PAWS
Internet-Draft
Intended status: Standards Track
Expires: March 28, 2015

V. Chen, Ed.
Google
S. Das
Applied Communication Sciences
L. Zhu
Huawei
J. Malyar
iconectiv
P. McCann
Huawei

September 24, 2014

Protocol to Access White-Space (PAWS) Databases
draft-ietf-paws-protocol-19

Abstract

Portions of the radio spectrum that are allocated to licensees are available for non-interfering use. This available spectrum is called "White Space." Allowing secondary users access to available spectrum "unlocks" existing spectrum to maximize its utilization and to provide opportunities for innovation, resulting in greater overall spectrum utilization.

One approach to managing spectrum sharing uses databases to report spectrum availability to devices. To achieve interoperability among multiple devices and databases, a standardized protocol must be defined and implemented. This document defines such a protocol, the "Protocol to Access White Space (PAWS) Databases".

Table of Contents

1. Introduction

[RFC Editor: In the Author's Addresses section, please list "iconectiv" as "iconectiv (formerly Telcordia Interconnection Solutions). One occurrence."]

This section provides some high level introductory material. Readers are strongly encouraged to read "Protocol to Access White-Space (PAWS) Databases: Use Cases and Requirements" [RFC6953] for use cases, requirements, and additional background.

A geospatial database can track available spectrum (in accordance with the rules of one or more regulatory domains) and make this information available to devices. This approach shifts the complexity of spectrum-policy conformance out of the device and into the database. This approach also simplifies adoption of policy changes, limiting updates to a handful of databases, rather than numerous devices. It opens the door for innovations in spectrum management that can incorporate a variety of parameters, including user location and time. In the future, it also can include other parameters, such as user priority, signal type and power, spectrum supply and demand, payment or micro-auction bidding, and more.

In providing this service, a database records and updates information necessary to protect primary users -- for example, this information may include parameters such as a fixed transmitter's call sign, its geolocation, antenna height, power, and periods of operation. The rules that the database is required to follow, including its schedule for obtaining and updating protection information, protection rules, and information reported to devices, vary according to regulatory domain. Such variations, however, should be handled by each database, and hidden from devices to the maximum extent possible.

This specification defines an extensible protocol, built on top of HTTP and TLS, to obtain available spectrum from a geospatial database by a device with geolocation capability. It enables a device to operate in a regulatory domain that implements this protocol and within which the device operates.

2. Conventions and Terminology

2.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

2.2. Terminology

Database or Spectrum Database: A database is an entity that contains current information about available spectrum at a given location and time, as well as other types of information related to spectrum availability and usage.

Device ID: An identifier for a device.

EIRP: Effective isotropically radiated power

ETSI: European Telecommunications Standards Institute (http://www.etsi.org)

FCC: The U.S. Federal Communications Commission (http://www.fcc.gov)

Listing server: A server that provides the URIs for one or more Spectrum Databases. A regulator, for example, may operate a Database Listing Server to publish the list of authorized Spectrum Databases for its regulatory domain.

Master Device: A device that queries the database, on its own behalf and/or on behalf of a slave device, to obtain available spectrum information.

Regulatory Domain: A location where certain rules apply to the use of white space spectrum, including the operation of databases and devices involved in its use. A regulatory domain is normally defined by a unit of government for a particular country, but PAWS is agnostic as to how a regulatory domain is constructed.

Ruleset: A ruleset represents a set of rules that governs the operation of white space devices and Spectrum Databases. A regulatory authority can define its own set of rules or adopt an existing ruleset. When a Database or device is said to "support a ruleset", it means that it contains out-of-band knowledge of the rules and that its hardware and software implementations conform to those rules.

Ruleset Identifier: A ruleset can be identified by an IANA-registered identifier (see PAWS Ruleset ID Registry (Section 9.1)). When a Database or device indicates it supports a ruleset identifier, it means that it conforms to the rules associated with that identifier. A regulatory authority can define and register its own ruleset identifiers, or it can use a previously registered identifier if it adopts an existing ruleset.

Slave Device: A device that queries the database through a master device.

3. Protocol Overview

A Master Device uses PAWS to obtain a schedule of available spectrum at its location. The security necessary to ensure the accuracy, privacy, and confidentiality of the device's location is described in the Security Considerations (Section 10). This document assumes that the Master Device and the Database are connected to the Internet.

A typical sequence of PAWS operations is outlined as follows. See Protocol Functionalities (Section 4) and Protocol Parameters (Section 5) for details:

1. The Master Device obtains (statically or dynamically) the URI for a Database appropriate for its location, to which to send subsequent PAWS messages.

2. The Master Device establishes an HTTPS session with the Database.

3. The Master Device optionally sends an initialization message to the Database to exchange capabilities.

4. If the Database receives an initialization message, it responds with an initialization-response message in the body of the HTTP response.

5. The Database may require the Master Device to be registered before providing service.

6. The Master Device sends an available-spectrum request message to the Database. The message may be on behalf of a Slave Device that made a request to the Master Device.

7. If the Master Device is making a request on behalf of a Slave Device, the Master Device may verify with the Database that the Slave Device is valid.

8. The Database responds with an available-spectrum response message in the body of the HTTP response.

9. The Master Device may send a spectrum-usage notification message to the Database. The notification is purely informational for notifying the Database what spectrum it intends to use and is not a request to the Database to get permission to use that spectrum.

10. If the Database receives a spectrum-usage notification message, it responds by sending the Master Device a spectrum-usage acknowledgement message. Since the notification is purely informational, the Master Device does not need to process the Database response.

Different regulatory domains may impose particular requirements, such as requiring Master Devices to register with the Database, performing Slave Device verification, and sending spectrum usage notifications.

3.1. Multi-ruleset Support

For a Master Device that supports multiple rulesets and operates with multiple databases, PAWS supports the following sequence of operations for each request by the Master Device:

1. The Master Device includes in its request its location and optionally includes the identifier of all the rulesets it supports and any parameter values it might need for the request

2. The Database uses the device location and also may use the ruleset list to determine its response, for example, to select the list of required parameters

3. If required parameters are missing from the request, the Database responds with an error and a list of names of the missing parameters

4. The Master Device makes the request again, adding the missing parameter values

5. The Database responds to the request, including the identifier of the applicable ruleset

6. The Master Device uses the indicated ruleset to determine how to interpret the Database response

NOTE Some regulatory domains specify sets of requirements for device behavior that may be complex and not easily parameterized. The

ruleset-id parameter provides a mechanism for the Database to inform the Master Device of an applicable ruleset, and, for devices with out-of-band knowledge of the particular regulatory domain requirements, to satisfy those requirements without having to specify the device-side behavior within the protocol. Ruleset identifiers will normally contain the name of the regulatory body that established the rules and version information, such as "FccTvBandWhiteSpace-2010".

By separating the regulatory "authority" from the "ruleset-id", it allows the protocol to support multiple regulatory authorities that use the same device-side ruleset. It also allows support for a single authority to define multiple rulesets.

4. Protocol Functionalities

PAWS consists of several components. As noted below, some regulatory domains or Database implementations may mandate the use of a component, even when its use is not mandated by PAWS.

- Database Discovery (Section 4.1) is a required component for the Master Device.

- Initialization (Section 4.3) is a required component for the Database. Its use allows the Master Device to determine necessary information that have not been preconfigured.

- Device Registration (Section 4.4) is an optional component for the Database. It can be implemented as a separate component or as part of the Available Spectrum Query (Section 4.5) component. It is used by the Master Device when the Database requires it. Note that some regulators require device registration for only specific device types, such as higher-power fixed (as opposed to mobile) devices to allow them to contact the operators to resolve any interference issues.

- Available Spectrum Query (Section 4.5) is a required component for the Master Device and the Database.

- Spectrum Use Notify (Section 4.5.5) is an optional component for the Master Device and the Database. When it is required, the Database informs the Master Device via its response to the Available Spectrum Query (Section 4.5).

- Device Validation (Section 4.6) as a separate component is optional for the Master Device and Database. When implemented by the Database, its use allows the Master Device to validate Slave Devices without having to use the full Available Spectrum Query.

This section describes the protocol components and their messages. Protocol Parameters (Section 5) contains a more thorough discussion of the parameters that make up the PAWS request and response messages. Message Encoding (Section 6) provides examples of message encodings. HTTPS Binding (Section 7) describes the use of HTTPS ("HTTP Over TLS" [RFC2818]) for transferring PAWS messages and optional device authentication.

The parameter tables in this section and Protocol Parameters (Section 5) are for reference and contain the name of each parameter, the data type of each parameter, and whether the existence of the parameter is required for the protocol transaction in question. The diagrams are loosely based on UML, and the data types are defined either in Protocol Parameters (Section 5) or are one of the following primitive or structured types:

string: A string, as defined by JSON [RFC7159], restricted to the UTF-8 encoding.

int: A number, as defined by JSON [RFC7159], without a fractional or exponent part.

float: A number, as defined by JSON [RFC7159].

boolean: A boolean, as defined by JSON [RFC7159].

list: A structured type that represents a list of elements, as defined by JSON [RFC7159] array type. All elements of the list are of the same data type, which is indicated in its diagram and description. The diagram notation and description may include additional constraints, such as minimum or maximum number of elements.

Also:

- All parameter names are case sensitive. Unless stated otherwise, all string values are case sensitive.

- All timestamps are in UTC and are expressed using exactly the form, YYYY-MM-DDThh:mm:ssZ, as defined by "Date and Time on the Internet: Timestamps" [RFC3339].

In some cases, specific rulesets may place additional requirements on message parameters. These additional requirements will be documented in the IANA PAWS Ruleset ID Registry. (Section 9.1). When a request message sent to the Database has missing parameters, whether they are required by PAWS or the applicable ruleset, the Database returns the

MISSING (Section 5.17.3) error, along with data indicating the missing parameters.

4.1. Database Discovery

Preconfiguration

The Master Device can be provisioned statically (preconfigured) with the URI of one or more Databases. For example, in a particular regulatory domain, there may be a number of certified Databases that any device operating in that domain is permitted to connect to, and those URIs can be preconfigured in the device.

Listing Server Support: As an alternative to preconfiguring devices with a list of certified Databases, some regulatory domains support the preconfiguration of devices with the URI of a certified listing server, to which devices can connect to obtain the list of certified Databases. See Listing Server Support (Appendix A) for further information.

Configuration Update: Database URI changes

To adapt to changes in the list of certified or approved databases, the device needs to update its preconfigured list of databases.

A Database MAY change its URI, but before it changes its URI, it MUST indicate so by including the URI of one or more alternate databases using DbUpdateSpec (Section 5.7) in its responses to devices. The Database MUST reply with DbUpdateSpec for a minimum of 2 weeks before disabling the old URI. A device will update its preconfigured entry for the Database sending the DbUpdateSpec by replacing this entry with the alternate databases listed in the DbUpdateSpec; the list of alternate databases does not affect any other entries. Note that the ordering of databases in the list does not imply any preference and does not need to remain the same for every request. The device SHOULD detect infinite redirection loops; if a suitable database cannot be contacted, the device MUST treat this as equivalent to a response indicating no available spectrum. This database-change mechanism is used, for example, before a Database ceases operation; it is not intended to be used for dynamic load balancing.

Error Handling

The device SHOULD select another database from its list of preconfigured databases if:

o The selected database is unreachable or does not respond.

- The selected database returns an UNSUPPORTED error (see Error Codes (Section 5.17)), which indicates the database does not support the device (based on its device type, model, etc.) or supports none of the rulesets specified in the request.

If a suitable database cannot be contacted, the device MUST treat this as equivalent to a response indicating no available spectrum. If the device had previously contacted a database to get available spectrum, but subsequently fails to contact a suitable database, the spectrum the device is currently using can be used for as long as the spectrum data is valid; but after that period, the device will no longer have valid spectrum to use. Some regulatory domains may have specific rules regarding how long the spectrum data remains valid in these cases.

4.2. PAWS Version

PAWS version uses a "<major>.<minor>" numbering scheme to indicate versions of the protocol. The protocol versioning policy is intended to allow the device or Database to indicate the format of a message and its understanding of PAWS functionality defined by that version. No change is made to the version string for the addition of message components which only add to extensible parameter values. The <minor> number is incremented when the changes made to the protocol add functionalities (methods), but do not change the existing functionalities. The <major> number is incremented when incompatible changes are made to existing functionality.

The current PAWS version is "1.0".

4.3. Initialization

A Master Device SHOULD use the initialization procedure to exchange capability information with the Database whenever the Master Device powers up or initiates communication with the Database. The initialization response informs the Master Device of specific parameterized-rule values for each supported ruleset, such as threshold distances and time periods beyond which the device must update its available-spectrum data (see RuleSetInfo (Section 5.6)). When parameterized-rule values are not preconfigured for the applicable ruleset at the specified location, a Master Device MUST use the initialization procedure.

It is important to note that, when parameterized-rule values are preconfigured in a Master Device, they are preconfigured on a per-ruleset basis. That is, values preconfigured for one ruleset is not applicable to any other ruleset.

For database implementations that require it, the initialization message also enables extra database-specific or ruleset-specific handshake parameters to be communicated before allowing available-spectrum requests.

The Initialization request procedure is depicted in Figure 1.

- INIT_REQ (Section 4.3.1) is the initialization request message
- INIT_RESP (Section 4.3.2) is the initialization response message

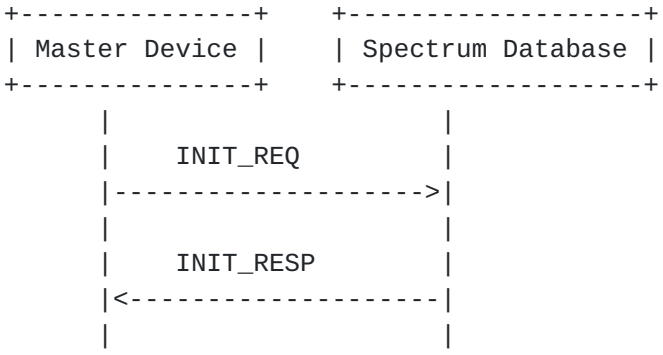


Figure 1

4.3.1. INIT_REQ

The initialization request message allows the Master Device to initiate exchange of capabilities with the Database.

INIT_REQ	
deviceDesc:DeviceDescriptor	required
location:GeoLocation	required
*other:any	optional

Parameters:

deviceDesc: The DeviceDescriptor (Section 5.2) for the device is REQUIRED. If the device descriptor does not contain any ruleset IDs, the Master Device is asking the Database to return a RulesetInfo (Section 5.6) list that specifies the rulesets that it supports at the specified location.

