with ReadyToChargeState equal to “FALSE”, if the EV does not wish to perform a welding detection, the EVCC shall send a SessionStopReq message and shall then wait for a SessionStopRes message.

[V2G-DC-749] When waiting for a WeldingDetectionRes message, if the EVCC receives a WeldingDetectionRes message, the EVCC shall process this response message.

In WeldingDetection error case there’s also a SessionStop message sent. No other message possible.

[V2G-DC-430] After the EVCC has successfully processed a received WeldingDetectionRes message with ResponseCode equal to “OK”, if the Welding Detection function has finished on EV side, the EVCC shall send a SessionStopReq message and shall then wait for a SessionStopRes message.

[V2G-DC-431] After the EVCC has successfully processed a received WeldingDetectionRes message with ResponseCode equal to "OK", if the Welding Detection function has not yet finished on EV side, the EVCC shall send another WeldingDetectionReq message and shall then wait for a WeldingDetectionRes message.

NOTE 6 See IEC 61851-23 for welding detection details.

[V2G-DC-413] When waiting for a SessionStopRes message, if the EVCC receives a SessionStopRes message, the EVCC shall process this response message.

[V2G-DC-750] After the EVCC has successfully processed a received SessionStopRes message, the EVCC shall change to CP state B as defined in IEC 61851-1 if the EVCC is not in CP state B.

Changing to CP state B after processing a received SessionStopRes message is necessary in some cases when the EVCC wishes to stop the communication session for a non-critical reason according to [V2G-DC-648].

[V2G-DC-657] After the EVCC has successfully processed a received SessionStopRes message with ResponseCode equal to "OK", the V2G communication session is properly stopped and the EVCC shall close the TCP connection according to

[V2G-DC-658] After the EVCC has successfully processed a received SessionStopRes message with ResponseCode not equal to "OK", the V2G communication session is stopped with an error and the EVCC shall close the TCP connection according to SECC message sequence requirements.

This subclause describes the SECC communication states for the DC V2G messaging.

9.7.4.1.5 General SECC sequence requirement

[V2G-DC-659] Each response messages of the message sequence as specified in 9.7.4.1.8 shall only be sent by the SECC at that/those position(s) in the message sequence that is/are explicitly specified in requirements within this chapter, and under that/those condition(s) explicitly specified in the respective requirement(s), if any condition(s) is/are specified.

Exception: When, according to a requirement in this specification, the SECC shall terminate the V2G communication session (e.g. due to an error condition), the SECC may still send one or more response messages, e.g. trying to properly terminate the bidirectional communication with the EV, but the SECC shall not rely on any request messages from the EV.

[V2G-DC-660] A change of the CP duty cycle shall only be caused by the SECC at that/those position(s) in the message sequence that is/are explicitly specified in requirements within this subclause, and under that/those condition(s) explicitly specified in the respective requirement(s), if any condition(s) is/are specified.

Exception: CP state changes that are specified in standards referenced by this document, e.g. IEC 61851-23, are also permissible

[V2G-DC-661] During the charging session, the SECC shall continuously monitor the CP state for any change between CP state A, B, C, and D.

9.7.4.1.6 SECC requirements for normal shutdown

[V2G-DC-662] If the SECC wishes to stop the communication session for a non-critical reason (e.g. user interaction) after having received a supportedAppProtocolReq message and before having sent a ContractAuthenticationRes message, the SECC shall send the next response message with parameter ResponseCode equal to "FAILED", but the SECC shall not turn off the CP oscillator unless this is required by another requirement in this document.

[V2G-DC-663] If the SECC wishes to stop the Communication Session for a non-critical reason (e.g. user interaction) after having sent a ContractAuthenticationRes message, the SECC shall send the next response message with parameter EVSEStatusCode equal to "EVSE_Shutdown", even if another requirement in this document specifies that a different value for EVSEStatusCode shall be used, but the SECC shall not turn off the CP oscillator unless this is required by another requirement in this document.

NOTE After the SECC informs the EVCC according to [V2G-DC-662] or [V2G-DC-663] that the SECC wishes to stop the communication session, the further steps are controlled by the EVCC according to [V2G-DC-649] and [V2G-DC-650].

If the SECC uses the parameter EVSEStatusCode equal to "EVSE_Shutdown" in a PowerDeliveryRes message sent in response to a PowerDeliveryReq message with ReadyToChargeState equal to "TRUE", the EVCC, in accordance with [V2G-DC-424], shall send a CurrentDemandReq message and wait for a CurrentDemandRes message before continuing according to [V2G-DC-673].

[V2G-DC-664] During a V2G communication session, after having sent a SessionSetupRes message and before having received a PowerDeliveryReq message with ReadyToChargeState equal to "TRUE", if the SECC receives a SessionStopReq message, it shall proceed according to [V2G-DC-450], [V2G-DC-451], and[V2G-DC-968].

9.7.4.1.7 SECC requirements for normal shutdown

[V2G-DC-986] If the SECC receives a request message that it expects according to the message sequence specified in this chapter, and if the SECC cannot process this request message, e. g. due to errors in the message parameters or due to impeding conditions in the EVSE, the SECC shall proceed as if the SECC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-663].

[V2G-DC-987] At any point in time in the V2G communication session, if the SECC receives a request message that it does not expect according to the message sequence specified in this subclause, the SECC shall proceed as if the SECC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-663].

[V2G-DC-667] At any point in time in the V2G communication session, if the SECC detects CP state A, the SECC shall close the TCP connection according to [V2G-DC-116].

NOTE 1 According to IEC 61851-23, if the SECC detects CP State A or an unexpected CP State B, it carries out, without any delay, an "EVSE-initiated emergency shutdown" which includes turning off the CP oscillator, if it is turned on, and a fast reduction of the output current. For details, refer to IEC 61851-23.

[V2G-DC-668] While monitoring the CP State, if the SECC detects CP state B when according to 9.7.4.1.4 and 9.7.4.1.8 it should detect CP state C, the SECC shall close the TCP connection according to [V2G-DC-116].

NOTE 2 According to IEC 61851-23, if the SECC detects an unexpected CP state B, the SECC carries out, without any delay, an "EVSE-initiated emergency shutdown", which includes turning off the CP oscillator, if it is turned on, and a fast reduction of the output current. For details, please refer to IEC 61851-23.

[V2G-DC-669] At any point in time in the V2G communication session, if the SECC changes to CP state E but remains operational (i. e. no complete power outage, for example), the SECC shall close the TCP connection according to [V2G-DC-116].

[V2G-DC-670] At any point in time in the V2G communication session, if the SECC changes to CP state F, the SECC shall close the TCP connection according to [V2G-DC-116].

NOTE 3 Other errors in the charging application than the ones mentioned in this subclause might also cause an emergency shutdown by the EVSE. For further information, please refer to IEC 61851-23.

9.7.4.1.8 Normal SECC message sequence

[V2G-DC-432] After the TCP connection is established, in order to initiate the V2G communication session according to [V2G-DC-106], the SECC shall set the timeout V2G_SECC_Sequence_Timeout to the value defined in Table 76 for the supportedAppProtocolReq message. It shall then reset and start the V2G_SECC_Sequence_Timer, and it shall start to continuously monitor the V2G_SECC_Sequence_Timer for reaching or exceeding V2G_SECC_Sequence_Timeout. The SECC shall then wait for a supportedAppProtocolReq message.

Before receiving the first message from the EVCC, the SECC has not sent any response message yet. Therefore, the SECC should start its V2G_SECC_Sequence_Timer when starting to wait for the first message.

[V2G-DC-436] When waiting for a supportedAppProtocolReq message, if the SECC receives a supportedAppProtocolReq message, the SECC shall process this request message.

[V2G-DC-437] After the SECC has successfully processed a received supportedAppProtocolReq message, the SECC shall respond with a supportedAppProtocolRes message with ResponseCode according to Table 25. The SECC shall then wait for a SessionSetupReq message.

[V2G-DC-438] When waiting for a SessionSetupReq message, if the SECC receives a SessionSetupReq message, the SECC shall process this request message.

[V2G-DC-439] After the SECC has successfully processed a received SessionSetupReq message, the SECC shall respond with a SessionSetupRes message with ResponseCode equal to “OK”. The SECC shall then wait for a ServiceDiscoveryReq message or a SessionStopReq message.

[V2G-DC-440] When waiting for a ServiceDiscoveryReq message, if the SECC receives a ServiceDiscoveryReq message, the SECC shall process this request message.

[V2G-DC-441] After the SECC has successfully processed a received ServiceDiscoveryReq message, the SECC shall respond with a ServiceDiscoveryRes message with ResponseCode equal to “OK”. The SECC shall then wait for a ServicePaymentSelectionReq message or a SessionStopReq message.

[V2G-DC-443] When waiting for a ServicePaymentSelectionReq message, if the SECC receives a ServicePaymentSelectionReq message, the SECC shall process this request message.

[V2G-DC-444] After the SECC has successfully processed a received ServicePaymentSelectionReq message, the SECC shall respond with a ServicePaymentSelectionRes message with ResponseCode equal to “OK”. The SECC shall then wait for a ContractAuthenticationReq message or a SessionStopReq message.

[V2G-DC-494] When waiting for a ContractAuthenticationReq message, if the SECC receives a ContractAuthenticationReq message, the SECC shall process this request message.

[V2G-DC-495] After the SECC has successfully processed a received ContractAuthenticationReq message, the SECC shall respond with a ContractAuthenticationRes message with ResponseCode equal to “OK” and EVSEProcessing equal to “Finished”, if the contract authentication has finished on EVSE side. The SECC shall then wait for a ChargeParameterDiscoveryReq message or a SessionStopReq message.

[V2G-DC-497] After the SECC has successfully processed a received ContractAuthenticationReq message, the SECC shall respond with a ContractAuthenticationRes message with ResponseCode equal to “OK” and EVSEProcessing equal to “Ongoing”, if the contract authentication has not yet finished on EVSE side. The SECC shall then wait for another ContractAuthenticationReq message or a SessionStopReq message.

[V2G-DC-445] When waiting for a ChargeParameterDiscoveryReq message, if the SECC receives a ChargeParameterDiscoveryReq message, the SECC shall process this request message.

[V2G-DC-671] An EVSE that supports also other charging types than DC charging according to DIN SPEC 70121 on a connector shall, if by means of the parameter “EVRequestedEnergyTransferType” in ChargeParameterDiscoveryReq, the EVCC requests DC charging according to this document, apply a CP duty cycle of 5 % on this connector at the latest before responding with a ChargeParameterDiscoveryRes message with ResponseCode equal to “OK”, unless an error condition requires applying a different CP duty cycle or CP voltage.

[V2G-DC-965] After the SECC has successfully processed a received ChargeParameterDiscoveryReq message, the SECC shall respond with a ChargeParameterDiscoveryRes message with ResponseCode equal to “OK”, EVSEStatusCode equal “EVSE_IsolationMonitoringActive”, and EVSEProcessing equal to “Finished”, if the determination of the charge parameters has finished on EVSE side. The SECC shall then wait for a CableCheckReq message or a SessionStopReq message.

[V2G-DC-966] After the SECC has successfully processed a received ChargeParameterDiscoveryReq message, the SECC shall respond with a ChargeParameterDiscoveryRes message with ResponseCode equal to “OK”, EVSEStatusCode equal to “EVSE_IsolationMonitoringActive”, and EVSEProcessing equal to “Ongoing”, if the determination of the charge parameters has not yet finished on EVSE side. The SECC shall then wait for another ChargeParameterDiscoveryReq message or a SessionStopReq message.

[V2G-DC-967] After the SECC has sent a ChargeParameterDiscoveryRes message with ResponseCode equal to “OK”, EVSEStatusCode equal to “EVSE_Ready” or “EVSE_IsolationMonitoringActive”, and EVSEProcessing equal to “Finished”, if the SECC fails to detect CP State C as defined in IEC 61851-1 at the latest V2G_SECC_CPState_Detection_Timeout after receiving a CableCheckReq message, the SECC shall proceed as if the SECC itself wished to stop the Communication Session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-663].

[V2G-DC-454] When waiting for a CableCheckReq message, if the SECC receives a CableCheckReq message, the SECC shall process this request message.

[V2G-DC-455] After the SECC has successfully processed a received CableCheckReq message, the SECC shall respond with a CableCheckRes message with ResponseCode equal to “OK”, EVSEStatusCode equal to “EVSE_Ready”, and EVSEProcessing equal to “Finished”, if the Cable Check has finished on EVSE side. The SECC shall then wait for a PreChargeReq message or a SessionStopReq message.

[V2G-DC-499] After the SECC has successfully processed a received CableCheckReq message, the SECC shall respond with a CableCheckRes message with ResponseCode equal to “OK”, EVSEStatusCode equal to “EVSE_IsolationMonitoringActive”, and EVSEProcessing equal to “Ongoing”, if the Cable Check has not yet finished on EVSE side. The SECC shall then wait for another CableCheckReq message or a SessionStopReq message.

[V2G-DC-886] If the SECC has received a CableCheckReq and has not detected CP state C/D and the timeout as defined in [V2G-DC-967] has not met, the SECC should send a CableCheckRes as defined in [V2G-DC-499]. The SECC shall start V2G_SECC_CPState_Detection_Timer on reception of the first CableCheckReq request. Consecutive CableCheckReq shall have no influence on the timer and shall not trigger a restart of the timer. The SECC shall send a CableCheckRes with the following set of parameters:

— EVSEProcessing = ongoing [V2G-DC-499]

— EVSEStatusCode = EVSE_IsolationMonitoringActive [V2G-DC-499]

— EVSEIsolationStatus = invalid;

— EVSENotification = none;

— ResponseCode = OK.

[V2G-DC-1000] When the SECC detects CP State C/D, it shall stop the V2G_SECC_CPState_Detection_Timer and end monitoring it.

[V2G-DC-887] While CableCheck messages are exchanged and insulation resistance check by the EVSE is not finished the parameters shall be as follows:

— EVSEProcessing = ongoing;

— EVSEStatusCode= EVSE_IsolationMonitoringActive;

— EVSEIsolationStatus = invalid;

— EVSENotification = none;

— ResponseCode = OK.

[V2G-DC-888] If the EVSE finished the isolation check successfully, the parameters for the next CableCheckRes shall be as follows:

— EVSEProcessing = finished;

— EVSEStatusCode = EVSE_Ready;

— EVSEIsolationStatus = valid;

— EVSENotification = none;

— ResponseCode = OK.

[V2G-DC-889] If the EVSE finished the isolation check but identified a situation where it wants to give a warning, the parameters for the next CableCheckRes shall be as follows:

— EVSEProcessing = finished;

— EVSEStatusCode = EVSE_Ready;

— EVSEIsolationStatus = warning;

— EVSENotification = none;

— ResponseCode = OK.

