|  |  |
| --- | --- |
| Project | **IEEE 802.16 Broadband Wireless Access Working Group <**<http://ieee802.org/16>**>** |
| Title | **Data model (draft)** |
| Date Submitted | **2015-03-04** |
| Source(s) | Antonio Bovo | Voice: +393771821211E-mail: antonio1234.pd@gmail.com |
| Re: | IEEE 802.16-15-0006-00-03R0 |
| Abstract | Data model (draft) |
| Purpose | Review |
| Notice | *This document does not represent the agreed views of the IEEE 802.16 Working Group or any of its subgroups*. It represents only the views of the participants listed in the “Source(s)” field above. It is offered as a basis for discussion. It is not binding on the contributor(s), who reserve(s) the right to add, amend or withdraw material contained herein. |
| Copyright Policy | The contributor is familiar with the IEEE-SA Copyright Policy <http://standards.ieee.org/IPR/copyrightpolicy.html>. |
| Patent Policy | The contributor is familiar with the IEEE-SA Patent Policy and Procedures:<<http://standards.ieee.org/guides/bylaws/sect6-7.html#6>> and <<http://standards.ieee.org/guides/opman/sect6.html#6.3>>.Further information is located at <<http://standards.ieee.org/board/pat/pat-material.html>> and <<http://standards.ieee.org/board/pat>>. |

Data model (draft)

Antonio Bovo

# 1.1.Scope

This note is a proposal for the procedures and data model to be standardized into the IEEE 802.16.3 measurements framework.

# 1.1. Procedures

## 1.1. Capabilities Exchange

This procedure allows the UE to communicate to the CONTROLLER about its ability to support specific test suites and protocols.

As also explained in DCN\_16\_14\_0099\_00\_03R0 document, the UE is sending a CAPABILITIES EXCHANGE REQUEST to the CONTROLLER and starts a guard timer. The value of this guard timer has to be configurable, with a default value defined in table 1.

***UE behavior***: in case of CAPABILITIES EXCHANGE REQUEST retry and collision with the CONTROLLER response, the UE is considering the procedure as successful and change the state.

***CONTROLLER behavior***: Once received the 1st request, the CONTROLLER is processing the message building a response including the negotiated values for protocol capabilities and test suites. If the CONTROLLER is receiving a further request message after sending the response, it will assume a collision or a response message lost and then it will resend the response. If the UE had already received the 1st response, it will ignore this 2nd message instance.

About collision handling, the behavior is the same for all the request/response procedures with retry.

Note the presence of the Device ID and Permanent Client ID into the request message. These identifiers are privacy-sensitive and so it is mandatory that the dialogue is secured, for example via TLS methodology. The presence of these identifiers allows checking that both the client and the device support the capabilities indicated.

The test suites have to be categorized in advance, including a well-specified set of measurements. It is possible even customizing a test suite with the possibility to configure a specific measurement to be performed.

The transaction identifier IE has a value assigned by the sender and mirrored by the receiver. Different instances of the same procedure have different transaction identifier values, in order to discriminate each procedure instance.

This is valid for all the 2-handshake procedures with a Request and a Response messages (e.g. Registration, Get Reg. Param., etc…)

## 1.1. Registration

Registration procedure allows the UE to register itself to the CONTROLLER and retrieve the dynamic parameters for the session (e.g. the temporary identifiers assigned to the UE and to the test session) and a session lifetime. As soon as this session lifetime expires, the session is considered concluded at both sides (UE and CONTROLLER) without the need of explicit signaling.

Similarly as before, UE starts a Tresp2 guard timer after sending the request message, to realign in case of missing response from the CONTROLLER. In Table 1 there are the default values and the possible range. This timer is stopped once REGISTRATION REQUEST is received by the UE.

It is considered implementation-dependent the possibility to retry with another REGISTRATION procedure in case Tresp2 expires. However, after two retries it is necessary to wait for a specific time interval (Tretry) before reattempting, to avoid congestion at the CONTROLLER side.

***UE behavior***: in case of REGISTRATION REQUEST retry and collision with the REGISTRATION RESPONSE from the CONTROLLER, the UE is considering the procedure as successful and change the state.