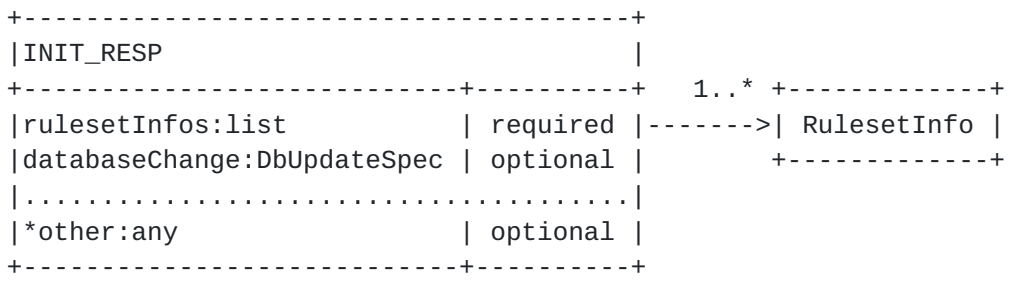
location: The GeoLocation (Section 5.1) of the device is REQUIRED. If the location is outside all regulatory domain supported by the

Database, the Database MUST respond with an OUTSIDE_COVERAGE (Table 1) error.

other: The Master Device MAY specify additional handshake parameters in the INIT_REQ message. The Database MUST ignore all parameters it does not understand. To simplify its initialization logic, a Master Device that supports multiple Databases and rulesets can include the union of all required parameters for all its supported rulesets. Consult the PAWS Parameters Registry (Section 9.2) for possible additional parameters.

4.3.2. INIT_RESP

The initialization response message communicates database parameters to the requesting device. This response is returned only when there is at least one ruleset. Otherwise, the Database returns an error response, as described in INIT_REQ (Section 4.3.1).



Parameters:

rulesetInfos: A RulesetInfo (Section 5.6) list MUST be included in the response. Each RulesetInfo corresponds to a ruleset supported by the Database and is applicable to the location specified in the INIT_REQ (Section 4.3.1) message.

If the device included a list of ruleset IDs in the DeviceDescriptor of its INIT_REQ message, each RulesetInfo in the response MUST match one of the specified ruleset IDs.

If the DeviceDescriptor did not contain any ruleset IDs, the Database SHOULD include in the rulesetInfos list a RulesetInfo for each ruleset it supports at the specified location.

If the Database does not support the device or supports none of the rulesets specified in the DeviceDescriptor, it MUST instead return an error with the UNSUPPORTED (Table 1) code in the error response.

databaseChange: The Database MAY include a DbUpdateSpec (Section 5.7) to notify the Master Device of a change to the Database URI, providing one or more alternate database URIs. The device needs to update its preconfigured entry for the responding database with the alternate databases listed in the DbUpdateSpec.

other: The Database MAY include additional handshake parameters in the INIT_RESP (Section 4.3.2) message. The Master Device MUST ignore all parameters it does not understand. Consult the PAWS Parameters Registry (Section 9.2) for possible additional parameters.

4.4. Device Registration

Some rulesets require a Master Device to send its registration information to the Database in order to establish certain operational parameters. FCC rules, for example, require that a 'Fixed Device' register its owner and operator contact information, its device identifier, its location, and its antenna height (see FCC CFR47-15H [FCC-CFR47-15H]).

The Database MAY implement device registration as a separate Device Registration request, or as part of the Spectrum Availability request. If the Database does not implement a separate Device Registration request, it MUST return an error with the UNIMPLEMENTED (Table 1) code in the error-response message.

The Device Registration request procedure is depicted in Figure 2.

- REGISTRATION_REQ (Section 4.4.1) is the device-registration request message
- REGISTRATION_RESP (Section 4.4.2) is the device-registration response message

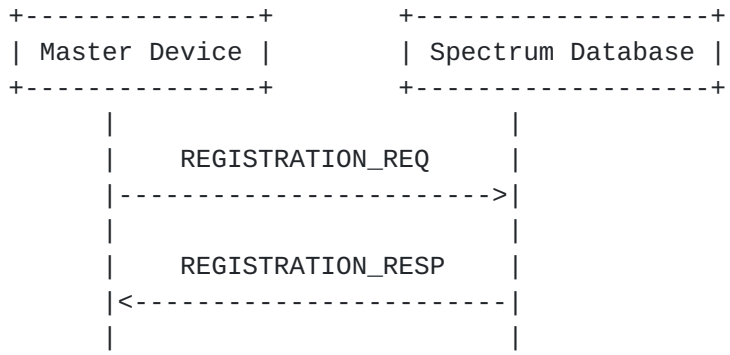


Figure 2

4.4.1. REGISTRATION_REQ

The registration request message contains the required registration parameters. A parameter marked as optional may be required by some rulesets.

REGISTRATION_REQ	
deviceDesc:DeviceDescriptor	required
location:GeoLocation	required
deviceOwner:DeviceOwner	optional
antenna:AntennaCharacteristics	optional
*other:any	optional

Parameters:

deviceDesc: The DeviceDescriptor (Section 5.2) for the Master Device is REQUIRED. The ruleset IDs included in the DeviceDescriptor indicate the rulesets for which the device wishes to register.

location: The GeoLocation (Section 5.1) for the device is REQUIRED. More precisely, this is the location at which the device intends to operate. If the location is outside all regulatory domains supported by the Database, the Database MUST respond with an OUTSIDE_COVERAGE (Table 1) error.

deviceOwner: The DeviceOwner (Section 5.5) information is OPTIONAL. Some rulesets may require deviceOwner information under certain conditions. See PAWS Ruleset ID Registry (Section 9.1) for ruleset-specific requirements.

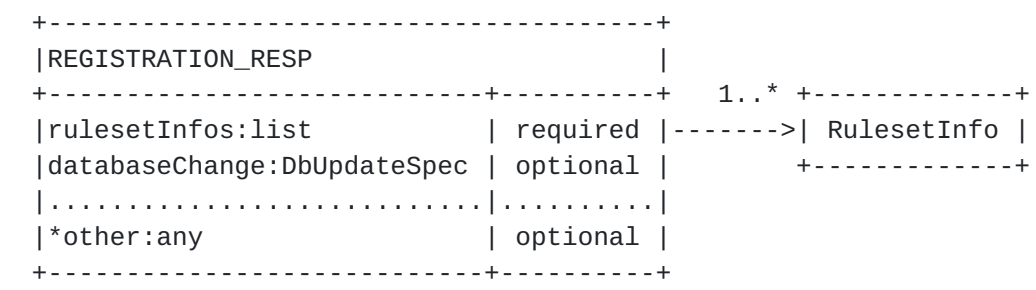
antenna: The AntennaCharacteristics (Section 5.3) is OPTIONAL.

other: Rulesets and database implementations may require additional registration parameters. To simplify its registration logic, the Master Device MAY send a union of the registration information required by all supported rulesets. The Database MUST ignore all parameters it does not understand. Consult the PAWS Parameters Registry (Section 9.2) for possible additional parameters.

4.4.2. REGISTRATION_RESP

The registration response message acknowledges successful registration by including a RulesetInfo message for each ruleset in which the registration is accepted. If the Database accepts the

registration for none of the specified rulesets, the Database MUST return the NOT_REGISTERED error (See Error Codes (Section 5.17)).



Parameters:

rulesetInfos: A RulesetInfo (Section 5.6) list MUST be included in the response. Each entry corresponds to a ruleset for which the registration was accepted. The list MUST contain at least one entry.

Each RulesetInfo in the response MUST match one of the ruleset IDs specified in the DeviceDescriptor of REGISTRATION_REQ.

If the Database does not support the device or supports none of the rulesets specified in the DeviceDescriptor, it MUST instead return an error with the UNSUPPORTED (Table 1) code in the error response.

databaseChange: The Database MAY include a DbUpdateSpec (Section 5.7) to notify the Master Device of a change to the Database URI, providing one or more alternate database URIs. The device needs to update its preconfigured entry for the responding database with the alternate databases listed in the DbUpdateSpec.

other: Database implementations MAY return additional parameters in the registration response. The Master Device MUST ignore any parameters it does not understand. Consult the PAWS Parameters Registry (Section 9.2) for possible additional parameters.

4.5. Available Spectrum Query

To obtain the available spectrum from the Database, a Master Device sends a request that contains its geolocation and any parameters required by the ruleset (such as device identifier, capabilities, and characteristics). The Database returns a response that describes which frequencies are available, at what permissible operating power levels, and a schedule of when they are available.

The Available Spectrum Query procedure is depicted in Figure 3.

- AVAIL_SPECTRUM_REQ (Section 4.5.1) is the available-spectrum request message.

- AVAIL_SPECTRUM_RESP (Section 4.5.2) is the available-spectrum response message.

- AVAIL_SPECTRUM_BATCH_REQ (Section 4.5.3) is an OPTIONAL batch version of the available-spectrum request message that allows multiple locations to be specified in the request.

- AVAIL_SPECTRUM_BATCH_RESP (Section 4.5.4) is the response message for the batch version of the available-spectrum request that contains available spectrum for each location in the request.

- SPECTRUM_USE_NOTIFY (Section 4.5.5) is the spectrum-usage notification message.

- SPECTRUM_USE_RESP (Section 4.5.6) is the spectrum-usage acknowledgment message.

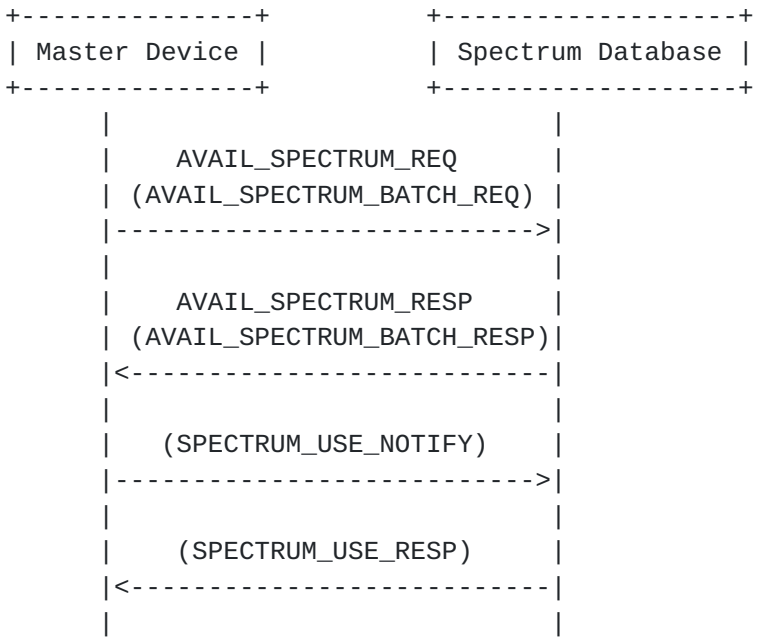


Figure 3

1. First, the Master Device sends an available-spectrum request message to the Database.

2. The Database MUST respond with an error using the NOT_REGISTERED (Table 1) code if:

   * registration information is required, and

* the request does not include registration information, and

* the device has not previously registered with the Database

3. If the location specified in the request is outside the regulatory domain supported by the Database, the Database MUST respond with an OUTSIDE_COVERAGE (Table 1) error. If some, but not all, locations within a batch request are outside the regulatory domain supported by the Database, the Database MUST return an OK response with available spectrum for only the valid locations; otherwise, if all locations within a batch request are outside the regulatory domain, the Database MUST respond with an OUTSIDE_COVERAGE error.

4. The Database MAY perform other validation of the request, (e.g., checking for missing required parameters, authorizations). If validation fails, the Database returns an appropriate error code (Table 1). If the request is missing required parameters, the Database MUST respond with a MISSING (Table 1) error and SHOULD include a list of the missing parameters.

5. If the request is valid, the Database responds with an available-spectrum response message. If the ruleset requires that devices must report anticipated spectrum usage, the Database will indicate so in the response message.

6. If the available-spectrum response indicates that the Master Device must send a spectrum-usage notification message, the Master Device sends the notification message to the Database.

7. If the Database receives a spectrum-usage notification message, it MUST send a spectrum-usage acknowledgment message to the Master Device.

The procedure for a Master Device asking for available spectrum on behalf of a Slave Device is similar, except that the process is initiated by the Slave Device. The device identifier, capabilities, and characteristics communicated in the AVAIL_SPECTRUM_REQ message MUST be those of the Slave Device, and:

- The "masterDeviceLocation" field specifying the location of the Master Device is REQUIRED.

- The "location" field specifying the location of the Slave Device is OPTIONAL, since the Slave Device may not have location-sensing capabilities.

Although the communication and protocol between the Slave Device and Master Device is outside the scope of this document (represented as dotted lines), the expected message sequence is shown in Figure 4.

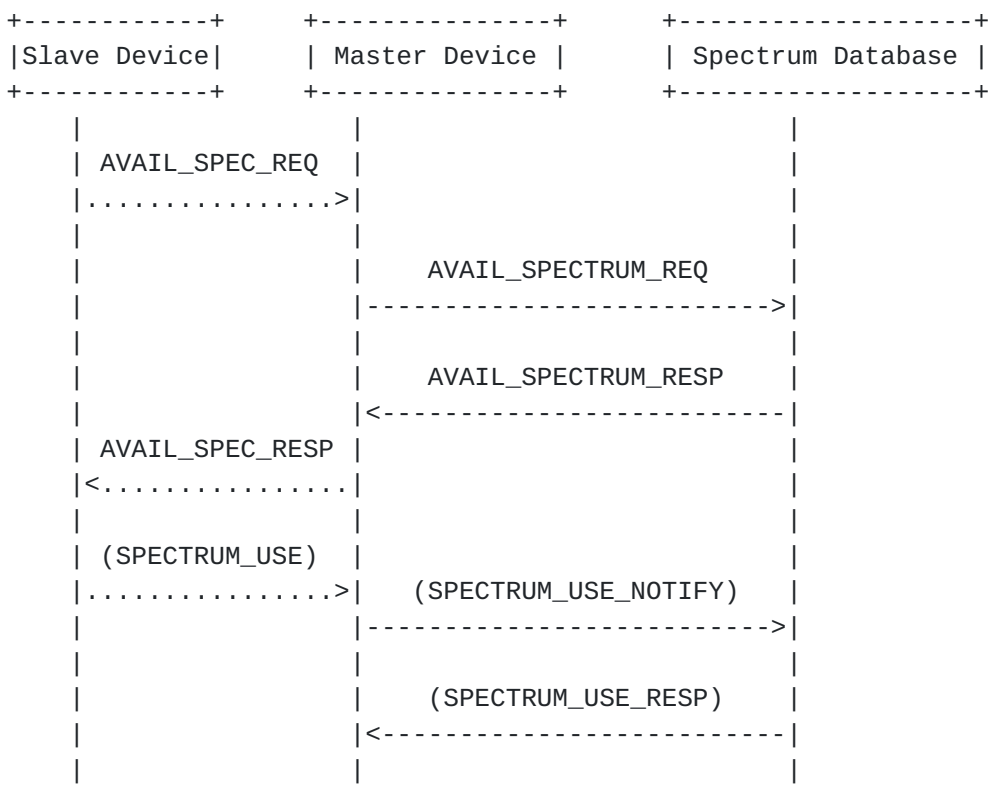


Figure 4

4.5.1. AVAIL_SPECTRUM_REQ

The request message for the Available Spectrum Query protocol MUST include a geolocation. Rulesets may mandate that it be the device's current location or allow it to be an anticipated location. A parameter marked as optional may be required by some rulesets.