[V2G-DC-890] If the SECC detects an isolation issue and the TCP connection is still established it shall send the next CableCheckRes EVSEIsolationStatus equal to “Fault” and ResponseCode equal to “FAILED”. Issue in isolation check (details see IEC61851-23):

— EVSEProcessing = finished;

— EVSEStatusCode = EVSE_Malfunction/EVSE_Shutdown [V2G-DC-663];

— EVSEIsolationStatus = fault;

— EVSENotification = none;

— ResponseCode = FAILED.

[V2G-DC-891] If the SECC wants to stop the process, it shall send the parameters in the next CableCheckRes as follows:

— EVSEProcessing = ongoing/finished;

— EVSEStatusCode = EVSE_Shutdown [V2G-DC-663];

— EVSEIsolationStatus = Invalid/valid/warning/fault;

— EVSENotification = none/StopCharging;

— ResponseCode = OK;

[V2G-DC-457] When waiting for a PreChargeReq message, if the SECC receives a PreChargeReq message, the SECC shall process this request message.

[V2G-DC-458] After the SECC has successfully processed a received PreChargeReq message, the SECC shall respond with a PreChargeRes message with ResponseCode equal to “OK”. The SECC shall then wait for another PreChargeReq message, a PowerDeliveryReq message, or a SessionStopReq message.

[V2G-DC-461] When waiting for a PowerDeliveryReq message, if the SECC receives a PowerDeliveryReq message, the SECC shall process this request message.

[V2G-DC-462] After the SECC has successfully processed a received PowerDeliveryReq message, and if the received PowerDeliveryReq message contained ReadyToChargeState equal to “TRUE”, the SECC shall respond with a PowerDeliveryRes message with ResponseCode equal to “OK”. The SECC shall then wait for a CurrentDemandReq message.

[V2G-DC-464] When waiting for a CurrentDemandReq message, if the SECC receives a CurrentDemandReq message, the SECC shall process this request message.

[V2G-DC-465] After the SECC has successfully processed a received CurrentDemandReq message, the SECC shall respond with a CurrentDemandRes message with ResponseCode equal to “OK”. The SECC shall then wait for another CurrentDemandReq message or a PowerDeliveryReq message.

[V2G-DC-459] After the SECC has successfully processed a received PowerDeliveryReq message, and if the received PowerDeliveryReq message contained ReadyToChargeState equal to “FALSE”, the SECC shall respond with a PowerDeliveryRes message with ResponseCode equal to “OK”. The SECC shall then wait for a WeldingDetectionReq message or a SessionStopReq message.

[V2G-DC-988] After the SECC has sent a PowerDeliveryRes message in response to a PowerDeliveryReq message with ReadyToChargeState equal to “FALSE”, if the SECC fails to detect CP State B as defined in IEC 61851-1 at the latest V2G_SECC_CPState_Detection_Timeout according to Table 78 after receiving the next request message, the SECC shall proceed as if the SECC itself wished to stop the communication session for a non-critical reason according to [V2G-DC-648] or [V2G-DC-663].

[V2G-DC-468] When waiting for a WeldingDetectionReq message, if the SECC receives a WeldingDetectionReq message, the SECC shall process this request message.

[V2G-DC-469] After the SECC has successfully processed a received WeldingDetectionReq message, the SECC shall respond with a WeldingDetectionRes message with ResponseCode equal to “OK”. The SECC shall then wait for another WeldingDetectionReq message or a SessionStopReq message.

[V2G-DC-450] When waiting for a SessionStopReq message, if the SECC receives a SessionStopReq message, the SECC shall process this request message.

[V2G-DC-451] After the SECC has successfully processed a received SessionStopReq message, the SECC shall respond with a SessionStopRes message with ResponseCode equal to “OK”.

[V2G-DC-968] After the SECC has sent a SessionStopRes message with ResponseCode equal to “OK”, it shall keep the CP oscillator switched on for V2G_SECC_CPOscillator_Retain_Time according to Table 78. The SECC shall then switch the CP oscillator off within 4 s after V2G_SECC_CPOscillator_Retain_Time expired.

[V2G-DC-962] If SECC detects state A, it shall then switch off the CP oscillator.

9.7.5 Message sequence example of successful DC charging session

This section gives an overview of the message flow between EV and EVSE during normal operation for a DC charging scenario. It does not provide an in-depth view on message timings and other constraints. The sequence shown in Figure 78 to Figure 81 gives a basic overview of the different communication phases and the respective message sequence. The contents of all messages are described in Clause 9. For reasons of clarity and readability, the individual messages are represented just by the message name rather than including the complete data structure and data types.

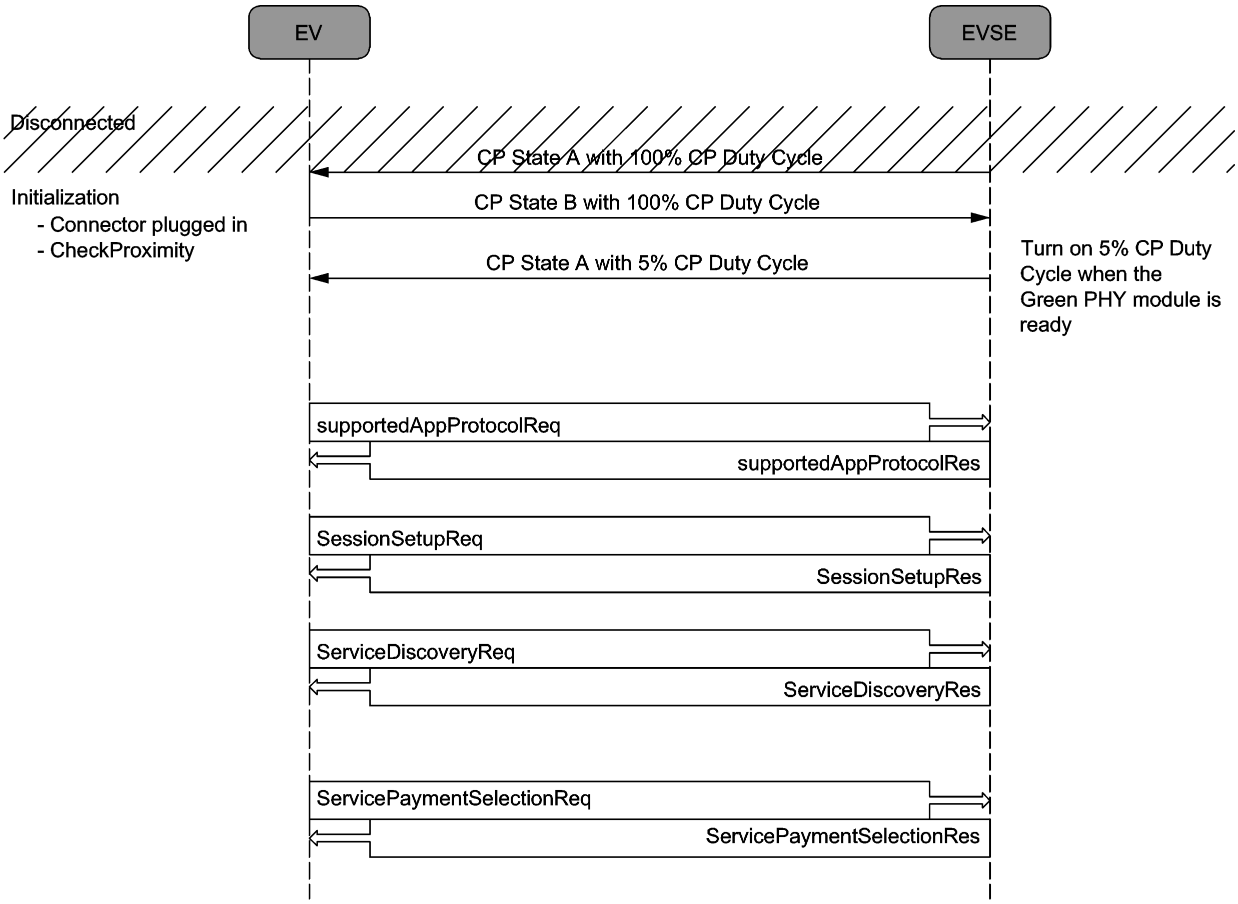


Figure 78 — Initialization sequence between EV and EVSE

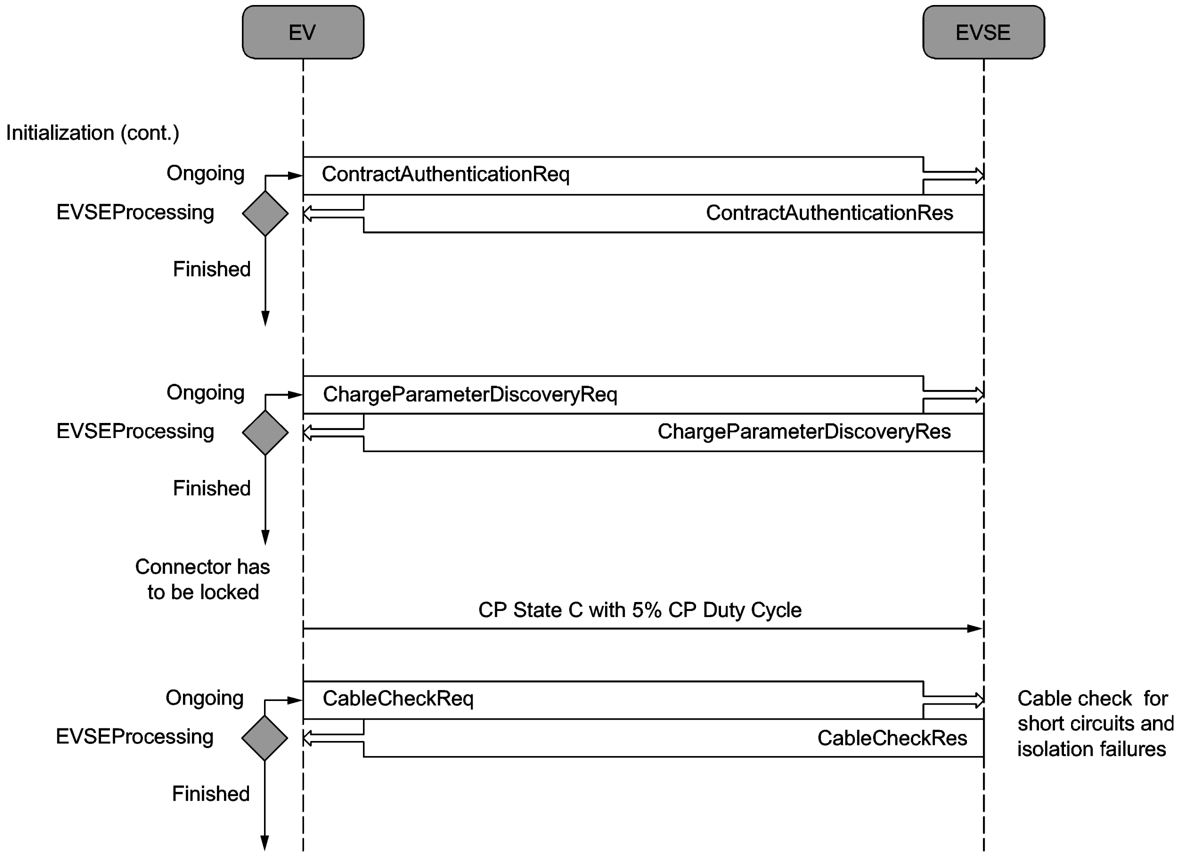


Figure 79 — Initialization sequence (cont.) between EV and EVSE

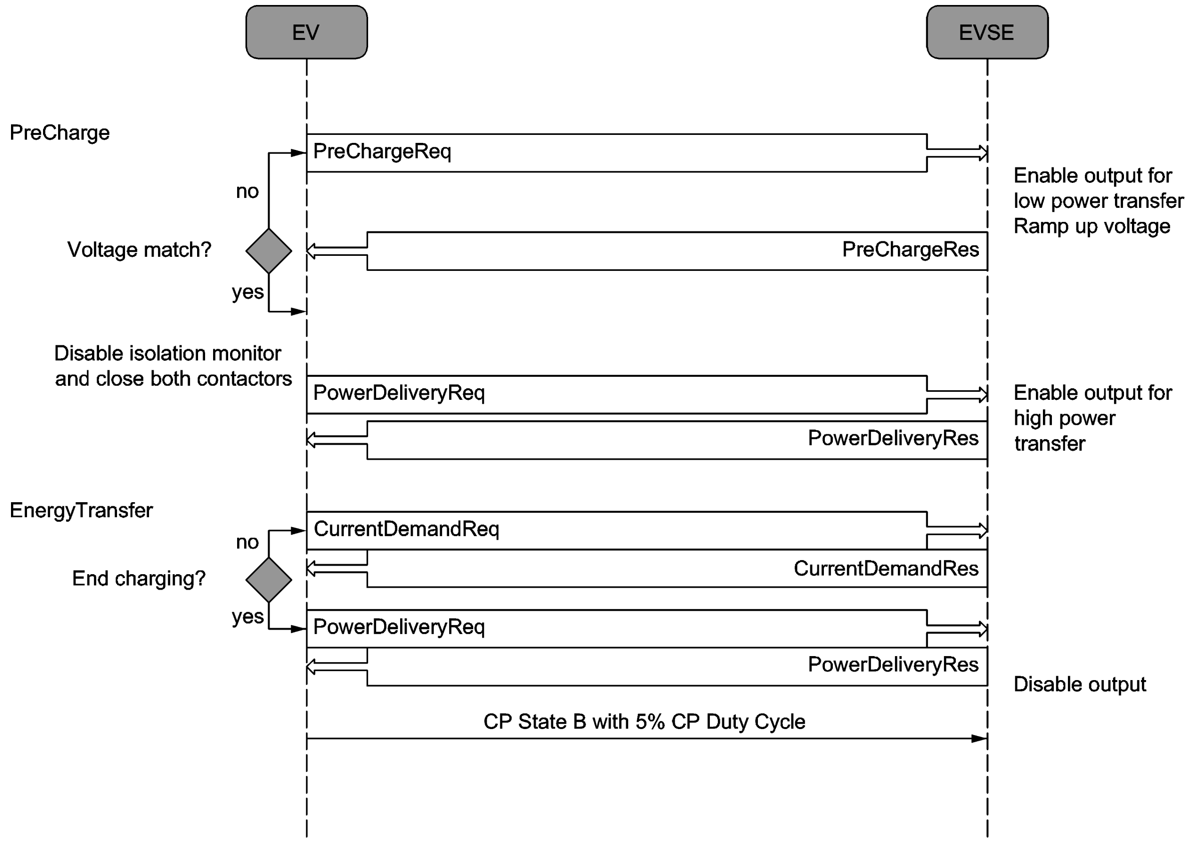
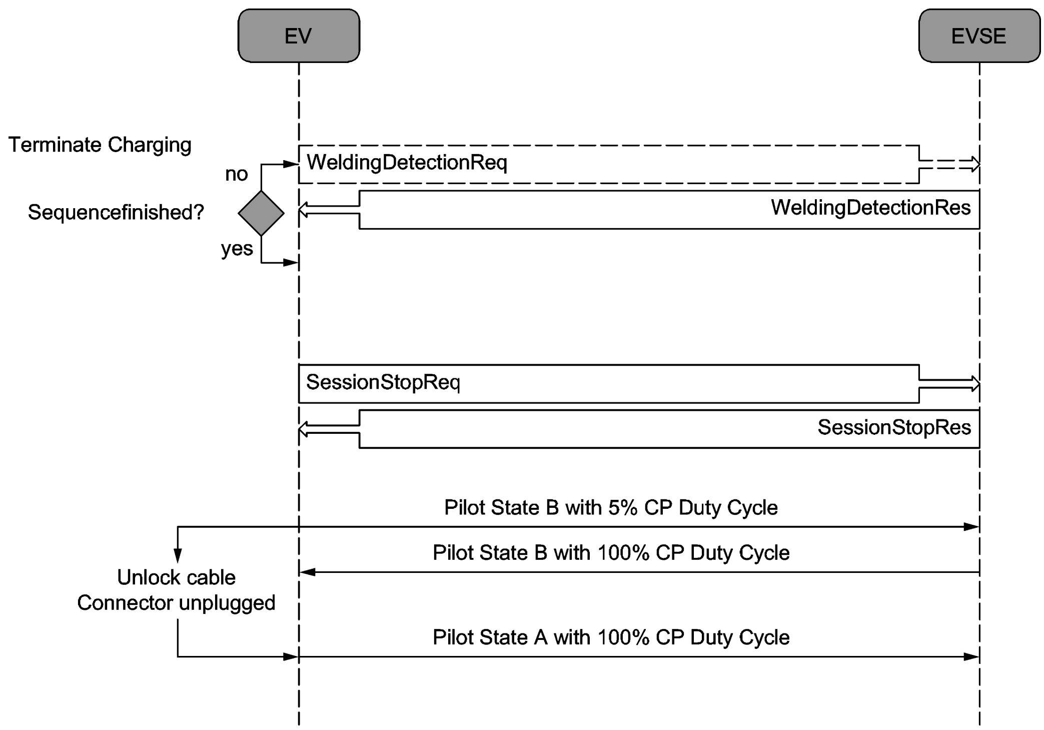