***CONTROLLER behavior***: once received the 1st request, the CONTROLLER is processing the message, building a REGISTRATION RESPONSE message with the Result value, the new assigned temporary T-ID and S-ID identifiers, for the client and for the session respectively.

If the CONTROLLER for any reason cannot accept the registration, it will populate the Result IE with proper unsuccessful value.

If the CONTROLLER is receiving a further request message after sending the response, it will assume a collision or a response message lost and then it will resend the response message, with the same temporary identifiers values just assigned. Then it will start a guard timer Tguard1 to check that there will be some activity (e.g. configuration request) after the registration. If Tguard1 expires then the CONTROLLER will assume that the UE has not proceeded with the registration at its side and return to idle. This timer will be stopped once this activity is received.

About the user credentials, the Client is including them into the REGISTRATION REQUEST message. These credentials have been negotiated with the CONTROLLER with a separate mechanism, that can be for example a captive portal mechanism, and this is considered out of scope in the present document.

If re-registration interval parameter is set, the UE will re-register after that time interval, provided that there is no active measurement process ongoing, otherwise the re-registration is delayed.

Note that the following procedures, GET REG PARAM and SET REG PARAM can occur in between the REGISTRATION REQUEST and REGISTRATION RESPONSE messages.

## 1.1. Get Reg Param

This optional procedure can be used in case the CONTROLLER needs to retrieve the path and current value of additional registration parameters in the Client. This allows the CONTROLLER to change them with the next procedure SET REG PARAM. This mechanism can give additional flexibility in case for some reasons it is necessary to check and set additional information present in the Client before accepting the registration.

***CONTROLLER behavior***: the CONTROLLER is sending the request and waiting for the response, including the logical name of the additional parameters to check. There are no guard timers because it is considered enough the timer started for the registration itself.

***UE behavior***: once the UE is receiving the GET REG PARAM REQUEST it will send the GET REG PARAM RESPONSE with the list of requested registration parameters, with their paths and values.

In case the CONTROLLER is not receiving any response it won’t send the REGISTRATION RESPONSE and so the registration timer will expire.

## 1.1. Set Reg Param

This optional procedure is used together with the previous Get Reg Param. Using this procedure the CONTROLLER can set specific additional registration parameters within the Client.

It has retrieved path and name of these parameters via the previous Get Reg Param procedure.

***CONTROLLER behavior***: the CONTROLLER is sending the request and waiting for the response, including the logical name of the additional parameters to check. There are no guard timers because it is considered enough the timer started for the registration itself.

***UE behavior***: once the UE is receiving the SET REG PARAM REQUEST it will update the registration parameters accordingly and send the SET REG PARAM RESPONSE with the “Result” outcome.

In case the CONTROLLER is not receiving any response it won’t send the REGISTRATION RESPONSE and so the registration timer will expire.

## 1.1. Configuration

This procedure can be initiated by the Client to get the configuration for a test session.

***UE behavior***: the UE is sending the CONFIGURATION REQUEST message and waiting for the response, including the temporary identifiers assigned to the Client and to the session respectively.

It is also triggered a guard timer Tresp3 that will be stopped once the CONFIGURATION RESPONSE is received.

Note that the CONFIGURATION REQUEST message is including the chosen test suite among those negotiated at the beginning during the CAPABILITIES EXCHANGE procedure.

***CONTROLLER behavior***: as soon as the CONFIGURATION REQUEST is received, CONTROLLER is processing the message identifying that a configuration process is initiated. Then it can trigger the retrieval of the configuration parameters available in the Client (optionally) and set the configuration parameters according to the tests to be performed. Once these procedures are completed (GET CONF PARAM and SET CONF PARAM) the CONTROLLER is sending back the CONFIGURATION RESPONSE message with the proper Result value.

In case of timer expiry before the entire procedure is completed, both the endpoints assume that the configuration process failed and will retry a second time. If even the 2nd tentative fails, the Client is waiting a predefined time (configurable) Tretry before further attempts in order to avoid Controller congestion.

## 1.1. Get Conf Param

This procedure (optional) can be initiated by the CONTROLLER to get the names, path and values for configuration parameters available in the Client. In this way CONTROLLER can configure different type of CLIENTs that can have multiple set of configuration parameters.