AVAIL_SPECTRUM_REQ	
deviceDesc:DeviceDescriptor	see description
location:GeoLocation	see description
owner:DeviceOwner	optional
antenna:AntennaCharacteristics	optional
capabilities:DeviceCapabilities	optional
masterDeviceDesc:DeviceDescriptor	optional
masterDeviceLocation:GeoLocation	see description
requestType:string	optional
.....	.....
*other:any	optional

Parameters:

deviceDesc: The DeviceDescriptor (Section 5.2) for the device requesting available spectrum. When the request is made by a Master Device on its own behalf, the descriptor is that of the Master Device and it is REQUIRED. When the request is made on behalf of a Slave Device, the descriptor is that of the Slave Device, and it is REQUIRED if the "requestType" parameter is not specified. The deviceDesc parameter may be OPTIONAL for some values of requestType.

location: The GeoLocation (Section 5.1) for the device requesting available spectrum. This is the location at which the device intends to operate, but more precisely, the location of the radiation center of the device's antenna. When the request is made by the Master Device on its own behalf, the location is that of the Master Device and it is REQUIRED. When the request is made by the Master Device on behalf of a Slave Device, the location is that of the Slave Device, and it is OPTIONAL (see also masterDeviceLocation). The location may be an anticipated position of the device to support mobile devices, but its use depends on the ruleset. If the location specifies a region, rather than a point, the Database MAY return an error with the UNIMPLEMENTED (Table 1) code, if it does not implement query by region.

owner: The DeviceOwner (Section 5.5) information MAY be included to register the device with the Database. This enables the device to register and get spectrum-availability information in a single request. Some rulesets mandate registration for specific device types.

antenna: The AntennaCharacteristics (Section 5.3) is OPTIONAL.

capabilities: The Master Device MAY include its DeviceCapabilities (Section 5.4) to limit the available-spectrum response to the spectrum that is compatible with its capabilities. The Database SHOULD NOT return spectrum that is not compatible with the specified capabilities.

masterDeviceDesc: When the request is made by the Master Device on behalf of a Slave Device, the Master Device MAY provide its own descriptor.

masterDeviceLocation: When the request is made by the Master Device on behalf of a Slave Device, the Master Device MUST provide its own GeoLocation (Section 5.1).

requestType: The request type is OPTIONAL, which may be used to modify the request, but its use depends on the applicable ruleset. The request type may be used, for example, to indicate that the response should include generic Slave Device parameters without having to specify the device descriptor for a specific device. When requestType is missing, the request is for a specific device (Master or Slave), so deviceDesc is REQUIRED. The maximum length of the value is 64 octets. See IANA Ruleset Registry, Initial Registry Contents (Section 9.1.2) for ruleset specifics.

other: Rulesets and database implementations may require additional request parameters. The Database MUST ignore all parameters it does not understand. Consult the PAWS Parameters Registry (Section 9.2) for possible additional parameters.

4.5.2. AVAIL_SPECTRUM_RESP

The response message for the Available Spectrum Query contains one or more SpectrumSpec (Section 5.9) elements, one for each ruleset supported at the location specified in the corresponding AVAIL_SPECTRUM_REQ (Section 4.5.1) request. Each SpectrumSpec element contains a list of one or more spectrum schedules, representing permissible power levels over time:

- Each spectrum schedule specifies permissible power level for a duration defined by a pair of start and stop times. The power levels refer to permissible EIRP over a resolution bandwidth.

- Within each list of schedules, event-time intervals MUST be disjoint and MUST be sorted in increasing time.

- A gap in the time schedule means no spectrum is available for that time interval.

Consider a Database that provides a schedule of available spectrum for the next 24 hours. If spectrum availability were to be different at different times of day, the response would contain a list of schedules, each transition representing some change to the spectrum availability. A device might use different strategies to select which spectrum to use, e.g.:

- Always use the frequencies that permit the highest power
- Use the frequencies that is available for the longest period of time
- Just use the first set of frequencies that matches its needs

Parameters:

timestamp: Timestamp of the response is expressed using the form, YYYY-MM-DDThh:mm:ssZ, as defined by "Date and Time on the Internet: Timestamps" [RFC3339]. This can be used by the device as a reference for the start and stop times in the spectrum schedules.

deviceDesc: The Database MUST include the DeviceDescriptor (Section 5.2) specified in the AVAIL_SPECTRUM_REQ message.

spectrumSpecs: The SpectrumSpec (Section 5.9) list MUST include at least one entry. Each entry contains the schedules of available spectrum for a ruleset. The Database MAY return more than one SpectrumSpec to represent available spectrum for multiple rulesets at the specified location.

databaseChange: The Database MAY include a DbUpdateSpec (Section 5.7) to notify the device of a change to the Database URI, providing one or more alternate database URIs. The device needs to update its preconfigured entry for the responding database with the alternate databases listed in the DbUpdateSpec.

other: Database implementations MAY return additional parameters in the response. The device MUST ignore any parameters that it does not understand. Consult the PAWS Parameters Registry (Section 9.2) for possible additional parameters and requirements they place on the device.

4.5.2.1. Update Requirements

When the stop time specified in the schedule has been reached, the device:

- MUST obtain a new spectrum-availability schedule, either by using the next one in the list (if provided) or making another Available Spectrum Query (Section 4.5).

- If the device is unable to contact the Database to obtain a new schedule, it MUST treat this as equivalent to a response with no available spectrum.

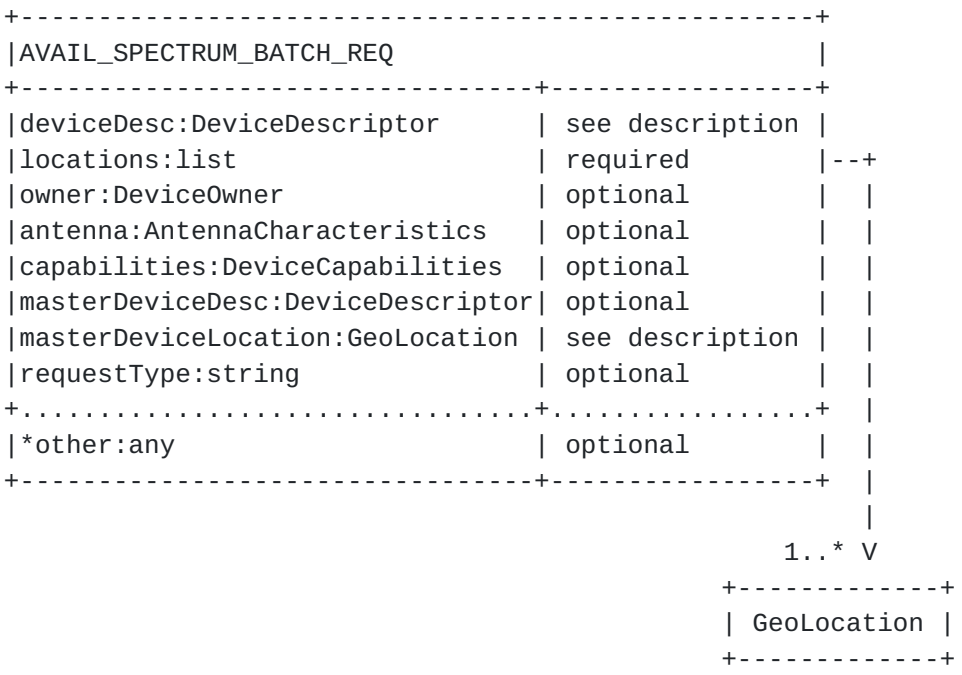
Some rulesets also mandate that a device must obtain a new specturm-availability schedule if the device moves beyond a threshold distance (established by the ruleset) away from the actual location and all anticipated location(s) it reported in previous AVAIL_SPECTRUM_REQ or AVAIL_SPECTRUM_BATCH_REQ requests (see "maxLocationChange" in RulesetInfo (Section 5.6)). If the device is unable to contact the Database to obtain a new schedule, it MUST treat this as equivalent to a response with no available spectrum.

NOTE: The ruleset determines required device behavior when spectrum is no longer available. The ruleset also governs whether a device may request and use spectrum at anticipated locations beyond the threshold distance from its current location.

4.5.3. AVAIL_SPECTRUM_BATCH_REQ

The Database MAY implement the batch request that allows multiple locations to be specified. This enables a portable Master Device, for example, to get available spectrum for a sequence of anticipated locations using a single request. The Database interprets each location in the batch request as if it were an independent request and returns results consistent with multiple individual AVAIL_SPECTRUM_REQ (Section 4.5.1) requests, but returns these results in a batched response (Section 4.5.4). The request message for the batch Available Spectrum Query protocol MUST include at least one GeoLocation (Section 5.1). If the Database does not implement batch requests, it MUST return an UNIMPLEMENTED (Table 1) error.

NOTE: Whether anticipated locations are allowed depends on the specified ruleset. A parameter marked as optional may be required by some rulesets.



Parameters:

deviceDesc: The DeviceDescriptor (Section 5.2) for the device requesting available spectrum. When the request is made by a Master Device on its own behalf, the descriptor is that of the

Master Device and it is REQUIRED. When the request is made on behalf of a Slave Device, the descriptor is that of the Slave Device, and it is REQUIRED if the "requestType" parameter is not specified. The deviceDesc parameter may be OPTIONAL for some values of requestType.

locations: The GeoLocation (Section 5.1) list for the device is REQUIRED. This allows the device to specify its actual location plus additional anticipated locations. At least one location MUST be included. This specification places no upper limit on the number of locations, but the Database MAY restrict the number of locations it supports by returning a response with fewer locations than specified in the request. If the locations specify regions, rather than points, the Database MAY return an error with the UNIMPLEMENTED (Table 1) code, if it does not implement query by region. When the request is made by a Master Device on its own behalf, the locations are those of the Master Device. When the request is made by the Master Device on behalf of a Slave Device, the locations are those of the Slave Device (see also masterDeviceLocation).

owner: The DeviceOwner (Section 5.5) information MAY be included to register the device with the Database. This enables the device to register and get spectrum-availability information in a single request. Some rulesets mandate registration for specific device types.

antenna: The AntennaCharacteristics (Section 5.3) is OPTIONAL.

capabilities: The Master Device MAY include its DeviceCapabilities (Section 5.4) to limit the available-spectrum response to the spectrum that is compatible with its capabilities. The Database SHOULD NOT return spectrum that is not compatible with the specified capabilities.

masterDeviceDesc: When the request is made by the Master Device on behalf of a Slave Device, the Master Device MAY provide its own descriptor.

masterDeviceLocation: When the request is made by the Master Device on behalf of a Slave Device, the Master Device MUST provide its own GeoLocation (Section 5.1).

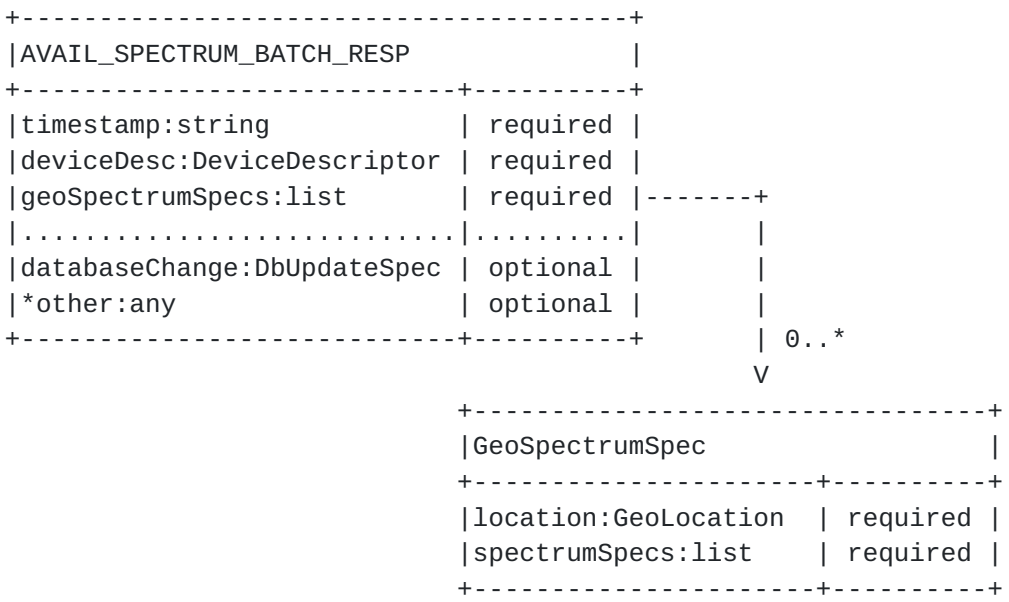
requestType: The request type is an OPTIONAL parameter that may be used to modify the request, but its use depends on applicable the ruleset. The request type may be used, for example, to request generic Slave Device parameters without having to specify the device descriptor for a specific device. When the requestType

parameter is missing, the request is for a specific device (Master or Slave), so deviceDesc is REQUIRED. The maximum length is 64 octets. See IANA Ruleset Registry, Initial Registry Contents (Section 9.1.2) for ruleset specifics.

other: Rulesets and database implementations may require additional request parameters. The Database MUST ignore all parameters it does not understand. Consult the PAWS Parameters Registry (Section 9.2) for possible additional parameters.

4.5.4. AVAIL_SPECTRUM_BATCH_RESP

The response message for the batch Available Spectrum Query contains a schedule of available spectrum for the device at multiple locations.



Parameters:

timestamp: Timestamp of the response of the form, YYYY-MM-DDThh:mm:ssZ, as defined by "Date and Time on the Internet: Timestamps" [RFC3339]. This can be used by the device as a reference for the start and stop times in the spectrum schedules.

deviceDesc: The Database MUST include the DeviceDescriptor (Section 5.2) specified in the AVAIL_SPECTRUM_BATCH_REQ message.

geoSpectrumSpecs: The geoSpectrumSpecs (Section 5.15) list is REQUIRED (although it MAY be empty if spectrum is unavailable). For each location, the Database MAY return one or more SpectrumSpecs (Section 5.9) to represent available spectrum for

one or more rulesets. The Database MAY return available spectrum for fewer locations than requested. The order of the entries in the list is not significant and the device MUST use the location value in each GeoSpectrumSpec entry to match available spectrum to a location.