Figure 80 — Pre-charge and energy transfer sequence between EV and EVSE



NOTE For the EV, it is optional to perform a WeldingDetection

Figure 81 — Welding Detection and terminate charging sequence between EV and EVSE

Annex A
(normative)

XML Schema application layer messages

A.1 Overview

The V2G application layer message specification consists of four XML Schema documents with the following scope:

— “V2G_CI_AppProtocol”: Defines the protocol handshake messages;

— “V2G_CI_MsgDef”: Defines the message structure definition;

— “V2G_CI_MsgHeader”: Defines the message header;

— “V2G_CI_MsgBody”: Defines the message body;

— “V2G_CI_MsgDataTypes”: Defines the data types;

Figure A.1 shows the dependency graph for all five XML schema documents.

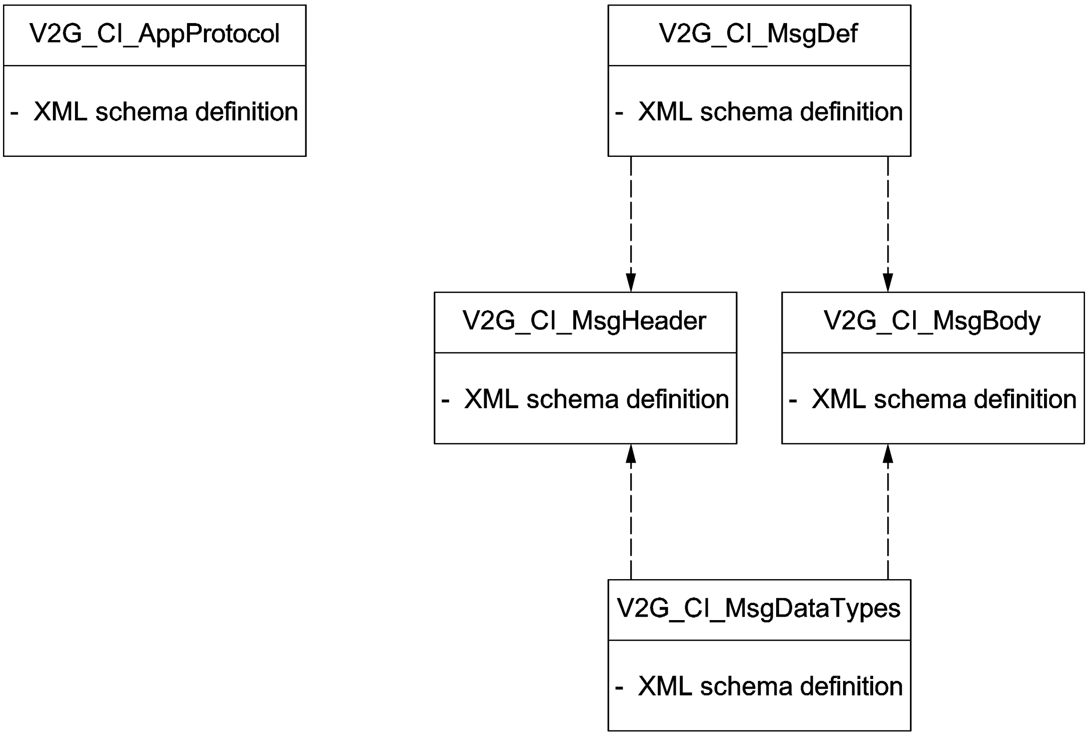


Figure A.1 — Dependency chart of the V2G CI XML schema definitions

A.2 V2G_CI_AppProtocol.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="urn:iso:15118:2:2010:AppProtocol"
targetNamespace="urn:iso:15118:2:2010:AppProtocol">
<xs:element name="supportedAppProtocolReq">
```

```
<xs:complexType>
<xs:sequence>
<xs:element name="AppProtocol" type="AppProtocolType" maxOccurs="20"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="supportedAppProtocolRes">
<xs:complexType>
<xs:sequence>
<xs:element name="ResponseCode" type="responseCodeType"/>
<xs:element name="SchemaID" type="idType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:complexType name="AppProtocolType">
<xs:sequence>
<xs:element name="ProtocolNamespace" type="protocolNamespaceType"/>
<xs:element name="VersionNumberMajor" type="xs:unsignedInt"/>
<xs:element name="VersionNumberMinor" type="xs:unsignedInt"/>
<xs:element name="SchemaID" type="idType"/>
<xs:element name="Priority" type="priorityType"/>
</xs:sequence>
</xs:complexType>
<xs:simpleType name="idType">
<xs:restriction base="xs:unsignedByte"/>
</xs:simpleType>
<xs:simpleType name="protocolNameType">
<xs:restriction base="xs:string">
<xs:maxLength value="30"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="protocolNamespaceType">
<xs:restriction base="xs:anyURI">
<xs:maxLength value="100"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="priorityType">
<xs:restriction base="xs:unsignedByte">
<xs:minInclusive value="1"/>
<xs:maxInclusive value="20"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="responseCodeType">
<xs:restriction base="xs:string">
<xs:enumeration value="OK_SuccessfulNegotiation"/>
<xs:enumeration value="OK_SuccessfulNegotiationWithMinorDeviation"/>
<xs:enumeration value="Failed_NoNegotiation"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>
```

A.3 V2G_CI_MsgDef.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="urn:din:70121:2012:MsgDef"
   xmlns="urn:din:70121:2012:MsgDef"
   xmlns:v2gci_h="urn:din:70121:2012:MsgHeader"
```

```
   elementFormDefault="qualified"
   attributeFormDefault="unqualified">
   <xs:import namespace="urn:din:70121:2012:MsgHeader"
    schemaLocation="V2G_CI_MsgHeader.xsd"/>
   <!-- Message Structure →
   <xs:element name="V2G_Message">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="Header" type="v2gci_h:MessageHeaderType"/>
            <xs:element name="Body" type="BodyType"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <!-- Body -->
   <xs:complexType name="BodyType">
      <xs:sequence>
         <xs:element ref="BodyElement" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:element name="BodyElement" type="BodyBaseType"/>
   <xs:complexType name="BodyBaseType" abstract="true"/>
</xs:schema>
```

A.4 V2G_CI_MsgHeader.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="urn:din:70121:2012:MsgHeader"
   xmlns="urn:din:70121:2012:MsgHeader"
   xmlns:v2gci_d="urn:din:70121:2012:MsgDef"
   xmlns:v2gci_t="urn:din:70121:2012:MsgDataTypes"
   xmlns:xmlsig="http://www.w3.org/2000/09/xmldsig#"
   elementFormDefault="qualified"
   attributeFormDefault="unqualified">
   <xs:import namespace="urn:din:70121:2012:MsgDef"
    schemaLocation="V2G_CI_MsgDef.xsd"/>
   <xs:import namespace="urn:din:70121:2012:MsgDataTypes"
    schemaLocation="V2G_CI_MsgDataTypes.xsd"/>
   <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
    schemaLocation="xmldsig-core-schema.xsd"/>
   <!-- Message Header -->
   <xs:complexType name="MessageHeaderType">
      <xs:sequence>
         <xs:element name="SessionID" type="v2gci_t:sessionIDType"/>
         <xs:element name="Notification" type="v2gci_t:NotificationType"
    minOccurs="0"/>
         <xs:element ref="xmlsig:Signature" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

A.5 V2G_CI_MsgBody.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="urn:din:70121:2012:MsgBody"
   xmlns:v2gci_d="urn:din:70121:2012:MsgDef"
   xmlns:v2gci_t="urn:din:70121:2012:MsgDataTypes"
   targetNamespace="urn:din:70121:2012:MsgBody"
```

```
elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xs:import namespace="urn:din:70121:2012:MsgDef"
 schemaLocation="V2G_CI_MsgDef.xsd"/>
<xs:import namespace="urn:din:70121:2012:MsgDataTypes"
 schemaLocation="V2G_CI_MsgDataTypes.xsd"/>
<!-- ...................... -->
<!-- Common Messages (AC/DC) -->
<!-- ...................... -->
<!-- -->
<!-- Session Setup -->
<!-- -->
<xs:element name="SessionSetupReq" type="SessionSetupReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="SessionSetupReqType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="EVCCID" type="v2gci_t:evccIDType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="SessionSetupRes" type="SessionSetupResType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="SessionSetupResType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
            <xs:element name="EVSEID" type="v2gci_t:evseIDType"/>
            <xs:element name="DateTimeNow" type="xs:long" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!-- -->
<!-- Service Discovery -->
<!-- -->
<xs:element name="ServiceDiscoveryReq" type="ServiceDiscoveryReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="ServiceDiscoveryReqType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="ServiceScope"
 type="v2gci_t:serviceScopeType" minOccurs="0"/>
            <xs:element name="ServiceCategory"
 type="v2gci_t:serviceCategoryType" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="ServiceDiscoveryRes" type="ServiceDiscoveryResType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="ServiceDiscoveryResType">
```

```
      <xs:complexContent>
         <xs:extension base="v2gci_d:BodyBaseType">
            <xs:sequence>
               <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
               <xs:element name="PaymentOptions"
 type="v2gci_t:PaymentOptionsType"/>
               <xs:element name="ChargeService"
 type="v2gci_t:ServiceChargeType"/>
               <xs:element name="ServiceList"
 type="v2gci_t:ServiceTagListType" minOccurs="0"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <!-- Service Detail -->
   <!-- -->
   <xs:element name="ServiceDetailReq" type="ServiceDetailReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
   <xs:complexType name="ServiceDetailReqType">
      <xs:complexContent>
         <xs:extension base="v2gci_d:BodyBaseType">
            <xs:sequence>
               <xs:element name="ServiceID" type="v2gci_t:serviceIDType"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="ServiceDetailRes" type="ServiceDetailResType"
 substitutionGroup="v2gci_d:BodyElement"/>
   <xs:complexType name="ServiceDetailResType">
      <xs:complexContent>
         <xs:extension base="v2gci_d:BodyBaseType">
            <xs:sequence>
               <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
               <xs:element name="ServiceID" type="v2gci_t:serviceIDType"/>
               <xs:element name="ServiceParameterList"
 type="v2gci_t:ServiceParameterListType" minOccurs="0"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <!-- Service Payment & Selection -->
   <!-- -->
   <xs:element name="ServicePaymentSelectionReq"
 type="ServicePaymentSelectionReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
   <xs:complexType name="ServicePaymentSelectionReqType">
      <xs:complexContent>
         <xs:extension base="v2gci_d:BodyBaseType">
            <xs:sequence>
               <xs:element name="SelectedPaymentOption"
 type="v2gci_t:paymentOptionType"/>
```

```
            <xs:element name="SelectedServiceList"
 type="v2gci_t:SelectedServiceListType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="ServicePaymentSelectionRes"
 type="ServicePaymentSelectionResType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="ServicePaymentSelectionResType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!-- -->
<!-- Payment Details -->
<!-- -->
<xs:element name="PaymentDetailsReq" type="PaymentDetailsReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="PaymentDetailsReqType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="ContractID" type="v2gci_t:contractIDType"/>
            <xs:element name="ContractSignatureCertChain"
 type="v2gci_t:CertificateChainType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="PaymentDetailsRes" type="PaymentDetailsResType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="PaymentDetailsResType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
            <xs:element name="GenChallenge"
 type="v2gci_t:genChallengeType"/>
            <xs:element name="DateTimeNow" type="xs:long"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!-- -->
<!-- Contract Authentification -->
<!-- -->
<xs:element name="ContractAuthenticationReq"
 type="ContractAuthenticationReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="ContractAuthenticationReqType">
```