***CONTROLLER behavior***: the Controller is sending the GET CONF PARAM REQUEST to retrieve the parameters names, path and values from the Client. It is including the logical name of such parameters. Timers are not necessary as there is the configuration guard timer that can control even this encapsulated procedure.

***UE behavior***: the UE is sending back the GET CONF PARAM RESPONSE message, including the name, path and configured values for the requested logical parameters.

As soon as the response is received the CONTROLLER can proceed with SET CONF PARAM procedure.

## 1.1. Set Conf Param

This procedure is initiated by the CONTROLLER to set the client configuration for a specific test suite, referring to the names, path and values for configuration parameters available in the CLIENT.

***CONTROLLER behavior***: the CONTROLLER is sending the SET CONF PARAM REQUEST to set the parameters using the names and paths known via the previous procedure, in case. It is including the logical name of such parameters. Timers are not necessary as there is the configuration guard timer that can control even this encapsulated procedure.

***UE behavior***: the UE is sending back the SET CONF PARAM RESPONSE message, including the outcome of the procedure.

In case there is no response from the UE then the Tresp3 timer for the entire set of configuration procedures will expire.

## 1.1. Operation

“Operation” procedure is used to synchronize the tasks of the UE and the Controller. For example it is used to communicate to the Controller the starting of the measurements. The procedure is initiated by the UE and its type is defined into the “Event type” parameter.

***UE behavior:*** the UE is sending the OPERATION REQUEST procedure to communicate the event. The UE is starting the Tresp4 timer to guard against the lack of responses. If the timer expires the UE retries the sending up to Nretry times. If at the end there is no response, the UE will stop the measurements, deregister from the CONTROLLER and come back to IDLE.

***CONTROLLER behavior***: the CONTROLLER is replying to the UE about the outcome of the procedure sending the OPERATION RESPONSE message with the proper result.

The handling of collisions is the same as previously described.

## 1.1. Notification

“Notification” procedure is used typically to notify the peer about an event, as for example the completion of the measurements upload. The main differences compared to OPERATION procedure is that the event is not referred to an operation command and one of the peers involved can be the DATA COLLECTOR.

***UE behavior:*** the UE is sending the message to notify the event (e.g. the completion of the measurements upload). It is started Tresp5 to guard again missing responses. As in previous procedures, if the timer expires, the UE is retrying Nreply times. In case all the retries fail the UE is deregistering from the Controller with proper cause and move the state to Idle.

***Data Collector behavior:*** DATA COLLECTOR is replying with NOTIFICATION RESPONSE, with proper result, including in case additional outcome details (e.g. the path of the data storage).

The procedure can be used even between the DATA COLLECTOR and the CONTROLLER, when it is necessary to inform the CONTROLLER about the successful measurements upload. The behavior is the same as before changing the role of UE into DATA COLLECTOR and DATA COLLECTOR into CONTROLLER.

Note that the CONTROLLER address parameter is present only when the NOTIFICATION REQUEST message is sent by the UE CLIENT.

# 1.1. Messages and parameters format

## 1.1. General on Message Format

A parameters has acronym IE (Information Element) and its format category can be a generic type, length and value or, in some cases, only some of these sub-elements.

For each IE it is specified which category the parameter belongs to.

In case of Structured Data, there will be separate definition of this IE, that can refer recursively to other sub-elements.

## 1.1. CAPABILITIES EXCHANGE REQUEST

Direction: Client 🡪 Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Device ID | Device identifier | M | Octet String (16) |  |
| Permanent Client ID | Client identifier | M | Octet String (16) |  |
| Supported protocol capabilities | Identifies which are the protocol capabilities supported | M | See 4.2 |  |
| Supported test suites  | Identifies which are the test suites supported | M | See 4.3 |  |
| Supported vendor ID  | Vendor dependent parameters | O | See 4.19 |  |
| Current RAT | Current Radio Access Technology | M | Enumerated(GERAN, UMTS, LTE, CDMA 1x, HRPD/eHRPD, WiFi, WiMAX, …) |  |

## 1.1. CAPABILITIES EXCHANGE RESPONSE

Direction: Controller 🡪 Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Result | Identifies the outcome of the procedure | M | Unsigned int 32 | See 4.7 |
| Negotiated protocol capabilities | Parameters negotiated | M | See 4.5 |  |
| Negotiated test suites  | Identifies which are the test suites negotiated | M | See 4.3 |  |

