**Comments and Suggested Revisions**

**Towards the Description of SCOS-communications**

**Related Sections of the 802.22.3 Standard Draft v1.0**

*Nilesh Khambekar*

Date: March 4th, 2017 [Comment version 1.0 ]

* + 1. Comments

Following are the comments on SCOS Communication described in Section of the standard version 1.0.

1. Communication model
2. Response codes
3. Message encoding structure
4. Message Transport
5. Details of the Message parameters
6. Table of Messages for SCOS Operational procedures
	* 1. Suggestions

Following draft attempts to address the above mentioned comments on the standard draft v1.0 towards communication between the SCOS system entities.

1. SCOS Communication
	1. SCOS communication interfaces

Following are the key SCOS communication interfaces.

* Actor and SSM Interface
* SSM and SSD Interface
	+ The interface between SSM and SSD is required to be synchronous.
* SSD and SDM Interface
* SDM and DRP Interface
	+ The interface between SSD and SDM is required to be asynchronous.
	1. Response codes

0: success

100 – 199: error events related to the entity association and disassociation

200 – 299: error events related to the entity configuration and policy enforcement

300 – 399: error events related to the scanning procedures

400 – 499: error events related to the data dissemination procedures

500 – 599: error events related to the heartbeat procedure and SCOS entity/platform health

600 – 699: error events related to the SCOS infrastructure administrative procedures

* 1. Message Encoding
* SCOS messages are encoded in JavaScript Object Notation (JSON). JSON is a language-independent data-interchange format that is easy for humans to read and write. There are code and functions readily available in C, C++, C#, Java, JavaScript, MATLAB, Perl, and Python for parsing and generating JSON. It is a lightweight alternative to XML, commonly used to transmit data between server and browser applications.
* The first five fields are the same for all messages; they are:

1. Ver = Schema/data transfer version with the major.minor.revision syntax (string)

2. Type = Type of JSON message (string) {“Sys”, ”Loc”, or “Data”}

3. SensorID = Unique identifier of sensor (string of URL unreserved characters)

4. SensorKey = Authentication key given out by MSOD (integer)

5. t = Time [seconds since Jan 1, 1970 UTC] (long integer)

* Each message begins with a header comprised of attribute-value pairs in ASCII characters. [#confirm]
* If an attribute is not relevant to the sensor implementation, then the value is set to NaN or "NaN".
* The following are specific formatting rules to be followed:
* All timestamps, i.e., t (defined above)and t1 (to be defined in Data message description) will be reported as seconds since 1/1/1970 midnight UTC in the UTC time zone.
* String values must only contain URL unreserved characters (i.e., uppercase and lowercase letters, decimal digits, hyphen, period, underscore, and tilde), and
* Field names cannot start with an underscore because that convention is reserved for internal implementation-specific uses.
	1. Message Transport
* The standard requires reliable transport mechanism.
	+ Within MQTT, QOS 1 (AT-LEAST-ONCE) is suggested.
* The standard recommends MQTT transport for communication between SSM and SSDs [#RecommendedTransport? #AddReference MQTT].
* Transport security is ensured using the security protocols and procedures within the transport mechanism.
	+ Example: MQTT messages can be secured using TLS. Please refer to [#Reference] for security mechanism in MQTT.
	1. SCOS Message exchanges

Following table identifies messages across different entities.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operational procedure**  | **Message Name** | **Description** | **Transport** | **Section Ref** |
| Actor association |  | From:To:Purpose: |  |  |
| Actor dis-association |  | From:To:Purpose: |  |  |
| DRP association |  | From:To:Purpose: |  |  |
| DRP dis-association |  | From:To:Purpose: |  |  |
| SSD association |  | From:To:Purpose: |  |  |
| SSD dis-association |  | From:To:Purpose: |  |  |
| Spectrum-scan |  | From:To:Purpose: |  |  |
| Spectrum-data-dissemination |  | From:To:Purpose: |  |  |
| Heartbeat |  | From:To:Purpose: |  |  |
| Calibration |  | From:To:Purpose: |  |  |
| Configuration |  | From:To:Purpose: |  |  |
| Query |  | From:To:Purpose: |  |  |
| Reporting |  | From:To:Purpose: |  |  |
| Management  |  | From:To:Purpose:Note: Platform control message |  |  |

* 1. Details of the Message Parameters
		1. SSD Association
			1. SSD-SSM Association Message exchange

[#Revise #AddFormatContent]

|  |  |
| --- | --- |
|  |  |
|  |  |
| SSD Association: |  |
|  |  |
| SSD Send: |  |
| SSD associate { |  |
|  Association [request new, request refresh, request disassociate] |  |
|  SSD Device ID |  |
|  Public Key} | SSM would have all possible SSDs’ public keys that can associate |
|  |  |
| SSM Reply: |  |
| SSM associate { |  |
|  SSD Device ID |  |
|  Association state granted [new, refresh, disassociate] |  |
|  SSD association max TTL | Tells SSD how long before it gets auto disconnected |
|  SSD association remaining TTL} | Tells SSD how long left on clock |
|  |  |

* + 1. SSM-SSD Scan

[#Revise SSD Send should be Separate #AddFormatContent]

|  |  |
| --- | --- |
| SSD Advertise: |  |
|  |  |
| SSD Send: |  |
| SSD spec {  |  |
|  Fmin |  |
|  Fmax |  |
|  Resolution |  |
|  Algorithm Type 1..n |  |
|  Antenna type |  |
|  Antenna direction |  |
|  GPS available |  |
|  GPS location |  |
|  …etc } | All standard defined metadata types |
|  |  |
| SSM Reply: |  |
| SSM scan schedule for (SSD Device ID)  |  |
|  Scan 1 time { |  |
|  Scan Schedule Sequence Number | Unique to scan |
|  Time Start Offset | Time in minutes from start of week |
|  Time Slots | How many minutes to block |
|  Repeat start offset | Interval to repeat start |
|  Number of repeats | How many repeats (anything outside week window are dropped) |
|  Scan 1 spec {  |  |
|  Fmin | Desired Fmin |
|  Fmax | Desired Fmax |
|  Resolution | Desired res |
|  Algorithm Type | Give desired algorithm type |
|  Antenna type | Give expected antenna type |
|  Antenna direction | Give expected/desired antenna direction |
|  GPS enable | Yes/No |
|  GPS location | Give expected GPS location |
|  …etc } | All standard defined metadata types |
|  |  |
|  |  |
|  Scan 1 destination {  |  |
|  Actor public key | Actor’s public key identifier  |
|  MQ topic or URL of data destination} | Where data will be published to |
|  |  |
| Etc … Scan n |  |
|  |  |
|  |  |
| SSD reply on scan execute: |  |
|  Scan Schedule Sequence Number |  |
|  Scan Completion Code | 1 – complete, 2 – incomplete, 3 – rejected  |
|  Scan Completion Fstart | Freq where scan was to start |
|  Scan Completion Fend | Freq that was attained (“0” for invalid) |
|  Scan parameter that caused rejection | If Completion Code 3, this gives problem metric |
|  |  |
|  |  |
|  |  |
|  |  |

* + 1. SSD Sensing Data Transmission
		2. Sensing Data Dissemination to DRP
		3. Heartbeat
		4. Configuration
		5. Calibration
		6. Query
		7. Log
		8. Report
		9. Management
		10. SSD Dis-association
		11. Actor Association
		12. DRP Association
		13. Actor Dis-association
		14. DRP Dis-association