```
      <xs:complexContent>
         <xs:extension base="v2gci_d:BodyBaseType">
            <xs:sequence>
               <xs:element name="GenChallenge"
 type="v2gci_t:genChallengeType" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="Id" type="xs:IDREF"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="ContractAuthenticationRes"
    type="ContractAuthenticationResType"
    substitutionGroup="v2gci_d:BodyElement"/>
   <xs:complexType name="ContractAuthenticationResType">
      <xs:complexContent>
         <xs:extension base="v2gci_d:BodyBaseType">
            <xs:sequence>
               <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
               <xs:element name="EVSEProcessing"
 type="v2gci_t:EVSEProcessingType"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <!-- -->
   <!-- Charge Parameter Discovery -->
   <!-- -->
   <xs:element name="ChargeParameterDiscoveryReq"
    type="ChargeParameterDiscoveryReqType"
    substitutionGroup="v2gci_d:BodyElement"/>
   <xs:complexType name="ChargeParameterDiscoveryReqType">
      <xs:complexContent>
         <xs:extension base="v2gci_d:BodyBaseType">
            <xs:sequence>
               <xs:element name="EVRequestedEnergyTransferType"
 type="v2gci_t:EVRequestedEnergyTransferType"/>
               <xs:element ref="v2gci_t:EVChargeParameter"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="ChargeParameterDiscoveryRes"
    type="ChargeParameterDiscoveryResType"
    substitutionGroup="v2gci_d:BodyElement"/>
   <xs:complexType name="ChargeParameterDiscoveryResType">
      <xs:complexContent>
         <xs:extension base="v2gci_d:BodyBaseType">
            <xs:sequence>
               <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
               <xs:element name="EVSEProcessing"
 type="v2gci_t:EVSEProcessingType"/>
               <xs:element ref="v2gci_t:SASchedules"/>
               <xs:element ref="v2gci_t:EVSEChargeParameter"/>
            </xs:sequence>
         </xs:extension>
```

```
   </xs:complexContent>
</xs:complexType>
<!-- -->
<!-- Power Delivery -->
<!-- -->
<xs:element name="PowerDeliveryReq" type="PowerDeliveryReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="PowerDeliveryReqType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="ReadyToChargeState" type="xs:boolean"/>
            <xs:element name="ChargingProfile"
 type="v2gci_t:ChargingProfileType" minOccurs="0"/>
            <xs:element ref="v2gci_t:EVPowerDeliveryParameter"
 minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="PowerDeliveryRes" type="PowerDeliveryResType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="PowerDeliveryResType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
            <xs:element ref="v2gci_t:EVSEStatus"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!-- -->
<!-- Charging Status -->
<!-- -->
<xs:element name="ChargingStatusReq" type="ChargingStatusReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="ChargingStatusReqType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="ChargingStatusRes" type="ChargingStatusResType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="ChargingStatusResType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
            <xs:element name="EVSEID" type="v2gci_t:evseIDType"/>
            <xs:element name="SAScheduleTupleID" type="v2gci_t:SAIDType"/>
            <xs:element name="EVSEMaxCurrent"
 type="v2gci_t:PhysicalValueType" minOccurs="0"/>
```

```
            <xs:element name="MeterInfo" type="v2gci_t:MeterInfoType"
 minOccurs="0"/>
            <xs:element name="ReceiptRequired" type="xs:boolean"/>
            <xs:element name="AC_EVSEStatus"
 type="v2gci_t:AC_EVSEStatusType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!-- -->
<!-- Metering Receipt -->
<!-- -->
<xs:element name="MeteringReceiptReq" type="MeteringReceiptReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="MeteringReceiptReqType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="SessionID" type="v2gci_t:sessionIDType"/>
            <xs:element name="SAScheduleTupleID" type="v2gci_t:SAIDType"
 minOccurs="0"/>
            <xs:element name="MeterInfo" type="v2gci_t:MeterInfoType"/>
         </xs:sequence>
         <xs:attribute name="Id" type="xs:IDREF"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="MeteringReceiptRes" type="MeteringReceiptResType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="MeteringReceiptResType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
            <xs:element name="AC_EVSEStatus"
 type="v2gci_t:AC_EVSEStatusType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!-- -->
<!-- SessionStop -->
<!-- -->
<xs:element name="SessionStopReq" type="SessionStopType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="SessionStopType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="SessionStopRes" type="SessionStopResType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="SessionStopResType">
   <xs:complexContent>
```

```
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!-- -->
<!-- Certificate Update -->
<!-- -->
<xs:element name="CertificateUpdateReq" type="CertificateUpdateReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="CertificateUpdateReqType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="ContractSignatureCertChain"
 type="v2gci_t:CertificateChainType"/>
            <xs:element name="ContractID" type="v2gci_t:contractIDType"/>
            <xs:element name="ListOfRootCertificateIDs"
 type="v2gci_t:ListOfRootCertificateIDsType"/>
            <xs:element name="DHParams" type="v2gci_t:dHParamsType"/>
         </xs:sequence>
         <xs:attribute name="Id" type="xs:IDREF"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="CertificateUpdateRes" type="CertificateUpdateResType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="CertificateUpdateResType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
         <xs:sequence>
            <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
            <xs:element name="ContractSignatureCertChain"
 type="v2gci_t:CertificateChainType"/>
            <xs:element name="ContractSignatureEncryptedPrivateKey"
 type="v2gci_t:privateKeyType"/>
            <xs:element name="DHParams" type="v2gci_t:dHParamsType"/>
            <xs:element name="ContractID" type="v2gci_t:contractIDType"/>
            <xs:element name="RetryCounter" type="xs:short"/>
         </xs:sequence>
         <xs:attribute name="Id" type="xs:IDREF" use="required"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!-- -->
<!-- Certificate Installation -->
<!-- -->
<xs:element name="CertificateInstallationReq"
 type="CertificateInstallationReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="CertificateInstallationReqType">
   <xs:complexContent>
      <xs:extension base="v2gci_d:BodyBaseType">
```

```
          <xs:sequence>
             <xs:element name="OEMProvisioningCert"
 type="v2gci_t:certificateType"/>
             <xs:element name="ListOfRootCertificateIDs"
 type="v2gci_t:ListOfRootCertificateIDsType"/>
             <xs:element name="DHParams" type="v2gci_t:dHParamsType"/>
          </xs:sequence>
          <xs:attribute name="Id" type="xs:IDREF" />
       </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:element name="CertificateInstallationRes"
 type="CertificateInstallationResType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="CertificateInstallationResType">
    <xs:complexContent>
       <xs:extension base="v2gci_d:BodyBaseType">
          <xs:sequence>
             <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
             <xs:element name="ContractSignatureCertChain"
 type="v2gci_t:CertificateChainType"/>
             <xs:element name="ContractSignatureEncryptedPrivateKey"
 type="v2gci_t:privateKeyType"/>
             <xs:element name="DHParams" type="v2gci_t:dHParamsType"/>
             <xs:element name="ContractID" type="v2gci_t:contractIDType"/>
          </xs:sequence>
          <xs:attribute name="Id" type="xs:IDREF" use="required"/>
       </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ...................... -->
<!-- DC-Messages -->
<!-- ...................... -->
<!-- -->
<!-- Cable Check -->
<!-- -->
<xs:element name="CableCheckReq" type="CableCheckReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="CableCheckReqType">
    <xs:complexContent>
       <xs:extension base="v2gci_d:BodyBaseType">
          <xs:sequence>
             <xs:element name="DC_EVStatus"
 type="v2gci_t:DC_EVStatusType"/>
          </xs:sequence>
       </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:element name="CableCheckRes" type="CableCheckResType"
 substitutionGroup="v2gci_d:BodyElement"/>
<xs:complexType name="CableCheckResType">
    <xs:complexContent>
       <xs:extension base="v2gci_d:BodyBaseType">
          <xs:sequence>
             <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
```

```
                <xs:element name="DC_EVSEStatus"
 type="v2gci_t:DC_EVSEStatusType"/>
                <xs:element name="EVSEProcessing"
 type="v2gci_t:EVSEProcessingType"/>
             </xs:sequence>
          </xs:extension>
       </xs:complexContent>
    </xs:complexType>
    <!-- -->
    <!-- Pre-Charge -->
    <!-- -->
    <xs:element name="PreChargeReq" type="PreChargeReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
    <xs:complexType name="PreChargeReqType">
       <xs:complexContent>
          <xs:extension base="v2gci_d:BodyBaseType">
             <xs:sequence>
                <xs:element name="DC_EVStatus"
 type="v2gci_t:DC_EVStatusType"/>
                <xs:element name="EVTargetVoltage"
 type="v2gci_t:PhysicalValueType"/>
                <xs:element name="EVTargetCurrent"
 type="v2gci_t:PhysicalValueType"/>
             </xs:sequence>
          </xs:extension>
       </xs:complexContent>
    </xs:complexType>
    <xs:element name="PreChargeRes" type="PreChargeResType"
 substitutionGroup="v2gci_d:BodyElement"/>
    <xs:complexType name="PreChargeResType">
       <xs:complexContent>
          <xs:extension base="v2gci_d:BodyBaseType">
             <xs:sequence>
                <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
                <xs:element name="DC_EVSEStatus"
 type="v2gci_t:DC_EVSEStatusType"/>
                <xs:element name="EVSEPresentVoltage"
 type="v2gci_t:PhysicalValueType"/>
             </xs:sequence>
          </xs:extension>
       </xs:complexContent>
    </xs:complexType>
    <!-- -->
    <!-- Current Demand -->
    <!-- -->
    <xs:element name="CurrentDemandReq" type="CurrentDemandReqType"
 substitutionGroup="v2gci_d:BodyElement"/>
    <xs:complexType name="CurrentDemandReqType">
       <xs:complexContent>
          <xs:extension base="v2gci_d:BodyBaseType">
             <xs:sequence>
                <xs:element name="DC_EVStatus"
 type="v2gci_t:DC_EVStatusType"/>
                <xs:element name="EVTargetCurrent"
 type="v2gci_t:PhysicalValueType"/>
```

```
              <xs:element name="EVMaximumVoltageLimit"
 type="v2gci_t:PhysicalValueType" minOccurs="0"/>
              <xs:element name="EVMaximumCurrentLimit"
 type="v2gci_t:PhysicalValueType" minOccurs="0"/>
              <xs:element name="EVMaximumPowerLimit"
 type="v2gci_t:PhysicalValueType" minOccurs="0"/>
              <xs:element name="BulkChargingComplete" type="xs:boolean"
 minOccurs="0"/>
              <xs:element name="ChargingComplete" type="xs:boolean"/>
              <xs:element name="RemainingTimeToFullSoC"
 type="v2gci_t:PhysicalValueType" minOccurs="0"/>
              <xs:element name="RemainingTimeToBulkSoC"
 type="v2gci_t:PhysicalValueType" minOccurs="0"/>
              <xs:element name="EVTargetVoltage"
 type="v2gci_t:PhysicalValueType"/>
           </xs:sequence>
        </xs:extension>
     </xs:complexContent>
  </xs:complexType>
  <xs:element name="CurrentDemandRes" type="CurrentDemandResType"
   substitutionGroup="v2gci_d:BodyElement"/>
  <xs:complexType name="CurrentDemandResType">
     <xs:complexContent>
        <xs:extension base="v2gci_d:BodyBaseType">
           <xs:sequence>
              <xs:element name="ResponseCode"
 type="v2gci_t:responseCodeType"/>
              <xs:element name="DC_EVSEStatus"
 type="v2gci_t:DC_EVSEStatusType"/>
              <xs:element name="EVSEPresentVoltage"
 type="v2gci_t:PhysicalValueType"/>
              <xs:element name="EVSEPresentCurrent"
 type="v2gci_t:PhysicalValueType"/>
              <xs:element name="EVSECurrentLimitAchieved"
 type="xs:boolean"/>
              <xs:element name="EVSEVoltageLimitAchieved"
 type="xs:boolean"/>
              <xs:element name="EVSEPowerLimitAchieved" type="xs:boolean"/>
              <xs:element name="EVSEMaximumVoltageLimit"
 type="v2gci_t:PhysicalValueType" minOccurs="0"/>
              <xs:element name="EVSEMaximumCurrentLimit"
 type="v2gci_t:PhysicalValueType" minOccurs="0"/>
              <xs:element name="EVSEMaximumPowerLimit"
 type="v2gci_t:PhysicalValueType" minOccurs="0"/>
           </xs:sequence>
        </xs:extension>
     </xs:complexContent>
  </xs:complexType>
  <!-- -->
  <!-- Welding Detection -->
  <!-- -->
  <xs:element name="WeldingDetectionReq" type="WeldingDetectionReqType"
   substitutionGroup="v2gci_d:BodyElement"/>
  <xs:complexType name="WeldingDetectionReqType">
     <xs:complexContent>
        <xs:extension base="v2gci_d:BodyBaseType">
           <xs:sequence>
```