## 1.1. REGISTRATION REQUEST

Direction: Client 🡪 Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Device ID | Device identifier | M | Octet String (16) |  |
| Permanent Client ID | Client identifier | M | Octet String (16) |  |
| Event type | Identifies which is the event the message refers to. | M | Unsigned int 16  | See 4.8 |
| Current RAT | Current Radio Access Technology | M | Enumerated(GERAN, UMTS, LTE, CDMA 1x, HRPD/eHRPD, WiFi, WiMAX, …) |  |
| User credentials | User name and password. They can be ciphered but the way to encrypt them is out of scope here. | M | Octet String (32) | See 4.9 |

## 1.1. REGISTRATION RESPONSE

Direction: Controller 🡪 Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Result | Identifies the outcome of the procedure | M | Unsigned int 32 | See 4.7 |
| New T-ID | Temporary ID assigned to the Client by the Controller | M | Octet string (4) |  |
| New S-ID | Temporary ID assigned to the session by the Controller | M | Octet string (4) |  |
| R-time | Re-registration interval (unit: seconds). If this IE is not present or the value is all 1’s, then UE assumes that it isn’t needed any re-registration. | O | Unsigned int (32) |  |
| S-time | Session lifetime (unit: seconds)If this IE is not present then it is considered the default value as session lifetime. | O | Unsigned int (32) | Default value: 3600 (s) |

## 1.1. GET REG PARAM REQUEST

Direction: Controller 🡪 Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| LogParam Name | List of additional registration parameters (logical name) | M | See 4.10 |  |

## 1.1. GET REG PARAM RESPONSE

Direction: Client 🡪 Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Result | Identifies the outcome of the procedure | M | Unsigned int 32 | See 4.7 |
| Permanent Client ID | Client identifier | M | Octet String (16) |  |
| Last Reg time | Time/day of last registration  | O | See 4.11 |  |
| Last S-ID | Last Session ID assigned to the session by the Controller | M | Octet string (4) |  |
| Last T-ID | Last Temporary ID assigned to the Client by the Controller. Null in case of 1st registration | M | Octet string (4) |  |
| S-Loc | Current location of the Client | M | See 4.12 |  |
| Last RAT | Last Radio Access Technology, where the Client registered previously. | M | Enumerated(GERAN, UMTS, LTE, CDMA 1x, HRPD/eHRPD, WiFi, WiMAX, …) |  |
| Current Registration parameters list | Path, name and values for the parameters requested by the CONTROLLER. The correlation between request and response is the logical name of the parameter. | M | See 4.13 |  |

## 1.1. SET REG PARAM REQUEST

Direction: Controller 🡪 Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Reg time | Time/day of registration  | O | See 4.11 |  |
| New Reg Param settings | List of additional registration parameters set  | M | See 4.13 |  |

## 1.1. SET REG PARAM RESPONSE

Direction: Client 🡪 Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Result | Identifies the outcome of the procedure | M | Unsigned int 32 | See 4.7 |
| Permanent Client ID | Client identifier | M | Octet String (16) |  |

## 1.1. CONFIGURATION REQUEST

Direction: Client 🡪 Controller;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| T-ID | Temporary ID assigned to the Client by the Controller | M | Octet string (4) |  |
| S-ID | Temporary ID assigned to the session by the Controller | M | Octet string (4) |  |
| Event type | Identifies which is the event the message refers to, for example UE-initiated configuration | M | Unsigned int 16  | See 4.8 |
| Test suite  | Test suite to be configured for further activation. Only 1 test suite is present in the parameter. | M | See 4.3 |  |

## 1.1. CONFIGURATION RESPONSE

Direction: Controller 🡪 Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Result | Identifies the outcome of the procedure | M | Unsigned int 32 | See 4.7 |

## 1.1. GET CONF PARAM REQUEST

Direction: Controller 🡪 Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| ConfParam Name | List of configuration parameters (logical name) | M | See 4.6 |  |