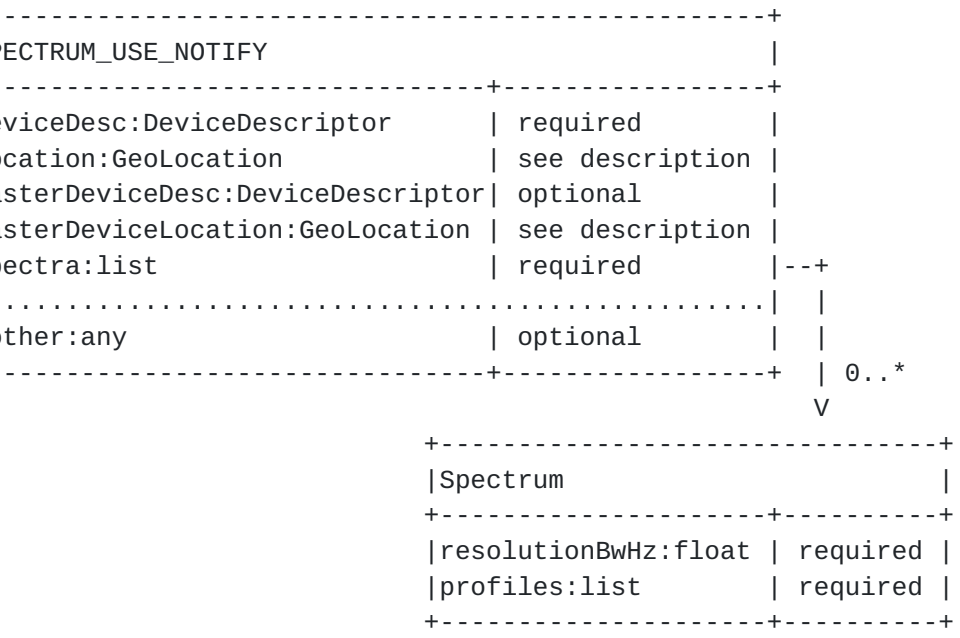
databaseChange: The Database MAY include a DbUpdateSpec (Section 5.7) to notify the device of a change to the Database URI, providing one or more alternate database URIs. The device needs to update its preconfigured entry for the responding database with the alternate databases listed in the DbUpdateSpec.

other: Database implementations MAY return additional parameters in the response. Consult the PAWS Parameters Registry (Section 9.2) for possible additional parameters and requirements they place on the device.

See Update Requirements (Section 4.5.2.1) for when the device must update its available spectrum data.

4.5.5. SPECTRUM_USE_NOTIFY

The spectrum-use notification message indicates the spectrum anticipated to be used by the device.



Parameters:

deviceDesc: The DeviceDescriptor (Section 5.2) for the device is REQUIRED.

location: The GeoLocation (Section 5.1) for the device. When the notification is made by a Master Device on its own behalf, the location is that of the Master Device and is REQUIRED. When the notification is made by a Master Device on behalf of a Slave Device, the location is that of the Slave Device and is OPTIONAL, but may be required by some rulesets.

spectra: The Spectrum (Section 5.11) list is REQUIRED, and specifies the spectrum anticipated to be used by the device, which includes profiles of frequencies and power levels. The list MAY be empty, if the device decides not to use any spectrum. For consistency, the resolution bandwidth value, "resolutionBwHz" MUST match that from one of the Spectrum (Section 5.11) elements in the corresponding AVAIL_SPECTRUM_RESP message, and the maximum power levels in the Spectrum element MUST be expressed as power (EIRP) over the specified "resolutionBwHz" value. The actual bandwidth to be used (as computed from the start and stop frequencies) MAY be different from the "resolutionBwHz" value. As an example, when the ruleset expresses maximum power spectral density in terms of maximum power over any 100 kHz band, then the "resolutionBwHz" value should be set to 100 kHz, even though the actual bandwidth used can be 20 kHz.

masterDeviceDesc: When the notification is made by the Master Device on behalf of a Slave Device, the Master Device MAY provide its own descriptor.

masterDeviceLocation: When the notification is made by the Master Device on behalf of a Slave Device, the Master Device MUST include its own GeoLocation (Section 5.1).

other: Depending on the ruleset, other parameters may be required. To simplify its logic, the device MAY include the union of all parameters required by all supported rulesets. The Database MUST ignore all parameters it does not understand.

4.5.6. SPECTRUM_USE_RESP

The spectrum-use response message simply acknowledges receipt of the notification.

```
+--------------------------------------+
|SPECTRUM_USE_RESP                     |
+----------------------------+---------+
|databaseChange:DbUpdateSpec | optional |
|....................................|
|*other:any                  | optional |
+----------------------------+---------+
```

Parameters:

databaseChange: The Database MAY include a DbUpdateSpec (Section 5.7) to notify the device of a change to the Database URI, providing one or more alternate database URIs. The device needs to update its preconfigured entry for the responding database with the alternate databases listed in the DbUpdateSpec.

other: Database implementations MAY return additional parameters in the response. Consult the PAWS Parameters Registry (Section 9.2) for possible additional parameters.

4.6. Device Validation

A Slave Device needs a Master Device to ask the Database on its behalf for available spectrum. Depending on the ruleset, the Master Device also must validate with the Database that the Slave Device is permitted to operate. When the ruleset allows a Master Device to "cache" the available spectrum for a period of time, the Master Device may use the simpler Device Validation component, instead of the full Available Spectrum Query component, to validate a Slave Device.

When validating one or more Slave Devices, the Master Device sends the Database a request that includes the device identifier -- and any other parameters required by the ruleset -- for each Slave Device. The Database MUST return a response with an entry for each device to indicate whether it is permitted to use the spectrum.

A typical sequence for using the Device Validation request is illustrated in Figure 5, where the Master Device already has a valid set of available spectrum for Slave Devices. Note that the communication and protocol between the Slave Device and Master Device is outside the scope of this document.

- DEV_VALID_REQ (Section 4.6.1) is the device-validation request message
- DEV_VALID_RESP (Section 4.6.2) is the device-validation response message

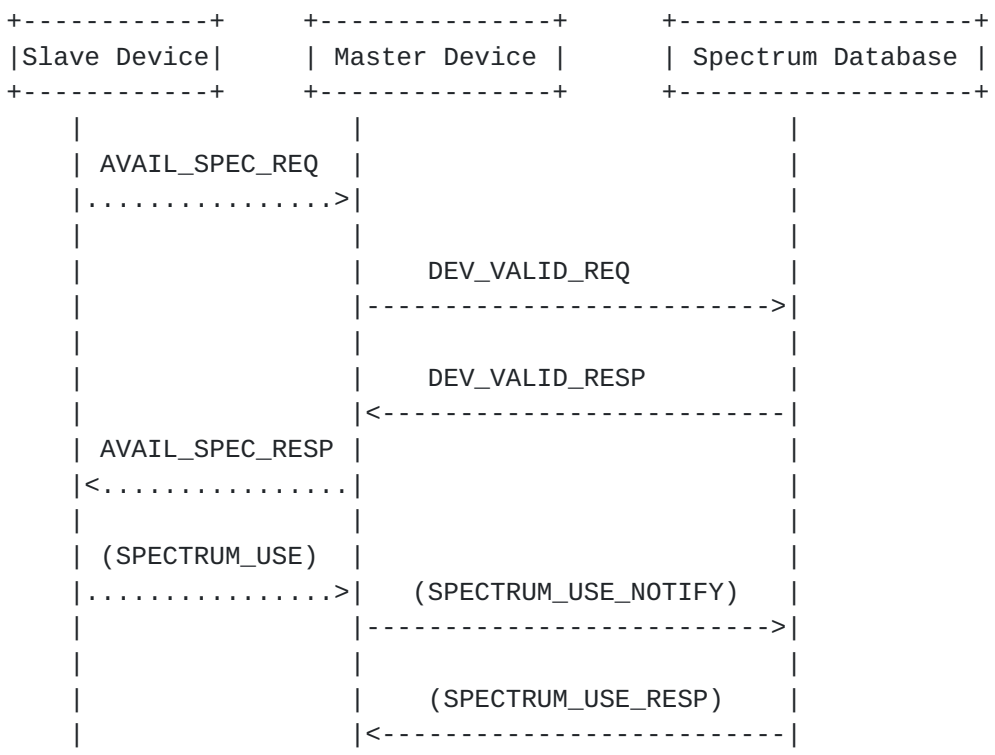
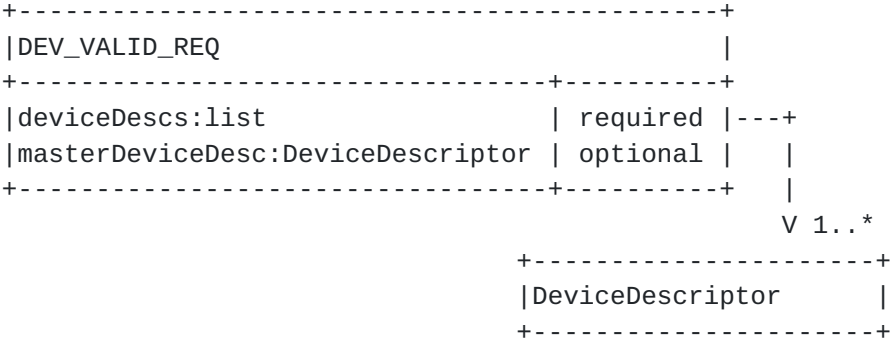


Figure 5

4.6.1. DEV_VALID_REQ

This request is used by a Master Device to determine which Slave Devices are permitted to operate.



Parameters:

deviceDescs: A DeviceDescriptor (Section 5.2) list is REQUIRED, which specifies the list of Slave Devices that are to be validated.

masterDeviceDesc: The Master Device MAY provide its own descriptor.

4.6.2. DEV_VALID_RESP

Parameters:

deviceValidities: A DeviceValidities (Section 5.16) list is REQUIRED to report the list of Slave Devices and whether each listed device is valid. The number of entries MUST match the number of DeviceDescriptors (Section 5.2) listed in the DEV_VALID_REQ message.

databaseChange: The Database MAY include a DbUpdateSpec (Section 5.7) to notify the device of a change to the Database URI, providing one or more alternate database URIs. The device needs to update its preconfigured entry for the responding database with the alternate databases listed in the DbUpdateSpec.

5. Protocol Parameters

This section presents more details of the parameters that make up the PAWS request and response messages. It also includes a sub-section defining response codes.

5.1. GeoLocation

GeoLocation is used to specify one of the following:

- A single point with optional uncertainty
- A region described by a polygon

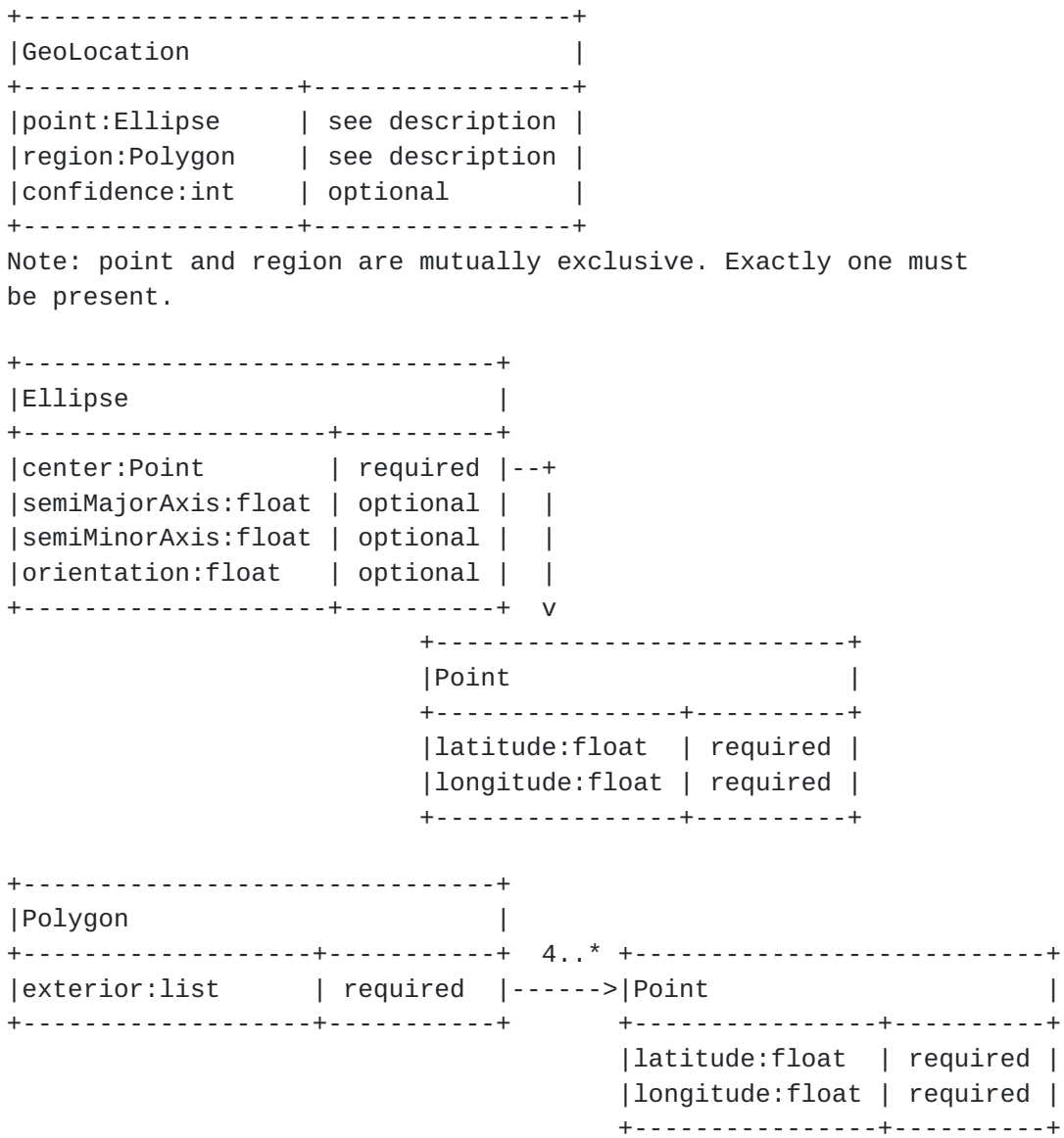
These are represented using geometric shapes defined in Section 5 of "GEOPRIV Presence Information Data Format Location Object" [RFC5491], where:

o A "point" with uncertainty is represented using the Ellipse shape

o A region is represented using the Polygon shape

The coordinates are expressed using the WGS84 datum [WGS-84], and units are degrees or meters. GeoLocation MAY also include a confidence level, expressed as a percentage. The confidence and uncertainty parameters may be required by some rulesets (see also Uncertainty and Confidence [I-D.ietf-geopriv-uncertainty]).