```
                <xs:element name="DC_EVStatus"
     type="v2gci_t:DC_EVStatusType"/>
             </xs:sequence>
          </xs:extension>
       </xs:complexContent>
    </xs:complexType>
    <xs:element name="WeldingDetectionRes" type="WeldingDetectionResType"
     substitutionGroup="v2gci_d:BodyElement"/>
    <xs:complexType name="WeldingDetectionResType">
       <xs:complexContent>
          <xs:extension base="v2gci_d:BodyBaseType">
             <xs:sequence>
                <xs:element name="ResponseCode"
     type="v2gci_t:responseCodeType"/>
                <xs:element name="DC_EVSEStatus"
     type="v2gci_t:DC_EVSEStatusType"/>
                <xs:element name="EVSEPresentVoltage"
     type="v2gci_t:PhysicalValueType"/>
             </xs:sequence>
          </xs:extension>
       </xs:complexContent>
    </xs:complexType>
</xs:schema>
```

A.6 V2G_CI_MsgDataTypes.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
     xmlns="urn:din:70121:2012:MsgDataTypes"
    xmlns:v2gci_b="urn:din:70121:2012:MsgBody"
    targetNamespace="urn:din:70121:2012:MsgDataTypes"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">
    <xs:import namespace="urn:din:70121:2012:MsgBody"
     schemaLocation="V2G_CI_MsgBody.xsd"/>
    <!-- ================== -->
    <!-- Complex types -->
    <!-- ================== -->
    <!--                  -->
    <!-- service-related types -->
    <!--                  -->
    <xs:complexType name="ServiceType">
       <xs:sequence>
          <xs:element name="ServiceTag" type="ServiceTagType"/>
          <xs:element name="FreeService" type="xs:boolean"/>
       </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceTagListType">
       <xs:sequence>
          <xs:element name="Service" type="ServiceType"
     maxOccurs="unbounded"/>
       </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceTagType">
       <xs:sequence>
          <xs:element name="ServiceID" type="serviceIDType"/>
          <xs:element name="ServiceName" type="serviceNameType"
     minOccurs="0"/>
```

```
        <xs:element name="ServiceCategory" type="serviceCategoryType"/>
        <xs:element name="ServiceScope" type="serviceScopeType"
 minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="SelectedServiceListType">
      <xs:sequence>
         <xs:element name="SelectedService" type="SelectedServiceType"
 maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="SelectedServiceType">
      <xs:sequence>
         <xs:element name="ServiceID" type="serviceIDType"/>
         <xs:element name="ParameterSetID" type="xs:short" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="ServiceParameterListType">
      <xs:sequence>
         <xs:element name="ParameterSet" type="ParameterSetType"
 maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="ParameterSetType">
      <xs:sequence>
         <xs:element name="ParameterSetID" type="xs:short"/>
         <xs:element name="Parameter" type="ParameterType"
 maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="ParameterType">
      <xs:choice>
         <xs:element name="boolValue" type="xs:boolean"/>
         <xs:element name="byteValue" type="xs:byte"/>
         <xs:element name="shortValue" type="xs:short"/>
         <xs:element name="intValue" type="xs:int"/>
         <xs:element name="physicalValue" type="PhysicalValueType"/>
         <xs:element name="stringValue" type="xs:string"/>
      </xs:choice>
      <xs:attribute name="Name" type="xs:string" use="required"/>
      <xs:attribute name="ValueType" type="valueType" use="required"/>
   </xs:complexType>
   <xs:simpleType name="valueType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="bool"/>
         <xs:enumeration value="byte"/>
         <xs:enumeration value="short"/>
         <xs:enumeration value="int"/>
         <xs:enumeration value="physicalValue"/>
         <xs:enumeration value="string"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:element name="ServiceCharge" type="ServiceChargeType"/>
   <xs:complexType name="ServiceChargeType">
      <xs:complexContent>
         <xs:extension base="ServiceType">
            <xs:sequence>
```

```
            <xs:element name="EnergyTransferType"
 type="EVSESupportedEnergyTransferType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!--                 -->
<!-- security related types -->
<!--                 -->
<xs:complexType name="CertificateChainType">
   <xs:sequence>
      <xs:element name="Certificate" type="certificateType"/>
      <xs:element name="SubCertificates" type="SubCertificatesType"
 minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
<xs:complexType name="SubCertificatesType">
   <xs:sequence>
      <xs:element name="Certificate" type="certificateType"
 maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
<xs:complexType name="ListOfRootCertificateIDsType">
   <xs:sequence>
      <xs:element name="RootCertificateID" type="rootCertificateIDType"
 maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
<!--                 -->
<!-- metering related types -->
<!--                 -->
<xs:complexType name="MeterInfoType">
   <xs:sequence>
      <xs:element name="MeterID" type="meterIDType" />
      <xs:element name="MeterReading" type="PhysicalValueType"
 minOccurs="0"/>
      <xs:element name="SigMeterReading" type="sigMeterReadingType"
 minOccurs="0"/>
      <xs:element name="MeterStatus" type="meterStatusType"
 minOccurs="0"/>
      <xs:element name="TMeter" type="xs:long" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
<xs:simpleType name="meterStatusType">
   <xs:restriction base="xs:short"/>
</xs:simpleType>
<!--                 -->
<!-- Physical value type -->
<!--                 -->
<xs:complexType name="PhysicalValueType">
   <xs:sequence>
      <xs:element name="Multiplier" type="unitMultiplierType"/>
      <xs:element name="Unit" type="unitSymbolType" minOccurs="0"/>
      <xs:element name="Value" type="xs:short"/>
   </xs:sequence>
</xs:complexType>
<!--           -->
```

```
<!-- header related types -->
<!--            -->
<xs:complexType name="NotificationType">
   <xs:sequence>
      <xs:element name="FaultCode" type="faultCodeType"/>
      <xs:element name="FaultMsg" type="faultMsgType" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
<!--            -->
<!-- Tariff related types -->
<!--            -->
<xs:complexType name="SASchedulesType" abstract="true"/>
<xs:element name="SASchedules" type="SASchedulesType"/>
<xs:element name="SAScheduleList" type="SAScheduleListType"
 substitutionGroup="SASchedules"/>
<xs:complexType name="SAScheduleListType">
   <xs:complexContent>
      <xs:extension base="SASchedulesType">
         <xs:sequence>
            <xs:element name="SAScheduleTuple" type="SAScheduleTupleType"
 maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:complexType name="SAScheduleTupleType">
   <xs:sequence>
      <xs:element name="SAScheduleTupleID" type="SAIDType"/>
      <xs:element name="PMaxSchedule" type="PMaxScheduleType"/>
      <xs:element name="SalesTariff" type="SalesTariffType"
 minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
<xs:complexType name="SalesTariffType">
   <xs:sequence>
      <xs:element name="SalesTariffID" type="SAIDType"/>
      <xs:element name="SalesTariffDescription"
 type="tariffDescriptionType" minOccurs="0"/>
      <xs:element name="NumEPriceLevels" type="xs:unsignedByte"/>
      <xs:element ref="SalesTariffEntry" maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="Id" type="xs:IDREF" use="required"/>
</xs:complexType>
<xs:complexType name="PMaxScheduleType">
   <xs:sequence>
      <xs:element name="PMaxScheduleID" type="SAIDType"/>
      <xs:element ref="PMaxScheduleEntry" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
<xs:element name="Entry" type="EntryType"/>
<xs:complexType name="EntryType" abstract="true">
   <xs:sequence>
      <xs:element ref="TimeInterval"/>
   </xs:sequence>
</xs:complexType>
<xs:element name="SalesTariffEntry" type="SalesTariffEntryType"
 substitutionGroup="Entry"/>
```

```
<xs:complexType name="SalesTariffEntryType">
   <xs:complexContent>
      <xs:extension base="EntryType">
         <xs:sequence>
            <xs:element name="EPriceLevel" type="xs:unsignedByte"/>
            <xs:element name="ConsumptionCost" type="ConsumptionCostType"
 minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="PMaxScheduleEntry" type="PMaxScheduleEntryType"
 substitutionGroup="Entry"/>
<xs:complexType name="PMaxScheduleEntryType">
   <xs:complexContent>
      <xs:extension base="EntryType">
         <xs:sequence>
            <xs:element name="PMax" type="PMaxType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:complexType name="IntervalType" abstract="true"/>
<xs:element name="TimeInterval" type="IntervalType"/>
<xs:element name="RelativeTimeInterval" type="RelativeTimeIntervalType"
 substitutionGroup="TimeInterval"/>
<xs:complexType name="RelativeTimeIntervalType">
   <xs:complexContent>
      <xs:extension base="IntervalType">
         <xs:sequence>
            <xs:element name="start" type="xs:unsignedInt"/>
            <xs:element name="duration" type="xs:unsignedInt"
 minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:complexType name="ConsumptionCostType">
   <xs:sequence>
      <xs:element name="startValue" type="xs:unsignedInt"/>
      <xs:element name="Cost" type="CostType" minOccurs="0"
 maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
<xs:complexType name="CostType">
   <xs:sequence>
      <xs:element name="costKind" type="costKindType"/>
      <xs:element name="amount" type="xs:unsignedInt"/>
      <xs:element name="amountMultiplier" type="unitMultiplierType"
 minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
<!--           -->
<!-- EV/EVSE related types -->
<!--           -->
<xs:complexType name="EVSEStatusType" abstract="true"/>
<xs:element name="EVSEStatus" type="EVSEStatusType"/>
```

```
<xs:element name="AC_EVSEStatus" type="AC_EVSEStatusType"
 substitutionGroup="EVSEStatus"/>
<xs:complexType name="AC_EVSEStatusType">
   <xs:complexContent>
      <xs:extension base="EVSEStatusType">
         <xs:sequence>
            <xs:element name="PowerSwitchClosed" type="xs:boolean"/>
            <xs:element name="RCD" type="xs:boolean"/>
            <xs:element name="NotificationMaxDelay"
 type="xs:unsignedInt"/>
            <xs:element name="EVSENotification"
 type="EVSENotificationType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:complexType name="EVStatusType" abstract="true"/>
<xs:element name="EVStatus" type="EVStatusType"/>
<xs:element name="DC_EVSEStatus" type="DC_EVSEStatusType"
 substitutionGroup="EVSEStatus"/>
<xs:complexType name="DC_EVSEStatusType">
   <xs:complexContent>
      <xs:extension base="EVSEStatusType">
         <xs:sequence>
            <xs:element name="EVSEIsolationStatus"
 type="isolationLevelType" minOccurs="0"/>
            <xs:element name="EVSEStatusCode"
 type="DC_EVSEStatusCodeType"/>
            <xs:element name="NotificationMaxDelay"
 type="xs:unsignedInt"/>
            <xs:element name="EVSENotification"
 type="EVSENotificationType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="DC_EVStatus" type="DC_EVStatusType"
 substitutionGroup="EVStatus"/>
<xs:complexType name="DC_EVStatusType">
   <xs:complexContent>
      <xs:extension base="EVStatusType">
         <xs:sequence>
            <xs:element name="EVReady" type="xs:boolean"/>
            <xs:element name="EVCabinConditioning" type="xs:boolean"
 minOccurs="0"/>
            <xs:element name="EVRESSConditioning" type="xs:boolean"
 minOccurs="0"/>
            <xs:element name="EVErrorCode" type="DC_EVErrorCodeType"/>
            <xs:element name="EVRESSSOC" type="percentValueType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!--           -->
<!-- EVSE/EV Charge Parameter related types -->
<!--          -->
<xs:complexType name="EVChargeParameterType" abstract="true"/>
```

```
<xs:element name="EVChargeParameter" type="EVChargeParameterType"/>
<xs:element name="AC_EVChargeParameter" type="AC_EVChargeParameterType"
 substitutionGroup="EVChargeParameter"/>
<xs:complexType name="AC_EVChargeParameterType">
   <xs:complexContent>
      <xs:extension base="EVChargeParameterType">
         <xs:sequence>
            <xs:element name="DepartureTime" type="xs:unsignedInt"/>
            <xs:element name="EAmount" type="PhysicalValueType"/>
            <xs:element name="EVMaxVoltage" type="PhysicalValueType"/>
            <xs:element name="EVMaxCurrent" type="PhysicalValueType"/>
            <xs:element name="EVMinCurrent" type="PhysicalValueType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="DC_EVChargeParameter" type="DC_EVChargeParameterType"
 substitutionGroup="EVChargeParameter"/>
<xs:complexType name="DC_EVChargeParameterType">
   <xs:complexContent>
      <xs:extension base="EVChargeParameterType">
         <xs:sequence>
            <xs:element name="DC_EVStatus" type="DC_EVStatusType"/>
            <xs:element name="EVMaximumCurrentLimit"
 type="PhysicalValueType"/>
            <xs:element name="EVMaximumPowerLimit"
 type="PhysicalValueType" minOccurs="0"/>
            <xs:element name="EVMaximumVoltageLimit"
 type="PhysicalValueType"/>
            <xs:element name="EVEnergyCapacity" type="PhysicalValueType"
 minOccurs="0"/>
            <xs:element name="EVEnergyRequest" type="PhysicalValueType"
 minOccurs="0"/>
            <xs:element name="FullSOC" type="percentValueType"
 minOccurs="0"/>
            <xs:element name="BulkSOC" type="percentValueType"
 minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:complexType name="EVSEChargeParameterType" abstract="true"/>
<xs:element name="EVSEChargeParameter" type="EVSEChargeParameterType"/>
<xs:element name="AC_EVSEChargeParameter"
 type="AC_EVSEChargeParameterType"
 substitutionGroup="EVSEChargeParameter"/>
<xs:complexType name="AC_EVSEChargeParameterType">
   <xs:complexContent>
      <xs:extension base="EVSEChargeParameterType">
         <xs:sequence>
            <xs:element name="AC_EVSEStatus" type="AC_EVSEStatusType"/>
            <xs:element name="EVSEMaxVoltage" type="PhysicalValueType"/>
            <xs:element name="EVSEMaxCurrent" type="PhysicalValueType"/>
            <xs:element name="EVSEMinCurrent" type="PhysicalValueType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
```

```
</xs:complexType>
<xs:element name="DC_EVSEChargeParameter"
 type="DC_EVSEChargeParameterType"
 substitutionGroup="EVSEChargeParameter"/>
<xs:complexType name="DC_EVSEChargeParameterType">
   <xs:complexContent>
      <xs:extension base="EVSEChargeParameterType">
         <xs:sequence>
            <xs:element name="DC_EVSEStatus" type="DC_EVSEStatusType"/>
            <xs:element name="EVSEMaximumCurrentLimit"
 type="PhysicalValueType"/>
            <xs:element name="EVSEMaximumPowerLimit"
 type="PhysicalValueType" minOccurs="0"/>
            <xs:element name="EVSEMaximumVoltageLimit"
 type="PhysicalValueType"/>
            <xs:element name="EVSEMinimumCurrentLimit"
 type="PhysicalValueType"/>
            <xs:element name="EVSEMinimumVoltageLimit"
 type="PhysicalValueType"/>
            <xs:element name="EVSECurrentRegulationTolerance"
 type="PhysicalValueType" minOccurs="0"/>
            <xs:element name="EVSEPeakCurrentRipple"
 type="PhysicalValueType"/>
            <xs:element name="EVSEEnergyToBeDelivered"
 type="PhysicalValueType" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!--            -->
<!-- EV Power Delivery related types -->
<!--           -->
<xs:complexType name="EVPowerDeliveryParameterType" abstract="true"/>
   <xs:element name="EVPowerDeliveryParameter"
 type="EVPowerDeliveryParameterType"/>
<xs:element name="DC_EVPowerDeliveryParameter"
 type="DC_EVPowerDeliveryParameterType"
 substitutionGroup="EVPowerDeliveryParameter"/>
<xs:complexType name="DC_EVPowerDeliveryParameterType">
   <xs:complexContent>
      <xs:extension base="EVPowerDeliveryParameterType">
         <xs:sequence>
            <xs:element name="DC_EVStatus" type="DC_EVStatusType"/>
            <xs:element name="BulkChargingComplete" type="xs:boolean"
 minOccurs="0"/>
            <xs:element name="ChargingComplete" type="xs:boolean"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!--            -->
<!-- ChargingProfileType -->
<!--           -->
<xs:complexType name="ChargingProfileType">
   <xs:sequence>
      <xs:element name="SAScheduleTupleID" type="SAIDType"/>
```

```
      <xs:element name="ProfileEntry" type="ProfileEntryType"
 maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ProfileEntryType">
    <xs:sequence>
      <xs:element name="ChargingProfileEntryStart" type="xs:unsignedInt"/>
      <xs:element name="ChargingProfileEntryMaxPower" type="PMaxType"/>
    </xs:sequence>
  </xs:complexType>
  <!-- =================== -->
  <!-- Simple types -->
  <!-- =================== -->
  <!-- -->
  <!-- General Types -->
  <!-- -->
  <xs:simpleType name="PMaxType">
    <xs:restriction base="xs:short"/>
  </xs:simpleType>
  <xs:simpleType name="percentValueType">
    <xs:restriction base="xs:byte">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="100"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="faultMsgType">
    <xs:restriction base="xs:string">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="EVSEProcessingType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Finished"/>
      <xs:enumeration value="Ongoing"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="EVSENotificationType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="None"/>
      <xs:enumeration value="StopCharging"/>
      <xs:enumeration value="ReNegotiation"/>
    </xs:restriction>
  </xs:simpleType>
  <!--        -->
  <!-- service related types -->
  <!--        -->
  <xs:simpleType name="serviceNameType">
    <xs:restriction base="xs:string">
      <xs:maxLength value="32"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="serviceCategoryType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="EVCharging"/>
      <xs:enumeration value="Internet"/>
      <xs:enumeration value="ContractCertificate"/>
      <xs:enumeration value="OtherCustom"/>
```

```
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="serviceScopeType">
      <xs:restriction base="xs:string">
         <xs:maxLength value="32"/>
      </xs:restriction>
   </xs:simpleType>
   <!--        -->
   <!-- EnergyTransferType -->
   <!--        -->
   <xs:simpleType name="EVSESupportedEnergyTransferType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="AC_single_phase_core"/>
         <xs:enumeration value="AC_three_phase_core"/>
         <xs:enumeration value="DC_core"/>
         <xs:enumeration value="DC_extended"/>
         <xs:enumeration value="DC_combo_core"/>
         <xs:enumeration value="DC_dual"/>
         <xs:enumeration value="AC_core1p_DC_extended"/>
         <xs:enumeration value="AC_single_DC_core"/>
         <xs:enumeration
    value="AC_single_phase_three_phase_core_DC_extended"/>
         <xs:enumeration value="AC_core3p_DC_extended"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="EVRequestedEnergyTransferType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="AC_single_phase_core"/>
         <xs:enumeration value="AC_three_phase_core"/>
         <xs:enumeration value="DC_core"/>
         <xs:enumeration value="DC_extended"/>
         <xs:enumeration value="DC_combo_core"/>
         <xs:enumeration value="DC_unique"/>
      </xs:restriction>
   </xs:simpleType>
   <!--        -->
   <!-- security types -->
   <!--        -->
   <xs:simpleType name="genChallengeType">
      <xs:restriction base="xs:string"/>
   </xs:simpleType>
   <xs:simpleType name="certificateType">
      <xs:restriction base="xs:base64Binary">
         <xs:maxLength value="1200"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="rootCertificateIDType">
      <xs:restriction base="xs:string">
         <xs:maxLength value="40"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="dHParamsType">
      <xs:restriction base="xs:base64Binary">
         <xs:maxLength value="256"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="privateKeyType">
```

```
      <xs:restriction base="xs:base64Binary">
         <xs:maxLength value="128"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="sigMeterReadingType">
      <xs:restriction base="xs:base64Binary">
         <xs:maxLength value="32"/>
      </xs:restriction>
   </xs:simpleType>
   <!--        -->
   <!-- Identification Numbers -->
   <!--        -->
   <xs:simpleType name="sessionIDType">
      <xs:restriction base="xs:hexBinary">
         <xs:maxLength value="8"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="evccIDType">
      <xs:restriction base="xs:hexBinary">
         <xs:maxLength value="8"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="evseIDType">
      <xs:restriction base="xs:hexBinary">
         <xs:maxLength value="32"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="serviceIDType">
      <xs:restriction base="xs:unsignedShort"/>
   </xs:simpleType>
   <xs:simpleType name="contractIDType">
      <xs:restriction base="xs:string">
         <xs:maxLength value="24"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="meterIDType">
      <xs:restriction base="xs:string">
         <xs:maxLength value="32"/>
      </xs:restriction>
   </xs:simpleType>
   <!--     -->
   <!-- Tariffs and payment -->
   <!--        -->
   <xs:simpleType name="SAIDType">
      <xs:restriction base="xs:short"/>
   </xs:simpleType>
   <xs:simpleType name="tariffDescriptionType">
      <xs:restriction base="xs:string">
         <xs:maxLength value="32"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="costKindType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="relativePricePercentage"/>
         <xs:enumeration value="RenewableGenerationPercentage"/>
         <xs:enumeration value="CarbonDioxideEmission"/>
      </xs:restriction>
```

```
   </xs:simpleType>
   <xs:complexType name="PaymentOptionsType">
      <xs:sequence>
         <xs:element name="PaymentOption" type="paymentOptionType"
 maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="paymentOptionType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="Contract"/>
         <xs:enumeration value="ExternalPayment"/>
      </xs:restriction>
   </xs:simpleType>
   <!--     -->
   <!-- Fault and Response Codes -->
   <!--         -->
   <xs:simpleType name="faultCodeType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="ParsingError"/>
         <xs:enumeration value="NoTLSRootCertificatAvailable"/>
         <xs:enumeration value="UnknownError"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="responseCodeType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="OK"/>
         <xs:enumeration value="OK_NewSessionEstablished"/>
         <xs:enumeration value="OK_OldSessionJoined"/>
         <xs:enumeration value="OK_CertificateExpiresSoon"/>
         <xs:enumeration value="FAILED"/>
         <xs:enumeration value="FAILED_SequenceError"/>
         <xs:enumeration value="FAILED_ServiceIDInvalid"/>
         <xs:enumeration value="FAILED_UnknownSession"/>
         <xs:enumeration value="FAILED_ServiceSelectionInvalid"/>
         <xs:enumeration value="FAILED_PaymentSelectionInvalid"/>
         <xs:enumeration value="FAILED_CertificateExpired"/>
         <xs:enumeration value="FAILED_SignatureError"/>
         <xs:enumeration value="FAILED_NoCertificateAvailable"/>
         <xs:enumeration value="FAILED_CertChainError"/>
         <xs:enumeration value="FAILED_ChallengeInvalid"/>
         <xs:enumeration value="FAILED_ContractCanceled"/>
         <xs:enumeration value="FAILED_WrongChargeParameter"/>
         <xs:enumeration value="FAILED_PowerDeliveryNotApplied"/>
         <xs:enumeration value="FAILED_TariffSelectionInvalid"/>
         <xs:enumeration value="FAILED_ChargingProfileInvalid"/>
         <xs:enumeration value="FAILED_EVSEPresentVoltageToLow"/>
         <xs:enumeration value="FAILED_MeteringSignatureNotValid"/>
         <xs:enumeration value="FAILED_WrongEnergyTransferType"/>
      </xs:restriction>
   </xs:simpleType>
   <!--     -->
   <!-- Multiplier and Unit Types -->
   <!--        -->
   <xs:simpleType name="unitMultiplierType">
      <xs:restriction base="xs:byte">
         <xs:minInclusive value="-3"/>
         <xs:maxInclusive value="3"/>
```

```
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="unitSymbolType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="h">
            <xs:annotation>
               <xs:documentation>Time in hours</xs:documentation>
            </xs:annotation>
         </xs:enumeration>
         <xs:enumeration value="m">
            <xs:annotation>
               <xs:documentation>Time in minutes</xs:documentation>
            </xs:annotation>
         </xs:enumeration>
         <xs:enumeration value="s">
            <xs:annotation>
               <xs:documentation>Time in seconds</xs:documentation>
            </xs:annotation>
         </xs:enumeration>
         <xs:enumeration value="A">
            <xs:annotation>
               <xs:documentation>Current in Ampere</xs:documentation>
            </xs:annotation>
         </xs:enumeration>
         <xs:enumeration value="Ah">
            <xs:annotation>
               <xs:documentation>Ampere hour</xs:documentation>
            </xs:annotation>
         </xs:enumeration>
         <xs:enumeration value="V">
            <xs:annotation>
               <xs:documentation>Voltage in Volt</xs:documentation>
            </xs:annotation>
         </xs:enumeration>
         <xs:enumeration value="VA">
            <xs:annotation>
               <xs:documentation>Apparent power in Volt
 Ampere</xs:documentation>
            </xs:annotation>
         </xs:enumeration>
         <xs:enumeration value="W">
            <xs:annotation>
               <xs:documentation>Active power in Watt</xs:documentation>
            </xs:annotation>
         </xs:enumeration>
         <xs:enumeration value="W/s">
            <xs:annotation>
               <xs:documentation>Watt per second</xs:documentation>
            </xs:annotation>
         </xs:enumeration>
         <xs:enumeration value="Wh">
            <xs:annotation>
               <xs:documentation>Real energy in Watt hours</xs:documentation>
            </xs:annotation>
         </xs:enumeration>
      </xs:restriction>
   </xs:simpleType>
```

```
   <!-- -->
   <!-- only DC related -->
   <!-- -->
   <xs:simpleType name="DC_EVSEStatusCodeType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="EVSE_NotReady"/>
         <xs:enumeration value="EVSE_Ready"/>
         <xs:enumeration value="EVSE_Shutdown"/>
         <xs:enumeration value="EVSE_UtilityInterruptEvent"/>
         <xs:enumeration value="EVSE_IsolationMonitoringActive"/>
         <xs:enumeration value="EVSE_EmergencyShutdown"/>
         <xs:enumeration value="EVSE_Malfunction"/>
         <xs:enumeration value="Reserved_8"/>
         <xs:enumeration value="Reserved_9"/>
         <xs:enumeration value="Reserved_A"/>
         <xs:enumeration value="Reserved_B"/>
         <xs:enumeration value="Reserved_C"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="isolationLevelType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="Invalid"/>
         <xs:enumeration value="Valid"/>
         <xs:enumeration value="Warning"/>
         <xs:enumeration value="Fault"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="DC_EVErrorCodeType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="NO_ERROR"/>
         <xs:enumeration value="FAILED_RESSTemperatureInhibit"/>
         <xs:enumeration value="FAILED_EVShiftPosition"/>
         <xs:enumeration value="FAILED_ChargerConnectorLockFault"/>
         <xs:enumeration value="FAILED_EVRESSMalfunction"/>
         <xs:enumeration value="FAILED_ChargingCurrentdifferential"/>
         <xs:enumeration value="FAILED_ChargingVoltageOutOfRange"/>
         <xs:enumeration value="Reserved_A"/>
         <xs:enumeration value="Reserved_B"/>
         <xs:enumeration value="Reserved_C"/>
         <xs:enumeration value="FAILED_ChargingSystemIncompatibility"/>
         <xs:enumeration value="NoData"/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

A.7 xmldsig-core-schema.xsd