## 1.1. GET CONF PARAM RESPONSE

Direction: Client 🡪 Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Result | Identifies the outcome of the procedure | M | Unsigned int 32 | See 4.7 |
| Current Conf Param list | Path, name and values for the configuration parameters requested by the CONTROLLER. The correlation between request and response is the logical name of the parameter. | M | See 4.14 |  |

## 1.1. SET CONF PARAM REQUEST

Direction: Controller 🡪 Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Server address | Address or FQDN of the Server | O | See 4.15 |  |
| Collector address | Address or FQDN of a Collector address | M | See 4.15 |  |
| Configuration time | Time interval for checking changes in configuration of the current test. | O |  | This IE forces the UE to reconfigure after that time interval. |
| New Conf. param. settings | New configuration parameter settings | M |  |  |

## 1.1. SET CONF PARAM RESPONSE

Direction: Client 🡪 Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Result | Identifies the outcome of the procedure | M | Unsigned int 32 | See 4.7 |

## 1.1. OPERATION REQUEST

Direction: Client 🡪 Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| T-ID | Temporary ID assigned to the Client by the Controller | M | Octet string (4) |  |
| S-ID | Temporary ID assigned to the session by the Controller | M | Octet string (4) |  |
| Event type | Identifies which is the event the message refers to. | M | Unsigned int 16  | See 4.8 |
| Cause | Identifies the reason for the event request (e.g. the reason for deregistration) | O | Unsigned int 16  | See 4.18 |

## 1.1. OPERATION RESPONSE

Direction: Controller 🡪 Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Result | Identifies the outcome of the procedure | M | Unsigned int 32 | See 4.7 |

## 1.1. NOTIFICATION REQUEST

Direction: Client 🡪 Data Collector or Data Collector 🡪 Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Originator type | Type of data origin | M | Enumerated(Client, Controller, Data Collector, …) |  |
| T-ID | Temporary ID assigned to the Client by the Controller | M | Octet string (4) |  |
| S-ID | Temporary ID assigned to the session by the Controller | M | Octet string (4) |  |
| Event type | Identifies which is the event the message refers to. | M | Unsigned int 16  | See 4.8 |
| Controller Addr | Address or FQDN of the Server | O | See 4.15 | Present only in case of Client🡪 Data Collector direction. |

## 1.1. NOTIFICATION RESPONSE

Direction: Data Collector 🡪 Client or Controller 🡪 Data Collector

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type | Note |
| Message ID | Message Identifier | M | Unsigned int 16  | See 4.1 |
| Transaction identifier | Identifies the transaction instance | M | Unsigned int 32 |  |
| Result | Identifies the outcome of the procedure | M | Unsigned int 32 | See 4.7 |
| Event details | Identify additional details to notify | O | tbd |  |

# 1.1. IEs format

## 1.1. Message ID

Message ID list:

* CAPABILITIES EXCHANGE REQUEST 0x01
* CAPABILITIES EXCHANGE RESPONSE 0x02
* REGISTRATION REQUEST 0x03
* REGISTRATION RESPONSE 0x04
* GET REG. PARAM. REQUEST 0x05
* GET REG. PARAM. RESPONSE 0x06
* SET REG. PARAM. REQUEST 0x07
* SET REG. PARAM. RESPONSE 0x08
* CONFIGURATION REQUEST 0x09
* CONFIGURATION RESPONSE 0x0A
* GET CONF.PARAM.REQUEST 0x0B
* GET CONF. PARAM. RESPONSE 0x0C
* SET CONF.PARAM.REQUEST 0x0D
* SET CONF. PARAM. RESPONSE 0x0E
* OPERATION REQUEST 0x0F
* OPERATION RESPONSE 0x10
* NOTIFICATION REQUEST 0x11
* NOTIFICATION RESPONSE 0x12
* Other values reserved

## 1.1. Supported protocol capabilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| Capabilities container | Include the details of the capabilities | M | Octet string | tbd |

## 1.1. Test suites

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| Test suites  | Include the details of the test suites | M | Octet string | tbd |

## 1.1. IE identifiers

* Supported protocol capabilities 0x01
* Test suites 0x02
* Supported Vendor ID 0x03
* Negotiated protocol capabilities 0x04
* Negotiated test suites 0x05
* LogParam name 0x06
* LastReg time 0x07
* S-Loc 0x08
* Registration parameter list 0x09
* Configuration Parameters names 0x0A
* Configuration parameter list 0x0B
* Address 0x0C
* Configuration time 0x0D
* New configuration parameter settings 0x0E
* Other values reserved

