

Summary of Changes

Managing gateways in Semtech LoRaWAN Localization

The manufacturer can be added when registering gateway

Semtech LoRaWAN Location algorithm & Localization Lora Recipes changes

The JSON root key for LoRaWAN receptions is now "lorawan"
Improvements of RSSI and TDOA algorithms

Previous release: A1.0.12

Release deployment date: 27th June 2018,

Release deployment time: 10:00 CEST

Version	ECO	Date	Changes and/or Modifications
1	-	22 June 2018	New document.

Contents

- 1. Detailed changes2
- 1.1. Semtech LoRaWAN Localization Query changes2
- 1.2. Various error messages improvements5
- 1.3. Add a gateway type property5
- 2. Deprecations5

1. Detailed changes

1.1. Semtech LoRaWAN Localization Query changes

There are changes to the query contract bringing significant improvements to the flexibility and usability.

With this version, gateways have multiple potential ways of being identified within queries sent to location ingredients or recipes (ie. The “gatewayId” parameter can refer to the following):

1. “gatewayId” = “label” from when GW was provisioned, in which case a location **is not** required
2. “gatewayId” = “uniqueId” from when GW was provisioned, in which case a location **is not** required
3. “gatewayId” = “chipId” from when GW was provisioned, in which case a location **is not** required
4. “gatewayId” = “chipId” from a GW that is not provisioned, in which case a location **is** required
5. “gatewayId” = “temporary label” from a GW that is not provisioned, in which case a location **is** required

Using the ‘temporary label’ choice does not support the decryption of encryptedToa fields since there is no way to cut the decryption key. For all other choices the gateway must be registered with a valid key.

With previous releases, the location recipes had two structures: (1) an array of gateway and (2) an array of WiFi access points. The gatewayReceptions has been renamed to lorawan to allow for the future when different types of gateways (non LoRaWAN) may be introduced.

LoRaWiFi and LoRa Skyhook recipes with this release (API V2)	Previous releases (API V1)
<pre> { "lorawan": [{ "gatewayId": "string", "antennaId": 0, "rssi": 0.0, "snr": 0.0, "toa": 0, "encryptedToa": "string", "antennaLocation": { "latitude": 0.0, "longitude": 0.0, "altitude": 0.0 } }], "wifiAccessPoints": [{ "age": 0, "macAddress": "string", "signalStrength": 0 }] } </pre>	<pre> { "gatewayReceptions": [{ "gatewayId": "string", "antennaId": 0, "rssi": 0.0, "snr": 0.0, "toa": 0, "encryptedToa": "string", "antennaLocation": { "latitude": 0.0, "longitude": 0.0, "altitude": 0.0 } }], "wifiAccessPoints": [{ "age": 0, "macAddress": "string", "signalStrength": 0 }] } </pre>

With previous releases, the location algorithms had an array of gateway receptions. With this release the gateway receptions have been pushed into a 'lorawan' structure. With this change, all endpoints have become 100% compatible.

This means that a single query can be formed and sent to the URL of any of the endpoints and they will be accepted. This considerably reduces friction in the integration process.

Location algorithm queries with this release (API V2)	Previous releases (API V1)
<pre> { "lorawan": [{ "gatewayId": "string", "antennaId": 0, "rssi": 0.0, "snr": 0.0, "antennaLocation": { "latitude": 0.0, "longitude": 0.0, "altitude": 0.0 } }] } </pre>	<pre> [{ "gatewayId": "string", "antennaId": 0, "rssi": 0.0, "snr": 0.0, "antennaLocation": { "latitude": 0.0, "longitude": 0.0, "altitude": 0.0 } }] </pre>

Considerable improvements have also been made to the returned information with a location result.

The algorithms will attempt to use multiple technologies to resolve a location from a query. If there are not enough timestamps for TDOA, then it may attempt an RSSI based location. An RSSI based location can be returned even if there is only a single gateway- in which case the location of the gateway is returned.

The new return codes include the algorithm used and the number of gateways used in the calculation. If the algorithm determined that some of the information appears to be inconsistent between gateways in the query, then it will discard some of the gateway reception information. Therefore, not only will the algorithm now return the algorithms and the number of gateways used in the calculation, but also the number of gateways that it was sent.

This allows simple user selection to use or discard locations based on too few gateways, for example.

Location algorithm responses with this release (API V2)	Previous releases (API V1)
<pre> { "result": { "latitude": 0.0, "longitude": 0.0, "altitude": 0.0, "accuracy": 0.0, "algorithmType": "string", "numberOfGatewaysReceived": 0, "numberOfGatewaysUsed": 0 }, "warnings": ["string"], "errors": ["string"], "correlationId": "string" } </pre>	<pre> { "result": { "latitude": 0.0, "longitude": 0.0, "altitude": 0.0, "accuracy": 0.0 }, "warnings": ["string"], "errors": ["string"], "correlationId": "string" } </pre>

1.2. Various error messages improvements

- If any gateway has a decryption failure, the warning message now returns its gatewayId
- correlationId is now returned only one time in JSON root object

1.3. Add a gateway type property

The gateway registration API has a new optional “type” field. This will allow Collos to cut decryption keys in the future.

This property is optional, but “other” can be used if the gateway manufacturer is not in the list. In that case gateway keys will not be cut by Collos.

Manufacturer list:

A2A	Data Alliance	Kerlink	Sagemcom
Aartsys	Foxconn	Multitech	Semtech
Actility	Helium	NASYS	Telenet
Chronos	IMST	Pipeline	ZTE
Contela	Invisible Systems	Proesys	Other

2. Deprecations

No deprecation in this release. Please note that V1 localization endpoints might be deprecated in the future release.

End of document