The data model for GeoLocation is illustrated below:



Note: point and region are mutually exclusive. Exactly one must be present.

Parameters:

point: If present, it specifies the GeoLocation as a point. Paradoxically, a "point" is parameterized using an Ellipse, where the center represents the location of the point and the distances along the major and minor axes represent the uncertainty. The uncertainty values may be required, depending on the ruleset. Exactly one of "point" or "region" MUST be present.

region: If present, it specifies the GeoLocation as a region. Exactly one of "point" or "region" MUST be present.

center: The center refers to the location of a GeoLocation point and is represented as the center of an ellipse.

latitude, longitude: Floating-point numbers that express the latitude and longitude in degrees using the WGS84 datum [WGS-84].

semiMajorAxis, semiMinorAxis: This OPTIONAL parameter expresses the location uncertainty, expressed in meters. It is parameterized using distances along the major and minor axes of the ellipse. The default value for each parameter is 0.

orientation: This defines the orientation of the ellipse, expressed as the rotation, in degrees, of the semi-major axis from North towards the East. For example, when the uncertainty is greatest along the North-South direction, orientation is 0 degrees; conversely, if the uncertainty is greatest along the East-West direction, orientation is 90 degrees. When orientation is not present, the default value is 0.

exterior: When GeoLocation describes a region, the "exterior" parameter refers to a list of latitude/longitude points that represents the vertices of a polygon. The first and last points MUST be the same. Thus, a minimum of 4 points is required. The following polygon restrictions from [RFC5491] apply:

* A connecting line MUST NOT cross another connecting line of the same polygon.

* The vertices MUST be defined in a counter-clockwise direction, looking at them from above.

* The edges of a polygon are defined by the shortest path between two points in space (not a geodesic curve). Consequently, the length between two adjacent vertices SHOULD be restricted to a maximum of 130 km.

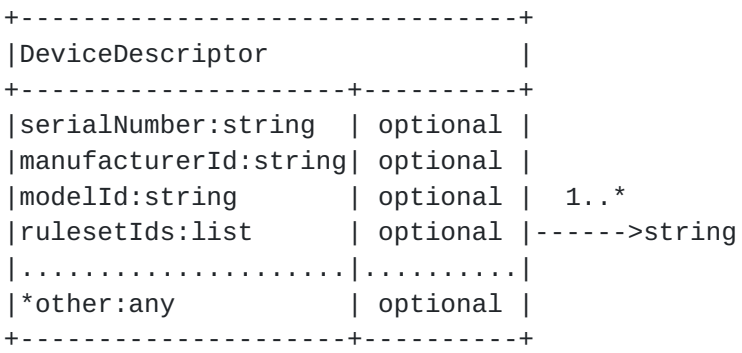
* All vertices are assumed to be at the same altitude.

* Polygon shapes SHOULD be restricted to a maximum of 15 vertices (16 points that includes the repeated vertex).

confidence: The location confidence level, as a percentage, MAY be provided. When this parameter is not provided, the default value is 95. Valid values range from 0 to 100, but, in practice, 100-percent confidence is not achievable. The confidence value is meaningful only when GeoLocation refers to a point with uncertainty.

5.2. DeviceDescriptor

The device descriptor contains parameters that identify the specific device, such as its manufacturer serial number, manufacturer's ID, and any other device characteristics required by ruleset.



Parameters:

serialNumber: The manufacturer's device serial number is OPTIONAL, although rulesets typically require it. Its maximum length is 64 octets.

manufacturerId: The manufacturer's ID is OPTIONAL, but may be required by some rulesets. This represents the name of the device manufacturer, and therefore ought to be consistent across all devices from the same manufacturer and distinct from that of other manufacturers. Its maximum length is 64 octets.

modelId: The device's model ID is OPTIONAL, but may be required by some rulesets. Its maximum length is 64 octets.

rulesetIds: The list of identifiers for rulesets supported by the device (see Ruleset ID Registry (Section 9.1)). A Database MAY require that the device provides this list before servicing the device requests. If the Database supports none of the rulesets specified in the list, the Database MAY refuse to service the device requests. See RulesetInfo (Section 5.6) for discussion on

ruleset identifier. If present, the list MUST contain at least one entry.

other: Depending on the ruleset, other parameters may be required. The Database MUST ignore all parameters in the message it does not understand. See PAWS Parameters Registry (Section 9.2) for additional valid parameters and for the process for extending the message with more parameters. Additionally, see PAWS Ruleset ID Registry (Section 9.1) for the valid set of parameters for each ruleset.

5.3. AntennaCharacteristics

Antenna characteristics provide additional information, such as the antenna height, antenna type, etc. Whether antenna characteristics must be provided in a request depends on the device type and ruleset. Additionally, a parameter marked as optional may be required by some rulesets.

AntennaCharacteristics	
height:float	optional
heightType:enum	optional
heightUncertainty:float	optional
*characteristics: various	optional

Parameters:

height: The antenna height in meters. Note that the height may be negative.

heightType: Valid values are:

AGL Above ground level (default)

AMSL Above mean sea level

heightUncertainty: The height uncertainty in meters.

NOTE: Depending on the ruleset, additional antenna characteristics may be required, such as:

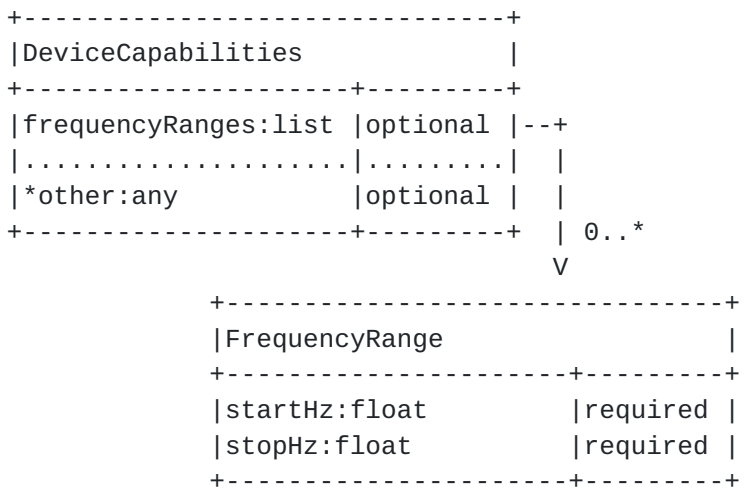
o antenna direction

- antenna radiation pattern
- antenna gain
- antenna polarization

These are not defined by the base protocol, but may be added to the PAWS Parameters Registry (Section 9.2), as needed.

5.4. DeviceCapabilities

Device capabilities provide additional information that may be used by the device to provide additional information to the Database that can help it to determine available spectrum. If the Database does not support device capabilities it MUST ignore the parameter altogether.



Parameters:

frequencyRanges: Optional FrequencyRange (Section 5.13) list. Each FrequencyRange element contains start and stop frequencies in which the device can operate. When specified, the Database SHOULD NOT return available spectrum that falls outside these ranges.

other Consult the PAWS Parameters Registry (Section 9.2) for possible additional parameters. The Database MUST ignore all parameters it does not understand.

5.5. DeviceOwner

DeviceOwner contains information on device ownership that is provided as part of device registration. Some rulesets may require additional parameters.

```
+-----------------------------+
|DeviceOwner                  |
+-----------------+-----------+
|owner:vcard      | required  |
|operator:vcard   | optional  |
+-----------------+-----------+
```

Parameters:

owner: The vCard contact information for the individual or business that owns the device is REQUIRED.

operator: The vCard contact information for the device operator is OPTIONAL, but may be required by specific rulesets.

See PAWS Ruleset ID Registry (Section 9.1) for ruleset-specific requirements on mandatory vCard properties. Depending on the ruleset, the Database may be required to validate the device-owner information. In these cases, the Database MUST respond with an INVALID_VALUE error (see Error Codes (Section 5.17)) if validation fails.

All contact information MUST be expressed using the structure defined by the "vCard Format Specification" [RFC6350], encoded in JSON [RFC7095]. Note that the vCard specification defines maximum lengths for each parameter, conforming to X.520 [ITUT.X520.2008] recommendations.

5.6. RulesetInfo

RulesetInfo contains parameters for the ruleset of a regulatory domain that is communicated using the Initialization component (Section 4.3), Device Registration (Section 4.4), and Available Spectrum Query (Section 4.5) components.

```
+-----------------------------------------+
|RulesetInfo                              |
+-----------------------------------------+
|authority:string        | required       |
|rulesetId:string        | required       |
|maxLocationChange:float | see description |
|maxPollingSecs:int      | see description |
|.........................................|
|*other:any              | optional       |
+------------------------+----------------+
```

Parameters:

authority: A string that indicates the regulatory domain to which the ruleset applies is REQUIRED. It will normally be a 2-letter country code defined by Country Codes - ISO 3166 [ISO3166-1].

rulesetId: The ID of a ruleset for the specified authority (see Ruleset ID Registry (Section 9.1)). The device can use this to determine additional device behavior required by the associated ruleset. To define new ruleset IDs, see Defining Ruleset Identifiers (Section 8.1).

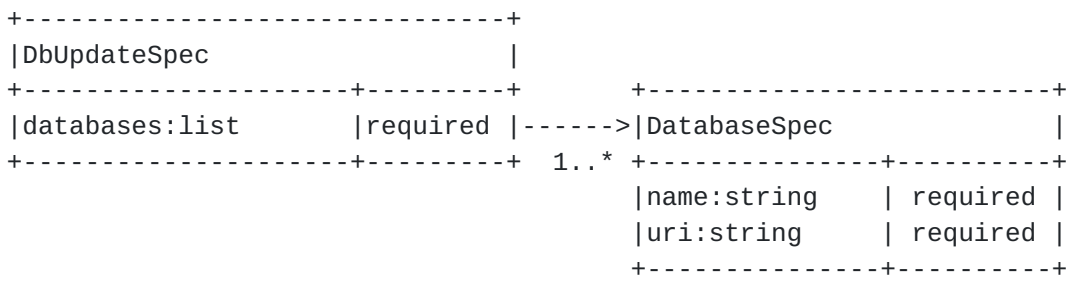
maxLocationChange: The maximum location change in meters is REQUIRED for Initialization Response (Section 4.3.2), but OPTIONAL otherwise. Some regulatory domains mandate that, when the device changes location by more than this specified distance, it contact the Database to get the available spectrum for the new location. If this value is provided by the Database within the context of an Available Spectrum Response (Section 4.5.2), it takes precedence over the value within the Initialization Response (Section 4.3.2).

maxPollingSecs: The maximum duration, in seconds, between requests for available spectrum is REQUIRED for the Initialization Response (Section 4.3.2), but OPTIONAL otherwise. The device MUST contact the Database to get available spectrum no less frequently than this duration. If this value is provided within the context of an Available Spectrum Response (Section 4.5.2), it takes precedence over the value within the Initialization Response (Section 4.3.2).

other: Depending on the ruleset, other parameters may be required. Devices MUST ignore all parameters in the message it does not understand. Consult the PAWS Parameters Registry (Section 9.2) for possible additional parameters.

5.7. DbUpdateSpec

This element is provided by the Database to notify devices of an upcoming change to the Database URI.

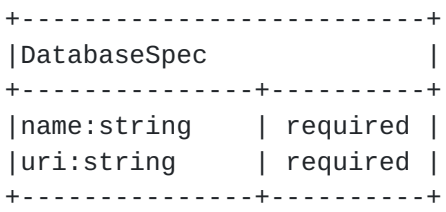


Parameters:

databases: List of one or more DatabaseSpec (Section 5.8) entries. A device needs to update its preconfigured entry for the responding database with the alternate databases listed in the DbUpdateSpec.

5.8. DatabaseSpec

This element contains the name and URI of a database.



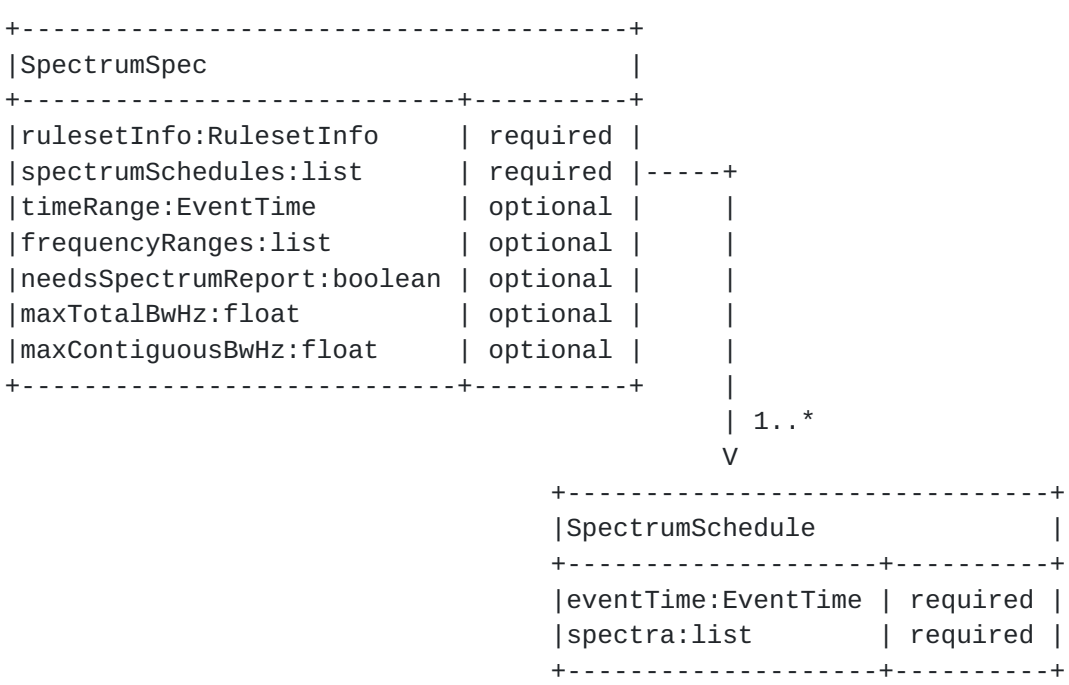
Parameters:

name: The display name. Its maximum length is 64 octets.

uri: The corresponding URI of the database. Its maximum length is 1024 octets.