## 1.1. Negotiated protocol capabilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| Negotiated capabilities container | Include the details of the capabilities | M | Octet string | tbd |

## 1.1. Configuration parameters names

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| Configuration parameters list  | Include the list of parameters with their logical names. | M | Octet string | tbd |

## 1.1. Result

The categories are the following while specific numbers into each category are tbd:

* 1xxx (Informational)
* 2xxx (Success)
* 3xxx (Protocol Errors)
* 4xxx (Transient Failures)
* 5xxx (Permanent Failure)

## 1.1. Event type

* Registration event 0x01
* UE-initiated configuration event 0x02
* Server-initiated configuration event 0x03
* Start 0x04
* Stop 0x05
* Deregistration 0x06
* Upload request 0x07
* Upload completed 0x08
* Other values reserved

## 1.1. User name and password

User name Octet string (16)

Password Octet string (16)

## 1.1. LogParam name

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| LogParam name container | Include the details of the parameters | M | Octet string | tbd |

## 1.1. Last Reg time

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| Last Reg Time container | Include the details of the parameters | M | Octet string | tbd |

## 1.1. S-loc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| Location Format | Type of location (e.g. Location Area + Cell ID, Routing Area Code + Cell ID, Sector, …) | M | Unsigned int 16 | tbd all the specific values |
| Location format container | Include the details of the location parameters | M | Octet string | tbd |

## 1.1. Registration parameter list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| Registration parameter format container | Include the details of the registration parameters. Each element has a TLV format again, to achieve flexibility for including any type of parameter.Each element must include the same logical name requested by the CONTROLLER. | M | Octet string | tbd |

## 1.1. Configuration parameter list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| Configuration parameter format container | Include the details of the configuration parameters. Each element has a TLV format again, to achieve flexibility for including any type of parameter.Each element must include the same logical name requested by the CONTROLLER. | M | Octet string | tbd |

## 1.1. Address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| Address format | Identifies if it is FQDN or IP address (IPv4 or IPv6) | M | Unsigned int16 | Value 1 = FQDNValue 2 = IPv4Value 3 = IPv6 |
| Address container | Address value | M | BIT STRING (1..160, …) in case of IP addressOCTET STRING in case of FQDN |  |

## 1.1. Configuration time

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| Configuration time | Time for configuration checking | M | Unsigned int16 |  |

## 1.1. New configuration parameter settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| Configuration parameter format container | Include the details of the configuration parameters. Each element has a TLV format. | M | Octet string | tbd |

## 1.1. Cause

* UE-initiated-noAnswer 0x01
* UE-initiated-Restart 0x02
* Controller-initiated-noAnswer 0x03
* Controller-initiated-Restart 0x04
* No-Answer-FromDataCollector 0x05
* Other values reserved

## 1.1. Supported vendor ID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE | Semantics | Mandatory/ Optional/ Conditional | Type |  |
| Type | Identifies the IE | M | Unsigned int 16  | See 4.4 |
| Length | Length of the parameter in octets, without Type and Length. | M | Unsigned int 32 |  |
| Vendor identifier | Identifies the specific vendor | M | Octet string(32) | tbd |
| Vendor dependent info | Include the details of the vendor dependent parameters. Each element has a TLV format. | M | Octet string | tbd |

# 1.1. Timers and Internal values

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Timer | Default value | Configurable (Yes – No) | Range of values |  |
| Tresp1 | 2s | Yes |  2 – 3600s |  |
| Tresp2 | 2s | Yes |  2 – 3600s |  |
| Tresp3 | 2s | Yes |  2 – 3600s |  |
| Tresp4 | 2s | Yes |  2 – 3600s |  |
| Tresp5 | 2s | Yes |  2 – 3600s |  |
| Tguard1 | 60s | Yes |  2 – 3600s |  |
| Tretry | 300s | Yes |  2 – 3600s |  |

Table 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Internal value | Default value | Configurable (Yes – No) | Range of values |  |
| Nretry | 2 | Yes |  2 - 100 |  |

Table 2