```
<?xml version="1.0" encoding="UTF-8"? >
<!DOCTYPE schema PUBLIC "-//W3C//DTD XMLSchema 200102//EN"
    "http://www.w3.org/2001/XMLSchema.dtd" [
<!ATTLIST schema
xmlns:ds CDATA #FIXED "http://www.w3.org/2000/09/xmldsig#">
<!ENTITY dsig 'http://www.w3.org/2000/09/xmldsig#'>
<!ENTITY % p ''>
<!ENTITY % s ''>
]>
<!-- Schema for XML Signatures
```

```
http://www.w3.org/2000/09/xmldsig#
$Revision: 1.1 $ on $Date: 2002/02/08 20:32:26 $ by $Author: reagle $
Copyright 2001 The Internet Society and W3C (Massachusetts Institute
of Technology, Institut National de Recherche en Informatique et en
Automatique, Keio University). All Rights Reserved.
http://www.w3.org/Consortium/Legal/
This document is governed by the W3C Software License [1] as described
in the FAQ [2].
[1] http://www.w3.org/Consortium/Legal/copyright-software-19980720
[2] http://www.w3.org/Consortium/Legal/IPR-FAQ-20000620.html#DTD
-->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    targetNamespace="http://www.w3.org/2000/09/xmldsig#" version="0.1"
    elementFormDefault="qualified">
<!-- Basic Types Defined for Signatures -->
<simpleType name="CryptoBinary">
<restriction base="base64Binary">
</restriction>
</simpleType>
<!-- Start Signature -->
<element name="Signature" type="ds:SignatureType"/>
<complexType name="SignatureType">
<sequence>
<element ref="ds:SignedInfo"/>
<element ref="ds:SignatureValue"/>
<element ref="ds:KeyInfo" minOccurs="0"/>
<element ref="ds:Object" minOccurs="0" maxOccurs="unbounded"/>
</sequence>
<attribute name="Id" type="ID" use="optional"/>
</complexType>
<element name="SignatureValue" type="ds:SignatureValueType"/>
<complexType name="SignatureValueType">
<simpleContent>
<extension base="base64Binary">
<attribute name="Id" type="ID" use="optional"/>
</extension>
</simpleContent>
</complexType>
<!-- Start SignedInfo -->
<element name="SignedInfo" type="ds:SignedInfoType"/>
<complexType name="SignedInfoType">
<sequence>
<element ref="ds:CanonicalizationMethod"/>
<element ref="ds:SignatureMethod"/>
<element ref="ds:Reference" maxOccurs="unbounded"/>
</sequence>
<attribute name="Id" type="ID" use="optional"/>
</complexType>
<element name="CanonicalizationMethod" type="ds:CanonicalizationMethodType"/>
<complexType name="CanonicalizationMethodType" mixed="true">
<sequence>
<any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
<!-- (0,unbounded) elements from (1,1) namespace -->
</sequence>
<attribute name="Algorithm" type="anyURI" use="required"/>
</complexType>
```