5.9. SpectrumSpec

The SpectrumSpec element encapsulates the schedule of available spectrum for a ruleset.



Parameters:

rulesetInfo: RulesetInfo (Section 5.6) is REQUIRED to identify the regulatory domain and ruleset to which the spectrum schedule applies (see Ruleset ID Registry (Section 9.1)). The device needs to use the corresponding ruleset to interpret the response. Values provided within rulesetInfo, such as maxLocationChange, take precedence over the values provided by the Initialization Procedure (Section 4.3).

spectrumSchedules: The SpectrumSchedule (Section 5.10) list is REQUIRED. At least one schedule MUST be included. More than one schedule MAY be included to represent future changes to the available spectrum. How far in advance a schedule may be provided depends on the ruleset. If more than one schedule is included, the eventTime intervals MUST be disjoint and MUST be sorted in increasing time. A gap in the time schedule indicates no available spectrum during that time-interval gap.

timeRange: The time range for which the specification is comprehensive is OPTIONAL. When specified, any gaps in time intervals within the "spectrumSchedules" element that overlaps with the range specified by "timeRange" are interpreted by the device as time intervals in which there is no available spectrum.

frequencyRanges: The frequency ranges for which the specification is comprehensive is OPTIONAL. It is a list of disjoint FrequencyRange (Section 5.13) entries. When specified, it typically corresponds to the frequency ranges governed by the ruleset, e.g., for TV whitespace, the frequency ranges can correspond to the VHF and UHF bands of the associated regulatory domain. A device can combine this information with the available-spectrum specification within the "spectrumSchedules" element to distinguish between "unavailable spectrum" and "spectrum for which no information has been provided".

needsSpectrumReport: The Database MAY return true for this parameter if spectrumSchedules list is non-empty; otherwise, the Database MAY omit this parameter altogether, in which case, the default value is false. If this parameter is present and its value is true, the device sends a SPECTRUM_USE_NOTIFY (Section 4.5.5) message to the Database; otherwise, the device SHOULD NOT send the SPECTRUM_USE_NOTIFY message. Some rulesets mandate this value to be true.

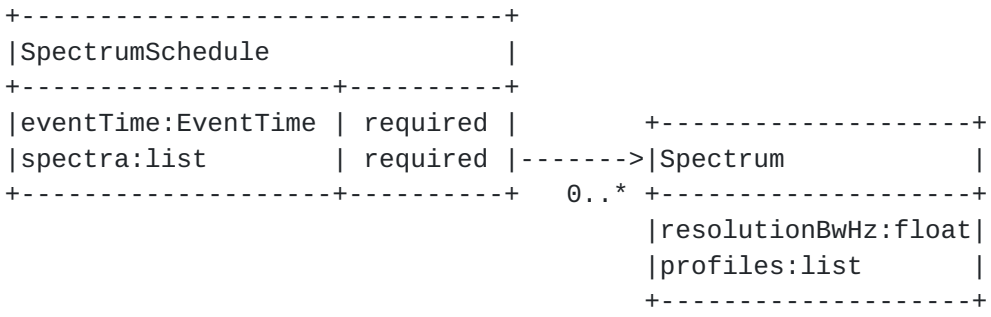
maxTotalBwHz: The Database MAY return a constraint on the maximum total bandwidth (in Hertz) allowed, which may or may not be contiguous. Some rulesets mandate the Database to return this

parameter. When present in the response, the device needs to apply this constraint to its spectrum-selection logic to ensure total bandwidth does not exceed this value.

maxContiguousBwHz: The Database MAY return a constraint on the maximum contiguous bandwidth (in Hertz) allowed. Some rulesets mandate the Database to return this parameter. When present in the response, the device needs to apply this constraint to its spectrum-selection logic to ensure no single block of spectrum has bandwidth that exceeds this value.

5.10. SpectrumSchedule

The SpectrumSchedule element combines EventTime (Section 5.14) with Spectrum (Section 5.11) to define a time period in which the spectrum is valid.



Parameters:

eventTime: The EventTime (Section 5.14) is REQUIRED to express "when" this specification is valid.

spectra: The Spectrum (Section 5.11) list is REQUIRED to specify the available spectrum and permissible power levels, one per resolutionBwHz. The list MAY be empty when there is no available spectrum.

5.11. Spectrum

Available spectrum can be characterized by an ordered list of spectrum profiles that defines permissible power levels over a set of frequency ranges. Each Spectrum element defines permissible power levels as maximum power spectral densities over a specified resolution bandwidth, "resolutionBwHz". Note that the spectrum profiles represent the "availability mask", as defined by the governing rule set; they are not intended to encode device-level transmission-mask requirements.

- To support a ruleset that defines different "wide-band" and "narrow-band" power levels, PAWS allows multiple Spectrum elements to be included in the available-spectrum response, each with a different resolution bandwidth.

- When multiple Spectrum elements are included in the response, each represents a constraint that the device must satisfy (logical AND).

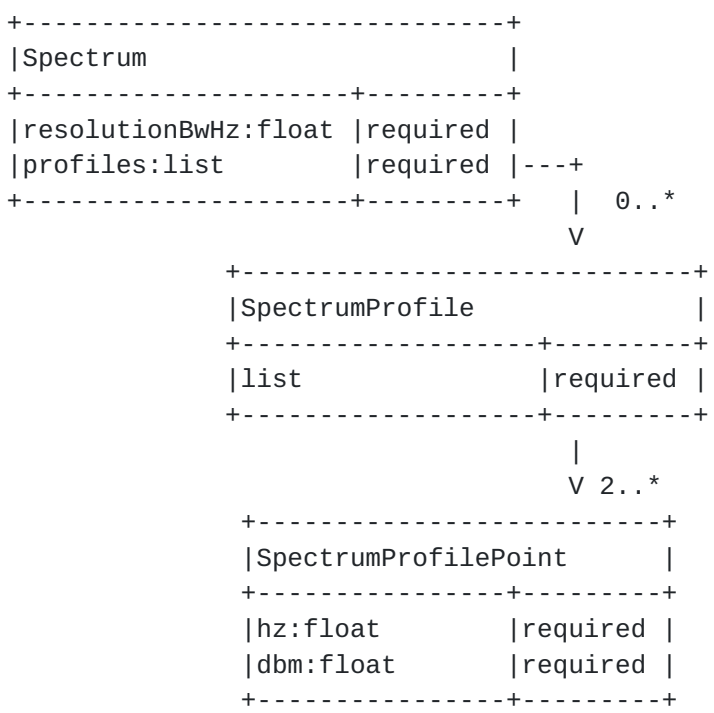
- Each Spectrum element covers the range of frequencies governed by a ruleset, rather than splitting the frequencies across multiple Spectrum elements for the same resolution bandwidth.

- Each spectrum profile represents the maximum permissible power spectral density over a contiguous range of frequencies.

- When multiple spectrum profiles are included, they MUST be disjoint and MUST be ordered in non-decreasing frequency value.

- Gaps in frequencies between consecutive spectrum profiles represent unavailability for those frequencies.

The following figure illustrates the Spectrum element and the SpectrumProfile list.



Parameters:

resolutionBwHz: This parameter defines the resolution bandwidth (in Hertz) over which permissible power spectral density is defined. For example, FCC regulation would require one spectrum specification at a bandwidth of 6MHz, and ETSI regulation would require two specifications, at 0.1MHz and 8MHz.

profiles: A SpectrumProfile (Section 5.12) list specifies permissible power levels over a set of frequency ranges. The list MAY be empty if there is no available spectrum.

The following example shows permitted power spectral densities for a single resolution bandwidth of 6MHz (for illustrative purposes only):

```
[
  {
    "resolutionBwHz": 6e6,
    "profiles": [
      [
        {"hz": 5.18e8, "dbm": 30.0},
        {"hz": 5.30e8, "dbm": 30.0}
      ],
      ...
    ]
  }
]
```

This is interpreted as:

- Over any 6MHz within the frequency range, [518MHz, 530MHz), maximum permitted power is 30.0dBm (1000mW)

Consider now an example with two different sets of permitted power spectral densities for the same set of frequencies over different resolution bandwidths (for illustrative purposes only):

```
[
  {
    "resolutionBwHz": 6e6,
    "profiles": [
      [
        {"hz": 5.18e8, "dbm": 30.0},
        {"hz": 5.30e8, "dbm": 30.0}
      ],
      ...
    ]
  },
  {
    "resolutionBwHz": 1e5,
    "profiles": [
      [
        {"hz": 5.18e8, "dbm": 27.0},
        {"hz": 5.30e8, "dbm": 27.0}
      ],
      ...
    ]
  }
]
```

This is interpreted as:

- Over any 6MHz within the frequency range, [518MHz, 530MHz), maximum permitted power is 30.0dBm (1000mW), and

- Over any 100 kHz within the frequency range, [518MHz, 530MHz), maximum permitted power is 27.0dBm (500mW)

This would allow, for example, operating two 100kHz sub-channels within the indicated 12MHz range at 500mW each, totaling 1000mW. Of course, many combinations are possible, as long as they satisfy both conditions.

The following example encodes multiple (two) spectrum profiles, each having a gap from 530 MHz to 536 MHz (for illustrative purposes only):

```
[
  {
    "resolutionBwHz": 6e6,
    "profiles": [
      [
        {"hz": 5.18e8, "dbm": 30.0},
        {"hz": 5.24e8, "dbm": 30.0},
        {"hz": 5.24e8, "dbm": 36.0},
        {"hz": 5.30e8, "dbm": 36.0}
      ],
      [
        {"hz": 5.36e8, "dbm": 30.0},
        {"hz": 5.42e8, "dbm": 30.0}
      ],
      ...
    ]
  },
  {
    "resolutionBwHz": 1e5,
    "profiles": [
      [
        {"hz": 5.18e8, "dbm": 27.0},
        {"hz": 5.24e8, "dbm": 27.0},
        {"hz": 5.24e8, "dbm": 30.0},
        {"hz": 5.30e8, "dbm": 30.0}
      ],
      [
        {"hz": 5.36e8, "dbm": 27.0},
        {"hz": 5.42e8, "dbm": 27.0}
      ],
      ...
    ]
  }
]
```

5.12. SpectrumProfile

A spectrum profile is characterized by an ordered list of (frequency, power) points that represents the shape of maximum permissible power levels over a range of frequencies as a piecewise linear curve.

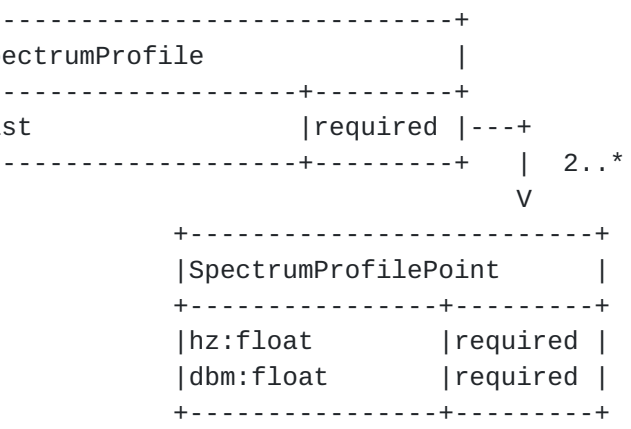
- It MUST contain a minimum of two entries.
- The entries in the list MUST be ordered in non-decreasing frequency values.
- Two consecutive points MAY have the same frequency value to represent a "step function".

o Three or more points MUST NOT share the same frequency value.

o The first frequency is inclusive; the last frequency is exclusive.

NOTE: This encoding allows presentation of "ramps" where the slope of a line segment may be finite and non-zero.

The following figure illustrates the SpectrumProfile element.



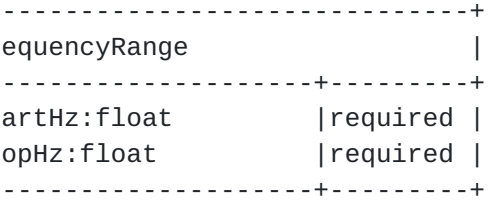
Parameters of each point in the profile:

hz: The frequency, in Hertz, at which the power level is defined.

dbm: The power level, expressed as dBm per resolution bandwidth, as defined by the "resolutionBwHz" element of the enclosing Spectrum (Section 5.11) element.

5.13. FrequencyRange

FrequencyRange specifies a frequency range.



Parameters:

startHz: The inclusive start of the frequency range (in Hertz) is REQUIRED.

stopHz: The exclusive end of the frequency range (in Hertz) is REQUIRED.

5.14. EventTime

The EventTime element specifies the start and stop times of an "event". This is used to indicate the time period for which a Spectrum (Section 5.11) is valid.

```
+--------------------------+
|EventTime                 |
+-----------------+--------+
|startTime:string |required |
|stopTime:string  |required |
+-----------------+--------+
```

Parameters:

startTime: The inclusive start of the event is REQUIRED.

stopTime: The exclusive end of the event is REQUIRED.

Both times are expressed using the format, YYYY-MM-DDThh:mm:ssZ, as defined by "Date and Time on the Internet: Timestamps" [RFC3339]. The times MUST be expressed using UTC.

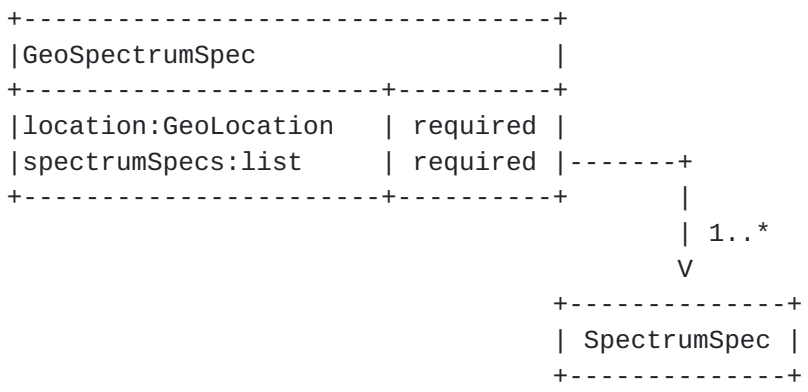
A device that does not have access to the current date and time MUST use the timestamp at the top-level of the response message as a substitute for the current time (see Available Spectrum Response (Section 4.5.2) and Available Spectrum Batch Response (Section 4.5.4)). E.g.,

- (startTime - timestamp) gives the duration that a device must wait before the event becomes "active". If the value is zero or negative, the event is already active.

- If the event is already active, (stopTime - timestamp) is the duration that the event remains active. If the value is zero or negative, the event is no longer active and MUST be ignored.