```
<element name="SignatureMethod" type="ds:SignatureMethodType"/>
<complexType name="SignatureMethodType" mixed="true">
<sequence>
<element name="HMACOutputLength" minOccurs="0"
    type="ds:HMACOutputLengthType"/>
<any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
<!-- (0,unbounded) elements from (1,1) external namespace -->
</sequence>
<attribute name="Algorithm" type="anyURI" use="required"/>
</complexType>
<!-- Start Reference -->
<element name="Reference" type="ds:ReferenceType"/>
<complexType name="ReferenceType">
<sequence>
<element ref="ds:Transforms" minOccurs="0"/>
<element ref="ds:DigestMethod"/>
<element ref="ds:DigestValue"/>
</sequence>
<attribute name="Id" type="ID" use="optional"/>
<attribute name="URI" type="anyURI" use="optional"/>
<attribute name="Type" type="anyURI" use="optional"/>
</complexType>
<element name="Transforms" type="ds:TransformsType"/>
<complexType name="TransformsType">
<sequence>
<element ref="ds:Transform" maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="Transform" type="ds:TransformType"/>
<complexType name="TransformType" mixed="true">
<choice minOccurs="0" maxOccurs="unbounded">
<any namespace="##other" processContents="lax"/>
<!-- (1,1) elements from (0,unbounded) namespaces -->
<element name="XPath" type="string"/>
</choice>
<attribute name="Algorithm" type="anyURI" use="required"/>
</complexType>
<!-- End Reference -->
<element name="DigestMethod" type="ds:DigestMethodType"/>
<complexType name="DigestMethodType" mixed="true">
<sequence>
<any namespace="##other" processContents="lax" minOccurs="0"
    maxOccurs="unbounded"/>
</sequence>
<attribute name="Algorithm" type="anyURI" use="required"/>
</complexType>
<element name="DigestValue" type="ds:DigestValueType"/>
<simpleType name="DigestValueType">
<restriction base="base64Binary"/>
</simpleType>
<!-- End SignedInfo -->
<!-- Start KeyInfo -->
<element name="KeyInfo" type="ds:KeyInfoType"/>
<complexType name="KeyInfoType" mixed="true">
<choice maxOccurs="unbounded">
<element ref="ds:KeyName"/>
<element ref="ds:KeyValue"/>
```

```
<element ref="ds:RetrievalMethod"/>
<element ref="ds:X509Data"/>
<element ref="ds:PGPData"/>
<element ref="ds:SPKIData"/>
<element ref="ds:MgmtData"/>
<any processContents="lax" namespace="##other"/>
<!-- (1,1) elements from (0,unbounded) namespaces -->
</choice>
<attribute name="Id" type="ID" use="optional"/>
</complexType>
<element name="KeyName" type="string"/>
<element name="MgmtData" type="string"/>
<element name="KeyValue" type="ds:KeyValueType"/>
<complexType name="KeyValueType" mixed="true">
<choice>
<element ref="ds:DSAKeyValue"/>
<element ref="ds:RSAKeyValue"/>
<any namespace="##other" processContents="lax"/>
</choice>
</complexType>
<element name="RetrievalMethod" type="ds:RetrievalMethodType"/>
<complexType name="RetrievalMethodType">
<sequence>
<element ref="ds:Transforms" minOccurs="0"/>
</sequence>
<attribute name="URI" type="anyURI"/>
<attribute name="Type" type="anyURI" use="optional"/>
</complexType>
<!-- Start X509Data -->
<element name="X509Data" type="ds:X509DataType"/>
<complexType name="X509DataType">
<sequence maxOccurs="unbounded">
<choice>
<element name="X509IssuerSerial" type="ds:X509IssuerSerialType"/>
<element name="X509SKI" type="base64Binary"/>
<element name="X509SubjectName" type="string"/>
<element name="X509Certificate" type="base64Binary"/>
<element name="X509CRL" type="base64Binary"/>
<any namespace="##other" processContents="lax"/>
</choice>
</sequence>
</complexType>
<complexType name="X509IssuerSerialType">
<sequence>
<element name="X509IssuerName" type="string"/>
<element name="X509SerialNumber" type="integer"/>
</sequence>
</complexType>
<!-- End X509Data -->
<!-- Begin PGPData -->
<element name="PGPData" type="ds:PGPDataType"/>
<complexType name="PGPDataType">
<choice>
<sequence>
<element name="PGPKeyID" type="base64Binary"/>
<element name="PGPKeyPacket" type="base64Binary" minOccurs="0"/>
```

```
<any namespace="##other" processContents="lax" minOccurs="0"
    maxOccurs="unbounded"/>
</sequence>
<sequence>
<element name="PGPKeyPacket" type="base64Binary"/>
<any namespace="##other" processContents="lax" minOccurs="0"
    maxOccurs="unbounded"/>
</sequence>
</choice>
</complexType>
<!-- End PGPData -->
<!-- Begin SPKIData -->
<element name="SPKIData" type="ds:SPKIDataType"/>
<complexType name="SPKIDataType">
<sequence maxOccurs="unbounded">
<element name="SPKISexp" type="base64Binary"/>
<any namespace="##other" processContents="lax" minOccurs="0"/>
</sequence>
</complexType>
<!-- End SPKIData -->
<!-- End KeyInfo -->
<!-- Start Object (Manifest, SignatureProperty) -->
<element name="Object" type="ds:ObjectType"/>
<complexType name="ObjectType" mixed="true">
<sequence minOccurs="0" maxOccurs="unbounded">
<any namespace="##any" processContents="lax"/>
</sequence>
<attribute name="Id" type="ID" use="optional"/>
<attribute name="MimeType" type="string" use="optional"/>
<attribute name="Encoding" type="anyURI" use="optional"/>
</complexType>
<element name="Manifest" type="ds:ManifestType"/>
<complexType name="ManifestType">
<sequence>
<element ref="ds:Reference" maxOccurs="unbounded"/>
</sequence>
<attribute name="Id" type="ID" use="optional"/>
</complexType>
<element name="SignatureProperties" type="ds:SignaturePropertiesType"/>
<complexType name="SignaturePropertiesType">
<sequence>
<element ref="ds:SignatureProperty" maxOccurs="unbounded"/>
</sequence>
<attribute name="Id" type="ID" use="optional"/>
</complexType>
<element name="SignatureProperty" type="ds:SignaturePropertyType"/>
<complexType name="SignaturePropertyType" mixed="true">
<choice maxOccurs="unbounded">
<any namespace="##other" processContents="lax"/>
<!-- (1,1) elements from (1,unbounded) namespaces -->
</choice>
<attribute name="Target" type="anyURI" use="required"/>
<attribute name="Id" type="ID" use="optional"/>
</complexType>
<!-- End Object (Manifest, SignatureProperty) -->
<!-- Start Algorithm Parameters -->
<simpleType name="HMACOutputLengthType">
```