5.15. GeoSpectrumSpec

The GeoSpectrumSpec element encapsulates the available spectrum for a location. It is returned within a AVAIL_SPECTRUM_BATCH_RESP (Section 4.5.4) batch response that contains multiple GeoSpectrumSpec entries, each matching a location provided in the batch request.



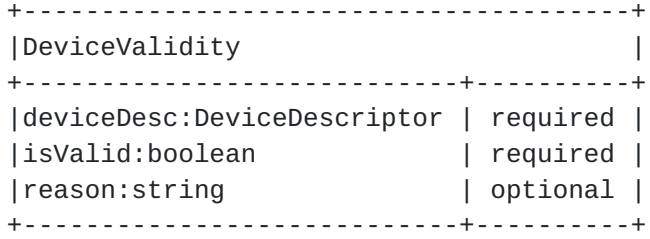
Parameters:

location: The GeoLocation (Section 5.1) identifies the location at which the spectrum schedule applies.

spectrumSpecs: The SpectrumSpec (Section 5.9) list is REQUIRED. At least one entry MUST be included. Each entry represents schedules of available spectrum for a ruleset. More than one entry MAY be included to support multiple rulesets at a location.

5.16. DeviceValidity

The DeviceValidity element is used to indicate whether a device is valid. See Section 4.6.2.



Parameters:

deviceDesc: The DeviceDescriptor (Section 5.2) that was used to check for validity is REQUIRED.

isValid: This is a REQUIRED boolean value that indicates whether the device is valid.

reason: If the device identifier is not valid, the Database MAY include a reason. The reason MAY be in any language. Its maximum length is 128 octets.

5.17. Error Element

If the Database responds to a PAWS request message with an error, it MUST include an Error element.

Error	
code:int	required
message:string	optional
data:any	see description

Parameters:

code: An integer code that indicates the error type is REQUIRED. Values MUST be within the range, -32768 to 32767, inclusive.

message: A description of the error is OPTIONAL. It MAY be in any language. Its maximum length is 128 octets.

data: The Database MAY include additional data. For some errors, additional data may be required (see Table 1). The device MUST ignore any data parameters it does not understand.

The following table lists predefined and reserved error codes. They are loosely grouped into the following categories:

-100s: Indicates compatibility issues, e.g., version mismatch, unsupported or unimplemented features.

-200s: Indicates that the device request contains an error that needs to be modified before making another request.

-300s: Indicates authorization-related issues.

Values that are not defined explicitly in the Error Codes Table (Table 1) below are unassigned. To define new error codes, see PAWS Error Code Registry (Section 9.3).

Code	Name	Description & Additional parameters
0	(reserved)	
-100	(reserved)	
-101	VERSION	The Database does not support the specified version of the message. This error does not use any additional data.
-102	UNSUPPORTED	The Database does not support the device. For

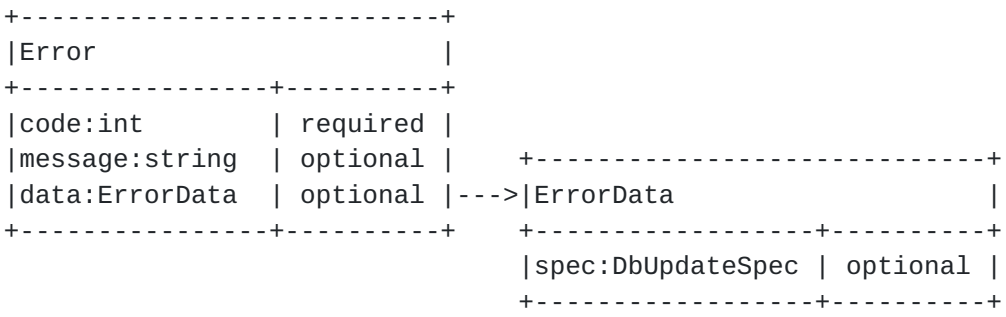
		example, it supports none of the ruleset specified in the request or does not support the device, based on its device type, model, etc. This error does not use any additional data.
-103	UNIMPLEMENTED	The Database does not implement the optional request or optional feature. This error does not use any additional data.
-104	OUTSIDE_COVERAGE	The specified geolocation is outside the coverage area of the Database. The Database MAY include a DbUpdateSpec (Section 5.7) to provide a list of alternate databases that might be appropriate for the requested location. See OUTSIDE_COVERAGE Error (Section 5.17.1) for more details.
-105	DATABASE_CHANGE	The Database has changed its URI. The Database MAY include a DbUpdateSpec (Section 5.7) in the error response to provide devices with one or more alternate database URIs. The device needs to update its preconfigured entry for the responding database with the alternate databases listed in the DbUpdateSpec. See DATABASE_CHANGE Error (Section 5.17.2) for more details.
-200	(reserved)	
-201	MISSING	A required parameter is missing. The Database MUST include a list of the required parameter names. The Database MAY include only names of parameters that are missing, but MAY include a full list. Including the full list of missing parameters may reduce the number of re-queries from the device. See MISSING Error (Section 5.17.3) for more details.
-202	INVALID_VALUE	A parameter value is invalid in some way. The Database SHOULD include a message indicating which parameter and why its value is invalid. This error does not use any additional data.
-300	(reserved)	
-301	UNAUTHORIZED	The device is not authorized to used the Database. Authorization may be determined by the ruleset or be dependent on prior arrangement between the device and Database. This error does not use any additional data.
-302	NOT_REGISTERED	Device registration required, but the device is not registered. This error does not use any additional data.
-32000 to	(reserved)	Reserved for JSON-RPC error codes.

-32768

Table 1: Error Codes

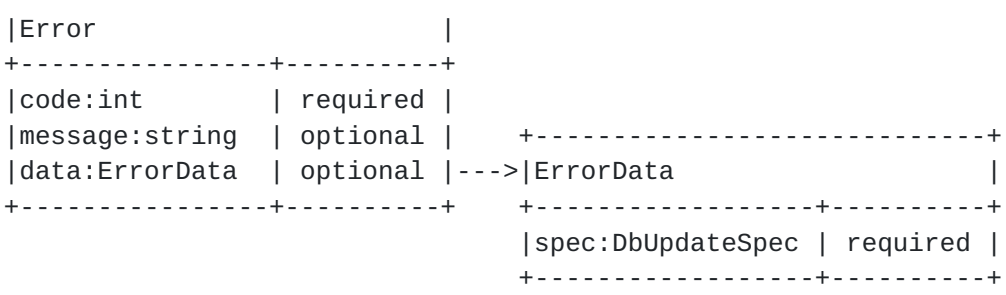
5.17.1. OUTSIDE_COVERAGE Error

When the error code is OUTSIDE_COVERAGE, the Database MAY include an ErrorData element within its Error response as the "data" parameter, and, if present, the ErrorData contains a DbUpdateSpec (Section 5.7) element that provides a list of alternate databases that might be appropriate for the requested location.



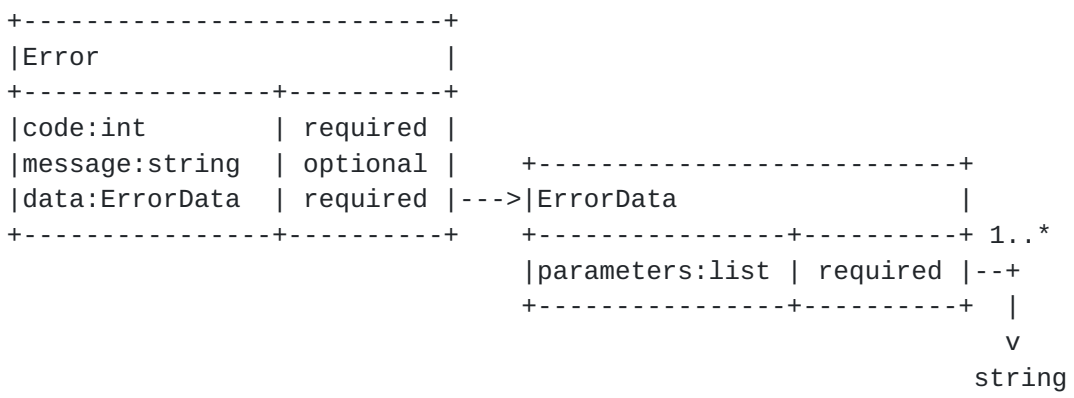
5.17.2. DATABASE_CHANGE Error

When the error code is DATABASE_CHANGE, the Database MAY include an ErrorData element within its Error response as the "data" parameter, and, if present, the ErrorData contains a DbUpdateSpec (Section 5.7) element that provides a list of alternate databases.



5.17.3. MISSING Error

When the error code is MISSING, the Database MUST include an ErrorData element within its Error response as the "data" parameter, and the ErrorData element MUST include a list of the missing required parameters and MAY include the list of all required parameters.



Parameters:

parameters: List of one or more parameter names (strings). The name of a parameter is expressed using dotted notation, when appropriate, e.g., "deviceDesc.serialNumber".

6. Message Encoding

PAWS is encoded using JSON-RPC [JSON-RPC] (see also "The JavaScript Object Notation (JSON) Data Interchange Format" [RFC7159]). Each component described in Protocol Functionalities (Section 4) corresponds to one or more JSON-RPC methods. This section discusses how to encode the data models presented in Section 4 and Section 5 into JSON and provides some example encodings. The JSON examples may contain ellipses (...) to represent additional properties or elements that have been omitted in order to make the examples more concise.

6.1. JSON-RPC Binding

The JSON-RPC [JSON-RPC] protocol consists of two basic objects, Request and Response:

- The JSON-RPC Request object encapsulates a PAWS functionality (operation) and the request message

- The JSON-RPC Response object encapsulates a PAWS response message and Error element

The Database and device MUST support JSON-RPC 2.0 encoding, with the restriction that the "id" parameter in the messages MUST be a string. The device should generate the "id" uniquely enough to allow the use of JSON-RPC batch.

The JSON-RPC Request for PAWS has the following form:

```
{
  "jsonrpc": "2.0",
  "method": "spectrum.paws.methodName",
  "params": <PAWS_REQ>,
  "id": "idString"
}
```

where "method" is the name of a PAWS functionality (operation), and <PAWS_REQ> represents one of the PAWS request messages associated with the method (see Section 4.3 through Section 4.6). Method names are defined with the prefix, "spectrum.paws.".

The non-error JSON-RPC Response for PAWS has the following form:

```
{
  "jsonrpc": "2.0",
  "result": <PAWS_RESP>,
  "id": "idString"
}
```

where <PAWS_RESP> represents one of the PAWS response messages associated with the method, and "id" is copied from the request.

The error JSON-RPC Response for PAWS has the following form:

```
{
  "jsonrpc": "2.0",
  "error": {
    "code": -102,
    "message": "An appropriate error message.",
    "data": { ... }
  },
  "id": "idString"
}
```

where the "error" object corresponds to the Error Element (Section 5.17), and "code" is an error code described in the same section. The Database SHOULD attempt to use the most specific applicable PAWS error code. When an accurate one is not available, it SHOULD fall back to standard JSONRPC error codes as defined in JSONRPC specification. For example, if the Database receives invalid JSON from the device, it should respond with "-32700", signifying a parse error. As a last resort, the Database MAY send a suitable HTTP 5xx response.

6.1.1. Method Names

Table 2 defines the method name, request object, and response object for each functionality defined in Protocol Functionalities (Section 4).

Method Name	Request	Response
spectrum.paws.init	INIT_REQ (Section 4.3.1)	INIT_RESP (Section 4.3.2)
spectrum.paws.register	REGISTRATION_REQ (Section 4.4.1)	REGISTRATION_RESP (Section 4.4.2)
spectrum.paws.getSpectrum	AVAIL_SPECTRUM_REQ (Section 4.5.1)	AVAIL_SPECTRUM_RESP (Section 4.5.2)
spectrum.paws.getSpectrumBatch	AVAIL_SPECTRUM_BATCH_REQ (Section 4.5.3)	AVAIL_SPECTRUM_BATCH_RESP (Section 4.5.4)
spectrum.paws.notifySpectrumUse	SPECTRUM_USE_NOTIFY (Section 4.5.5)	SPECTRUM_USE_RESP (Section 4.5.6)
spectrum.paws.verifyDevice	DEV_VALID_REQ (Section 4.6.1)	DEV_VALID_RESP (Section 4.6.2)

Table 2

6.1.2. JSON Encoding of Data Models

JSON [RFC7159] encoding of the data models described in Section 4 and Section 5 is straightforward:

- Each data model describes the contents of a JSON object
- Each parameter of a data model corresponds to a member of the corresponding JSON object:
  - The parameter name of the data model is the same as the member name of the JSON object
  - The parameter data type describes the type of the member value
- Primitive types map to JSON type, as described in Section 4, repeated here:

  string: A JSON string, restricted to UTF-8 encoding

  int: A JSON number, without a fractional or exponent part

float: A JSON number

boolean: One of the JSON values, true or false

- The list type maps to a JSON array, except that all values in the array are of the same type

- When the parameter data type refers to another data model, that data model describes a nested JSON object

- The encoded JSON object for each of the Request and Response message listed in the Method Names Table (Table 2) also includes the following members:

  type: The name of the message, e.g., "INIT_REQ"

  version: The PAWS version, e.g., "1.0"

See the following sections for examples.