```
<restriction base="integer"/>
</simpleType>
<!-- Start KeyValue Element-types -->
<element name="DSAKeyValue" type="ds:DSAKeyValueType"/>
<complexType name="DSAKeyValueType">
<sequence>
<sequence minOccurs="0">
<element name="P" type="ds:CryptoBinary"/>
<element name="Q" type="ds:CryptoBinary"/>
</sequence>
<element name="G" type="ds:CryptoBinary" minOccurs="0"/>
<element name="Y" type="ds:CryptoBinary"/>
<element name="J" type="ds:CryptoBinary" minOccurs="0"/>
<sequence minOccurs="0">
<element name="Seed" type="ds:CryptoBinary"/>
<element name="PgenCounter" type="ds:CryptoBinary"/>
</sequence>
</sequence>
</complexType>
<element name="RSAKeyValue" type="ds:RSAKeyValueType"/>
<complexType name="RSAKeyValueType">
<sequence>
<element name="Modulus" type="ds:CryptoBinary"/>
<element name="Exponent" type="ds:CryptoBinary"/>
</sequence>
</complexType>
<!-- End KeyValue Element-types -->
<!-- End Signature -->
</schema>
```

Annex B
(informative)

Difference between multi-network broadcast, broadcast and unicast

Due to the increased number of questions regarding the terms “broadcast” and “unicast” used in this document and “multi-network broadcast” used in the HomePlug Green PHY Spec, this Annex will try to clarify the difference.

Multi-Network Broadcast (MNBC) is a term defined in the HomePlug Green PHY Spec. It is a transmission mechanism that enables exchange of management messages between stations that are not part of the same logical network (AVLN). The stations may be part of different AVLNs or not associated to an AVLN at all.

Since the purpose of SLAC is to make connected EV and EVSE associate (build a logical network), it makes use of MNBC because in the initial state the EV and EVSE have different NMK / NID pairs and are not part of the same AVLN. MNBC enables EV SLAC messages to be received at the EVSE and vice versa.

An application running in the HLE and sending MMEs cannot choose whether the MMEs shall be sent as MNBC. This decision is made automatically by the HomePlug Green PHY chip.

Broadcast and unicast are terms that describe the destination address in the Ethernet header of an MME. This is the layer above HomePlug Green PHY, so the terms are in no relation to multi-network broadcast.

If a unicast MME is sent as an MNBC, it will be received by all stations in range (even if not part of the same AVLN) but the MAC filter in the HomePlug Green PHY chip will not forward it to the HLE if this unicast address does not match the address of the recipient station. If the Ethernet destination address is broadcast, then the message will be forwarded to the HLE.

On the other hand, an MME that is not part of the SLAC process will not be transmitted as an MNBC by the HomePlug Green PHY chip and will only be received by stations part of the same AVLN. It will then be forwarded to their HLEs if it has a broadcast Ethernet destination address or if it is sent as unicast and the address matches the address of the recipient.

Thus, looking at the HomePlug Green PHY Spec and this document, one should differentiate between MNBC and broadcast / Unicast. All SLAC messages are sent as MNBC by the HomePlug Green PHY chip and the user cannot affect this. The user can only choose what the destination address of the Ethernet header will be. Generally, messages from EV to EVSE shall be sent as broadcast, except CM_ATTEN_CHAR.RSP, CM_VALIDATE.REQ and CM_SLAC_MATCH.REQ. Messages from EVSE to EV shall be sent as unicast. CM_AMP_MAP.REQ/CNF and CM_SET_KEY.REQ shall also be sent as unicast.

Annex C
(informative)

SLAC attenuation calculation example

Figure C.1 gives an example of the SLAC process with details on attenuation calculation:

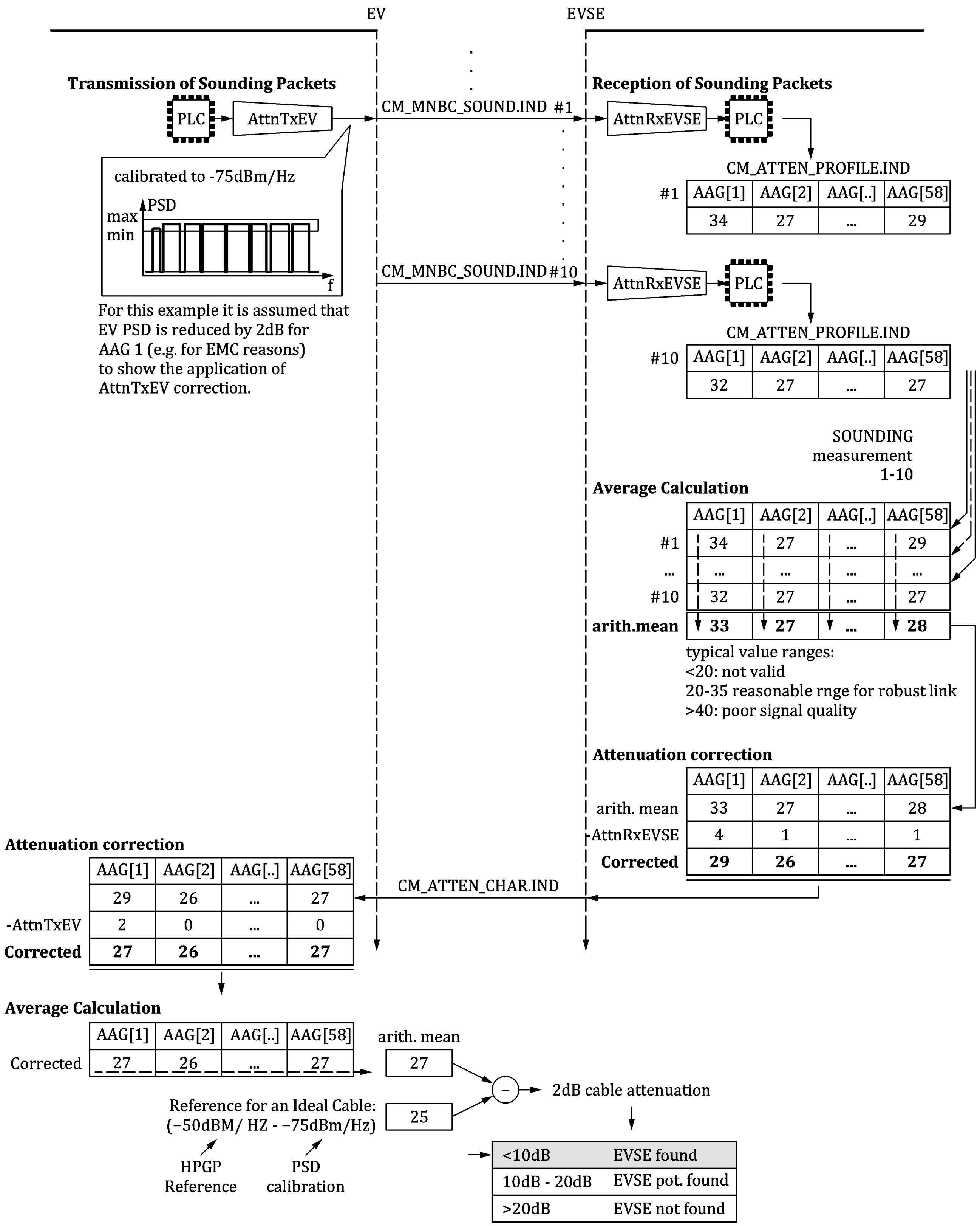


Figure C.1 — SLAC attenuation calculation example

Annex D
(informative)

Begriffe

Reihenfolge und Inhalt der folgenden Begriffe sind identisch mit denen im Abschnitt Begriffe der Englischen Fassung.

3.1
Amplitudenabbild
enamplitude map
Faktor zur Reduzierung der Übertragungsleistung für jeden Zwischenträger bezogen auf das Tonabbild

3.2
zentraler Koordinator
CCo
Betreiber eines „HomePlug Green PHY“-Netzwerkes

3.3
Kanal-Zugriffspriorisierung
CAP
Methode zur Priorisierung des Kanalzugriffes (siehe „HomePlug Green PHY“-Spezifikation)

3.4
Ladedauer
Zeit während des Beginns (Anschluss des Kabels) und dem Ende (Trennen des Kabels) eines Ladeprozesses

3.5
Koexistenz
Fähigkeit verschiedener „HomePlug Green PHY“-Systeme dasselbe physikalische Medium zu verwenden und simultan zu funktionieren

3.6
Kommunikationsmedium
physikalisches Medium, welches das „HomePlug Green PHY“-Signal überträgt, dargestellt durch die Verkabelung, welche die Ladeinfrastruktur mit dem Elektrofahrzeug verbindet

3.7
Kommunikationskonfigurationszeitgeber
Zeitgeber, der die Zeit vom Einstecken bis zum Empfang der SessionSetupRes-Botschaft vom Elektrofahrzeug überwacht

3.8
Verbindungskoordinierung
Einheit, welche die vollständige Funktionalität zum Abgleich und zur Initialisierung zwischen Elektrofahrzeug und Ladeeinrichtung über die Datenverbindungskontrolle SAP Dienstzugangspunkt zur Datenübertragungssteuerung bereitstellt

Anmerkung 1 zum Begriff: Diese Einheit steuert außerdem die Beziehung zwischen der Basissignalisierung und den oberen Schichten.

Anmerkung 2 zum Begriff: Die Einheit liefert den Verbindungsstatus und Fehlerinformationen an die oberen Schichten. Die Steuerung der Netzwerkmanagement-Parameter zur Kommunikation auf den unteren Schichten wird über den Dienstzugangspunkt zur Datenübertragungssteuerung-SAP gehandhabt.

3.9
übersprechen
kapazitive oder induktive Kopplung zwischen zwei unabhängigen elektrischen Schaltkreisen, die beide die Basis für ein HomePlug Green-PHY-Netzwerk sind, in der Art und Weise, bei der sich beide Netzwerke untereinander beeinflussen

3.10
Daten-Dienstzugangspunkt
Dienstzugangspunkt, der die Schnittstelle zwischen den Kommunikationsschichten 2 und 3 zum Austausch relevanter Informationsdaten zwischen Elektrofahrzeug und Ladeeinrichtung festlegt

3.11
Kommunikationssteuerung Elektrofahrzeug
EVCC
Entität, welche die Kommunikation zu einer oder mehreren Ladeeinrichtung-Kommunikationssteuerung(en) durchführt

3.12
Ethernet-Dienstzugangspunkt
geschichteter Dienstzugangspunkt, der Anwendungen für Klasse II-Ethernet Pakete, inklusive IEEE 802.3, mit oder ohne IEEE 802.2 (LLC), IEEE 802.1 (SNAP) Erweiterungen, und/oder VLAN-Identifizierung unterstützt

3.13
globale Adresse
IP-Adresse mit unbegrenztem Anwendungsbereich

3.14
HomePlug-Green-PHY-Knoten
Gerät, welches mit einem HomePlug-Green-PHY-V1.1-kompatiblen Modemchip ausgestattet ist

Anmerkung 1 zum Begriff: Es charakterisiert ein logisches und physikalisches Kommunikationsgerät, das mit einem physikalischen Medium verbunden ist und in der Lage ist, Informationen über einen HomePlug-Green-PHY-Kanal zu senden, zu empfangen oder weiterzuleiten.

3.15
übergeordnete Kommunikation
HLC
digitale Kommunikation, die für den Austausch von Ladeparametern verwendet wird

BEISPIEL Spannung und Strom.

Anmerkung 1 zum Begriff: HLC beginnt mit der CM_SLAC_PARM.REQ-Anforderung und endet, wenn die TCP-Verbindung geschlossen wird.

3.16
Initialisierung
Austauschprozess zwischen dem Elektrofahrzeug, der Ladeeinrichtung und einem externen Auslöser, beginnend bei der Steckvorrichtung des Ladekabels bis hin zur Entscheidung für den anzuwendenden Lademodus

3.17
Intersystem Protokoll
ISP
Protokoll, welches verschiedenen Breitband Stromleitungssystemen Ressourcen für Stromleitungskommunikation in Zeit (Multiplexing im Zeitbereich), in Frequenz (Multiplexing im Frequenzbereich), oder beides, aufzuteilen

Anmerkung 1 zum Begriff: Für weitere Informationen siehe IEEE 1901.

3.18
IP-Adresse
Bezeichner der IP-Schicht für eine Schnittstelle oder eine Menge von Schnittstellen

3.19
verbindungslokale Adresse
IP-Adresse mit Anwendungsbereich von lokalen Verbindungen, der dazu verwendet werden kann, um benachbarte Knoten auf derselben Verbindung zu erreichen

3.20
logisches Netzwerk
Netzwerk, welches ausschließlich auf OSI-Schicht 2 festgelegt ist

Anmerkung 1 zum Begriff: Dies ist eine Menge von Green-PHY-Stationen, welche denselben Netzwerkschlüssel besitzen.

Anmerkung 2 zum Begriff: Nur Mitglieder desselben logischen Netzwerks können verschlüsselte Nutzdaten austauschen und sind auf höheren Schichten untereinander sichtbar. Verschiedene logische Netzwerke können zur selben Zeit auf demselben physikalischen Medium bestehen und werden typischerweise zur Netzwerksegmentierung verwendet

3.21
MAC-Adresse
eindeutiger Bezeichner, der für Netzwerkschnittstellen zur Kommunikation auf der Sicherungsschicht vergeben wird

3.22
Abgleich
Prozess zur Bestimmung der typischerweise, über ein Ladekabel, direkt verbundenen HomePLug Green-PHY-Knoten des Elektrofahrzeugs und des Green-PHY-Knoten der Ladeeinrichtung

Anmerkung 1 zum Begriff: Der Abgleichprozess startet mit „CM_SLAC_PARAM.REQ“ und endet mit der Bestätigung „CM_SLAC_MATCH.CNF“.

3.23
Botschaftsmenge
Menge von notwendigen V2G-Botschaften und –Parametern für die Elektrofahrzeug- Kommunikationssteuerung oder Ladeeinrichtung-Kommunikationssteuerung, die einen oder mehrere Anwendungsfall-Elemente abdeckt

3.24
Botschaftszeitgeber
Zeitgeber, der den Austausch eines Paares von Anfrage/Antwort-Botschaften überwacht

3.25
Netzwerkabschnitt
Menge von Geräten, die Daten direkt auf Sicherungsschicht-Ebene über Sicherungsschicht-Adressen austauschen

BEISPIEL Ethernet: Alle Geräte, die sich über MAC-Adressen austauschen.

3.26
Netzwerkabschnittgröße
Menge von Geräten, die in einem Netzwerk-Abschnitt gehandhabt werden

3.27
Knoten
Gerät, das Teil eines Kommunikationsnetzwerkes ist

BEISPIEL Ein Gerät, welches IPv6 in einem IP-Netzwerk implementiert hat.

3.28
Leistungszeit
nicht-funktionale Zeitgeberanforderung, die eine Zeit festlegt, die von einer V2G-Einheit nicht überschritten werden sollte, wenn diese eine bestimmte Funktionalität aus- oder durchführt

3.29
PHY
verbindet ein Sicherungsschicht Gerät mit einem physischen Medium

Anmerkung 1 zum Begriff: Schicht 1 im ISO/OSI-Referenzmodell.

3.30
Pilotfunktionssteuerung
System, welches die Pilotfunktionsverbindung auf EVSE-Seite steuert

Anmerkung 1 zum Begriff: Nach IEC 61851-1.

3.31
Profil
Gruppe verbindlicher und optionaler Botschaftsmengen, welche eine Menge ähnlicher Ladeszenarien für ein spezifisches Identifikationsmittel abdecken

3.32
QPSK-Modulation
Phasenmodulationstechnologie, die zwei Bits in vier Modulationsstati überträgt (Quadraturphasenumtastung)

3.33
Ladebereitschaft-Zeitgeber
Zeitgeber, der die Zeit vom Einstecken bis zur ersten Leistungslieferungsbotschaft überwacht

3.34
Anfrage-Antwort-Botschaftspaar
Anfrage-Botschaft, die zwischen Sender und Empfänger übertragen wird und dessen korrespondierende Antwort-Botschaft, welche durch die Anfrage-Botschaft im Empfänger ausgelöst wurde

3.35
Anfrage-Antwort-Botschaftssequenz
Sequenz von einem oder mehreren Anfrage-Antwort-Botschaftspaar(en)

3.36
ROBO-Modus
Kommunikationsmodus, welcher QPSK nur für die Trägermodulation innerhalb des orthogonalen Frequenzmultiplexverfahren (OFDM) verwendet, um höhere Robustheit in der Übertragung zu erreichen

Anmerkung 1 zum Begriff: Der ROBO-Modus kann auf drei unterschiedliche Leistungsniveaus gesetzt werden. Mini-ROBO, Standard-ROBO und High-Speed-ROBO.

Anmerkung 2 zum Begriff: Siehe für weitere Details die HomePlug Green-PHY-Spezifikation.

3.37
SDP-Klient
V2G-Entiät, die den SDP-Server benutzt, um Konfigurationsinformationen über den SECC zu erhalten, damit auf den SECC zugegriffen werden kann

3.38
SDP-Server
V2G-Entiät, die Konfigurationsinformationen für den SECC-Zugriff bereitstellt

3.39
Sequenzzeitgeber
Zeitgeber, der die Antwort-Anfrage-Botschaftssequenz überwacht

3.40
geteilte Bandbreite
Netzwerkfähigkeit, die von mehreren Nutzern oder Geräten zur Übertragung von Daten verwendet wird

Anmerkung 1 zum Begriff: Die resultierende Zuordnung ist geteilte Bandbreite.

3.41
Signalkopplung
Schaltung, welche die Methode zur Kopplung des HomePlug Green-PHY-Signals mit dem Kommunikationsmedium beschreibt

3.42
Signalniveau Abschwächungscharakteristik
SLAC
Protokoll zur Messung der Signalstärke des Signals zwischen HomePlug Green-PHY-Konten (wird zur Abstimmung verwendet)

3.43
Signalstärkemessung
Messung zwischen den Botschaften CM_START_ATTEN_CHAR.IND und CM_ATTEN_CHAR.RSP

3.44
Ladeeinrichtung Kommunikationssteuerung
SECC
Entität, welche die Kommunikation zu einer oder mehreren EVCC(s) durchführt, und die befähigt sein könnte, mit zweitrangigen Akteuren zu interagieren

3.45
TCP_DATA
Socket/Schnittstelle für die Datenübertragung, basierend auf einer TCP-Verbindung

3.46
Zeitüberschreitung
spezifische Zeit, während der eine V2G-Entität das Kommunikationssystem überwacht, um ein bestimmtes auftretendes Ereignis zu entdecken

Anmerkung 1 zum Begriff: Wenn die festgelegte Zeit abgelaufen ist, initiiert die betroffene V2G-Entität die notwendige Fehlerbehebung.

3.47
Zeitgeber
Gerät oder SW-Komponente, die in einer Implantierung dazu verwendet wird, um Zeit zu messen

Anmerkung 1 zum Begriff: Abhängig vom Anwendungsfall wird der Zeitgeber dafür verwendet, um bestimmte Systemereignisse zusätzlich auszulösen.

3.48
Tonmaskierung
Menge von OFDM-Trägern oder „Tönen", welche in einer gegebenen behördlichen Zuständigkeit oder Anwendung verwendet wird

3.49
V2G-Kommunikationssitzung
halbdauernder, interaktiver Informationsaustausch zwischen zwei spezifischen V2G-Entitäten für den Austausch von V2G-Botschaften in einer verbindungsorientierten Kommunikation

3.50
V2G-Entität
primärer oder zweitrangiger Akteur, der an einer V2G-Kommunikation teilnimmt

3.51
V2G-Botschaft
Botschaft, die auf Anwendungsebene ausgetauscht wird

3.52
V2GTP-Entität
V2G-Entität, die das V2G-Übertragungsprotokoll unterstützt

Annex E
(informative)

Modified requirements

The following Table E.1 shows the requirements that have been modified.

Table E.1 — Modified requirements

Old Number	New Number
V2G-DC-736	V2G-DC-800
V2G-DC-053	V2G-DC-801
V2G-DC-578	V2G-DC-802
V2G-DC-579	V2G-DC-803
V2G-DC-564	V2G-DC-807
V2G-DC-024	V2G-DC-821
V2G-DC-035	V2G-DC-819
V2G-DC-716	V2G-DC-823
V2G-DC-701	V2G-DC-824
V2G-DC-718	V2G-DC-825 and V2G-DC-826
V2G-DC-216	V2G-DC-849
V2G-DC-501	V2G-DC-880
V2G-DC-685	V2G-DC-806
V2G-DC-693	V2G-DC-822
V2G-DC-729	V2G-DC-828
V2G-DC-615	V2G-DC-943
V2G-DC-587	V2G-DC-960
V2G-DC-175	V2G-DC-963
V2G-DC-453	V2G-DC-965
V2G-DC-498	V2G-DC-966
V2G-DC-547	V2G-DC-967
V2G-DC-672	V2G-DC-968
V2G-DC-361	V2G-DC-975
V2G-DC-371	V2G-DC-976
V2G-DC-375	V2G-DC-977
V2G-DC-380	V2G-DC-978
V2G-DC-385	V2G-DC-979
V2G-DC-651	V2G-DC-980
V2G-DC-652	V2G-DC-981
V2G-DC-653	V2G-DC-982
V2G-DC-402	V2G-DC-983

Table E.1 (*continued*)

Old Number	New Number
V2G-DC-656	V2G-DC-984
V2G-DC-366	V2G-DC-985
V2G-DC-665	V2G-DC-986
V2G-DC-666	V2G-DC-987
V2G-DC-556	V2G-DC-988
V2G-DC-654	V2G-DC-990

Bibliography

[1] ISO 15118-2, *Road vehicles — Vehicle-to-grid communication interface — Part 2: Network and application protocol requirements*

[2] ISO/IEC 10731, Information technology — Open Systems Interconnection — Basic Reference Model — Conventions for the definition of OSI services

[3] IETF RFC 5871, IANA Allocation Guidelines for the IPv6 Routing Header (Mai 2010)

[4] IETF RFC 5220, Problem Statement for Default Address Selection in Multi-Prefix Environments: Operational Issues of RFC 3484 Default Rules (July 2008)

[5] Altova XMLSpy Manual [viewed 2011-01-12], Available from <http://manual.altova.com/XMLSpy/spyprofessional/index.html?xseditingviews_schview_contmodview.htm>

[6] IETF RFC 5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (May 2008)

[7] Object Identifier (OID) Repository [viewed 2011-01-12], Available from <http://www.oid-info.com/>

[8] W3C EXI Profile, Efficient XML Interchange (EXI) (August 2011)

[9] W3C XML, Extensible Markup Language (XML) 1.0 (Fifth Edition) (November 2008)

[10] W3C XMLSchema 0, XML Schema Part 0: Primer Second Edition (October 2004)

[11] W3C XMLSchema 1, XML Schema Part 1: Structures Second Edition (October 2004)

[12] W3C XMLSchema 2, XML Schema Part 2: Datatypes Second Edition (October 2004)

[13] W3C XMLSig, XML Signature Syntax and Processing (Second Edition) (June 2008)

[14] W3C XMLEnc, XML Encryption Syntax and Processing Version 1.1 (January 2012)