6.2. Example Encoding: spectrum.paws.init Method

An example of the "spectrum.paws.init" JSON-RPC request is shown below;

```
{
 "jsonrpc": "2.0",
 "method": "spectrum.paws.init",
 "params": {
  "type": "INIT_REQ",
  "version": "1.0",
  "deviceDesc": {
   "serialNumber": "XXX",
   "fccId": "YYY",
   "rulesetIds": ["FccTvBandWhiteSpace-2010"]
  },
  "location": {
   "point": {
    "center": {"latitude": 37.0, "longitude": -101.3}
   }
  }
 },
 "id": "xxxxxx"
}
```

An example of the corresponding JSON-RPC response is shown below:

```
{
 "jsonrpc": "2.0",
 "result": {
  "type": "INIT_RESP",
  "version": "1.0",
  "rulesetInfos": [
    {
      "authority": "us",
      "rulesetId": "FccTvBandWhiteSpace-2010",
      "maxLocationChange": 100,
      "maxPollingSecs": 86400
    }
  ]
 },
 "id": "xxxxxx"
}
```


6.3. Example Encoding: spectrum.paws.getSpectrum Method

An example of the "spectrum.paws.getSpectrum" JSON-RPC request is shown below:


```
{
 "jsonrpc": "2.0",
 "method": "spectrum.paws.getSpectrum",
 "params": {
  "type": "AVAIL_SPECTRUM_REQ",
  "version": "1.0",
  "deviceDesc": {
   "serialNumber": "XXX",
   "fccId": "YYY",
   "rulesetIds": ["FccTvBandWhiteSpace-2010"]
  },
  "location": {
   "point": {
    "center": {"latitude": 37.0, "longitude": -101.3}
   }
  },
  "antenna": {"height": 10.2, "heightType": "AGL"}
 },
 "id": "xxxxxx"
}
```


The following example "spectrum.paws.getSpectrum" JSON-RPC response contains:

- A schedule with two time ranges

- A spectrum profile for one resolution bandwidth (6 MHz)
- The power levels for two frequency segments are shown:
  - From 518 MHz to 542 MHz
  - From 620 MHz to 626 MHz
- In practice, each "profiles" list contains (frequency, power) points to cover all frequencies governed by the associated ruleset. See the Spectrum (Section 5.11) section for a more detailed discussion on the representation.

```
{
 "jsonrpc": "2.0",
 "result": {
  "type": "AVAIL_SPECTRUM_RESP",
  "version": "1.0",
  "timestamp": "2013-03-02T14:30:21Z",
  "deviceDesc": {
   "serialNumber": "XXX",
   "fccId": "YYY",
   "rulesetIds": ["FccTvBandWhiteSpace-2010"]
  },
  "spectrumSpecs": [
   {
    "rulesetInfo": {
      "authority": "us",
      "rulesetId": "FccTvBandWhiteSpace-2010"
    },
    "needsSpectrumReport": false,
    "spectrumSchedules": [
     {
      "eventTime": {
       "startTime": "2013-03-02T14:30:21Z",
       "stopTime": "2013-03-02T20:00:00Z"
      },
      "spectra": [
        {
         "resolutionBwHz": 6e6,
         "profiles": [
           ...
           [
            {"hz":5.18e8, "dbm":30.0},
            {"hz":5.36e8, "dbm":30.0},
            {"hz":5.36e8, "dbm":36.0},
            {"hz":5.42e8, "dbm":36.0}
           ],
```

```
              [
               {"hz":6.20e8, "dbm":30.0},
               {"hz":6.26e8, "dbm":30.0}
              ],
              ...
            ]
           }
         ]
        },
        {
         "eventTime": {
          "startTime": "2013-03-02T22:00:00Z",
          "stopTime": "2013-03-03T14:30:21Z"
         },
         "spectra": [
          ...
         ]
        }
       ]
     }
    ]
   },
   "id": "xxxxxx"
  }
```

The following example "spectrum.paws.getSpectrum" JSON-RPC response includes a spectrum profile that contains specifications for two different bandwidth resolutions (6 MHz and 100 kHz):

```
{
 "jsonrpc": "2.0",
 "result": {
  "type": "AVAIL_SPECTRUM_RESP",
  "version": "1.0",
  "timestamp": "2013-03-02T14:30:21Z",
  "deviceDesc": {
   "serialNumber": "XXX",
   ...
  },
  "spectrumSpecs": [
   {
    "rulesetInfo": {
      "authority": "xx",
      ...
    },
    "needsSpectrumReport": false,
    "spectrumSchedules": [
     {
```

```
          "eventTime": {
           "startTime": "2013-03-02T14:30:21Z",
           "stopTime": "2013-03-02T20:00:00Z"
          },
          "spectra": [
            {
             "resolutionBwHz": 6e6,
             "profiles": [
               ...
               [
                {"hz":5.18e8, "dbm":30.0},
                {"hz":5.36e8, "dbm":30.0},
                {"hz":5.36e8, "dbm":36.0},
                {"hz":5.42e8, "dbm":36.0}
               ],
               [
                {"hz":6.20e8, "dbm":30.0},
                {"hz":6.26e8, "dbm":30.0}
               ],
               ...
             ]
            },
            {
             "resolutionBwHz": 1e5,
             "profiles": [
               ...
               [
                {"hz":5.18e8, "dbm":27.0},
                {"hz":5.36e8, "dbm":27.0},
                {"hz":5.36e8, "dbm":30.0},
                {"hz":5.42e8, "dbm":30.0}
               ],
               [
                {"hz":6.20e8, "dbm":27.0},
                {"hz":6.26e8, "dbm":27.0}
               ],
               ...
             ]
            }
           ]
          },
          {
           "eventTime": {
            "startTime": "2013-03-02T22:00:00Z",
            "stopTime": "2013-03-03T14:30:21Z"
           },
           "spectra": [
            ...
```

```
        ]
       }
      ]
     }
    ]
   },
   "id": "xxxxxx"
  }
```

6.4. Example Encoding: DeviceOwner vCard

The DeviceOwner (Section 5.5) data model contains member values that are JSON encodings of vCard, as described in "jCard: The JSON format for vCard" [RFC7095]. An example fragment is provided below:

```
  {
    ...
    "deviceOwner": {
      "owner": [
        "vcard", [
          ["version", {}, "text", "4.0"],
          ["kind", {}, "text", "org"],
          ["fn", {}, "text", "Racafrax, Inc."]
        ]
      ],
      "operator": [
        "vcard", [
          ["version", {}, "text", "4.0"],
          ["fn", {}, "text", "John Frax"],
          ["adr", {}, "text",
            ["", "", "100 Main Street",
             "Summersville", "CA", "90034", "USA"
            ]
          ],
          ["tel", {}, "uri", "tel:+1-213-555-1212"],
          ["email", {}, "text", "j.frax@rackafrax.com"]
        ]
      ]
    }
  }
```

7. HTTPS Binding

This section describes the use of "HTTP Over TLS" [RFC2818] (HTTPS) as the transfer mechanism for PAWS. TLS provides message integrity and confidentiality between the Master Device and the Database, but only when best current practices are adopted, including use of recommended cipher suites and modes of operation. Consequently, to

improve PAWS security and interoperability, implementations of the Database and Master Device MUST follow best current practices defined by Recommendations for Secure Use of TLS and DTLS [I-D.ietf-uta-tls-bcp].

Depending on prior relationship between a database and device, the server MAY require client authentication, as described in the "Transport Layer Security (TLS) Protocol" [RFC5246], to authenticate the device. When client authentication is required, the database MUST specify, by prior arrangement, acceptable root Certificate Authorities (CAs) to serve as trust anchors for device certificates.

To enable databases to handle large numbers of requests from large numbers of devices, the Database MAY support and devices SHOULD support "Stateless TLS Session Resumption" [RFC5077].

A PAWS request message is carried in the body of an HTTP POST request. A PAWS response message is carried in the body of an HTTP response. A PAWS response SHOULD include a Content-Length header.

The POST method is the only method REQUIRED for PAWS. If a database chooses to support GET, it MUST be an escaped URI, but the encoding of the URI is outside the scope of this document. The database MAY refuse to support the GET request by returning an HTTP error code, such as 405 (method not allowed).

The Database MAY redirect a PAWS request by returning a HTTP 3xx response (as defined by "HTTP/1.1: Semantics and Content" [RFC7231], Section 6.4). The Database MUST provide the redirect URI in the Location header of the 3xx response, and the device MUST handle redirects by using the Location header provided by the Database. When redirecting, the device MUST observe the delay indicated by the Retry-After header. The device MUST authenticate the Database that returns the redirect response before following the redirect. Also, the device MUST authenticate the Database indicated in the redirect. Since the device may communicate with a Database (which it authenticated) without user interaction, when the response code is 301 (Moved Permanently), the device MAY redirect without asking a user for confirmation, even thought it is in response to an HTTP POST method.

The Database SHOULD use HTTP status code "307 Temporary Redirect" to indicate that the device SHOULD resubmit the same request to an alternate URI. The device MAY revert to the original URI for the very next request, or MAY continue to use the alternate URI for a period of time, e.g.,:

o For the remainder of its session, or

- For a fixed period of time, or
- Until power cycled, or
- Until it receives another redirect

However, the device does not need to modify its stored list of URIs.

For a minimum of two weeks before the URI of the Database changes permanently, it MUST use the database-change (DbUpdateSpec (Section 5.7)) mechanism to notify devices, as described in the Configuration Update portion of the Database Discovery (Section 4.1) section. After the Database has moved, requests to the original URI MAY return HTTP status code "301 Moved Permanently" to indicate that the device SHOULD resubmit the request, and all future requests, to the indicated alternate URI.

8. Extensibility

This section describes procedures for extending PAWS. No extensions should be made that would return sensitive device-specific information in Database responses.

8.1. Defining Ruleset Identifiers

A ruleset represents a set of device-side requirements for which the device has been certified. It typically corresponds to, but is not limited to, a set of rules that govern a specific set of radio spectrum for a regulatory domain.

Ruleset identifiers are defined and registered in the Ruleset ID Registry following the procedure in Section 9.1. Ruleset ID values MUST conform to the ruleset-id ABNF. If the Ruleset ID requires additional parameters, they are registered in the PAWS Parameters Registry, as described by Section 9.2.

```
ruleset-id = 1*64ruleset-char
ruleset-char = ALPHA / DIGIT / "_" / "."
```

When defining a Ruleset ID:

- It can be useful for the identifier to be descriptive of the set of rules that allow a device to operate within one or more regulatory domains. For example, it might include the name of a regulatory body or a certification process.
- The identifier SHOULD include some sort of version information, such as a year and/or version number.

- The maximum length of the identifier is 64 octets.

8.2. Defining New Message Parameters

New request or response parameters for use with PAWS are defined and registered in the parameters registry following the procedure in Section 9.2.

Parameter names MUST conform to the param-name ABNF and parameter values syntax MUST be well-defined (e.g., using ABNF, or a reference to the syntax of an existing parameter).

```
param-name = 1*64name-char
name-char = ALPHA / DIGIT / "_"
```

Parameter names use lowerCamelCase by convention. The maximum length of a name is 64 octets.

Unregistered vendor-specific parameter extensions that are not commonly applicable, and are specific to the implementation details of the Database where they are used SHOULD use a vendor-specific prefix that is not likely to conflict with other registered values (e.g., begin with 'companyname').

8.3. Defining Additional Error Codes

Additional error codes can be registered to extend the set listed in Section 5.17, following the procedures in Section 9.3. If the error code requires additional response parameters, they are registered in the PAWS Parameters Registry, as described by Section 9.2.

By convention, the error code is a negative integer value, using one of the range of values defined in Error Codes (Section 5.17). If an appropriate category does not exist, a value from a different range may be used.

9. IANA Considerations

[RFC Editor/IANA: Please replace "[[this document]]" with the RFC number of this document as indicated below, and remove this note prior to publication]

There are three registries associated with PAWS:

- PAWS Ruleset ID Registry (Section 9.1)

- PAWS Parameter Registry (Section 9.2)

o PAWS Error Code Registry (Section 9.3)

Prior to registration, the registrant is encouraged to post to the paws@ietf.org mailing list, including the specification or its draft, to get early feedback.

All registries use the Specification Required policy [RFC5226], with a Designated Expert appointed by the IESG. Specific criteria that the Designated Expert should use in assessing registrations are given below in the description of each registry. The Designated Expert should take advice from the community through the paws@ietf.org mailing list, and the registrant is encouraged to post to the mailing list before formally requesting the registration from IANA. The intention is that new registrations will be accompanied by a published specification. But in order to allow for the allocation of values prior to publication of the specification, the Designated Expert can approve allocations once it seems clear that the specification will be published. Upon approval, IANA will post each registration template that is not included in the text of an RFC.

9.1. PAWS Ruleset ID Registry

This specification establishes the PAWS Ruleset ID Registry.

Ruleset type names for inclusion in PAWS messages are registered on the advice of one or more Designated Experts, with Specification Required [RFC5226]. The specification must include a reference to the regulatory domain to which it applies. To increase interoperability, it is more desirable to have fewer rulesets than to have many rulesets with small variations. Consequently, the Designated Expert should avoid duplication and should encourage the registrant to look for alternatives if there are only small variations from an existing ruleset. The Designated Expert should ensure that the proposed registration is complete with respect to its associated regulatory domain and may seek an expert familiar with those rules to participate in the review on the paws@ietf.org mailing list.

The PAWS Ruleset ID Registry will include the following: 'Ruleset identifier', 'Specification document(s)', and 'Template'. The Template column will include links to the registration templates, either posted by IANA or the relevant sections of RFCs.

9.1.1. Registration Template

Ruleset identifier: The name of the ruleset. See [[this document]], Section 8.1 for the format requirements of this identifier.

Specification document(s): Reference to the document that specifies the parameter, preferably including a URI that can be used to retrieve a copy of the document. An indication of the relevant sections also may be included, but is not required.

Additional Parameter Requirements: Listing of additional parameter requirements to associate with the ruleset. Note that new parameters are registered separately in the PAWS Parameters Registry, as described by Section 8.2. Two types of additional parameter requirements are:

* Addition of new parameters to existing structures, or modification of the REQUIRED and OPTIONAL requirements for existing parameters.

* Modification of requirements to existing parameter values.

For adding new parameters or modifying requirements of existing parameters, the registration should include a table for each affected structure that lists the structure's parameter changes. Each table should include a structure name in its heading and have the following columns:

Parameter name: Name of the parameter added or modified.

Type: Data type of the parameter value.

Requirement: Whether the parameter is REQUIRED or OPTIONAL for the ruleset.

Notes: Any additional notes that might be useful to implementors.

For modifying requirements to existing parameter values, the registration should include a table for each affected structure that lists the structure's parameter changes. Each table should include a structure name in its heading and have the following columns:

Parameter name: Name of the parameter.

Type: Data type of the parameter value.

Additional requirements: Additional requirements on the parameter value.

IANA will post each registration template that is not included in the text of an RFC.

Note that the Additional Parameter Requirements section can be quite extensive, so it will not appear directly in the IANA Ruleset ID Registry table. The table, however, does contain a link to the full registration template for easy access to the additional requirements.

9.1.2. Initial Registry Contents

The PAWS Ruleset ID Registry enables protocol extensibility to support any regulatory domain and ruleset. The initial contents of the registry, however, include only FCC-specific and ETSI-specific entries, because, as of this writing, they are the only regulatory domains that have finalized rules. There is no intent to restrict the protocol to any particular set of authorities.

The initial contents of the PAWS Ruleset ID Registry are listed below; each section corresponds to a single entry in the registry.

9.1.2.1. Federal Communications Commission (FCC)

For the additional parameters that start with the "fcc" prefix, see PAWS Parameters Registry Initial Contents (Section 9.2.2) for more information.

Ruleset identifier: FccTvBandWhiteSpace-2010

Specification document(s): This ruleset refers to the FCC rules for TV-band White Space operations established in the Code of Federal Regulations (CFR), Title 47, Part 15, Subpart H [FCC-CFR47-15H].

Additional Parameter Requirements

Each of the following tables defines additional parameters for the indicated PAWS message. Note that the Requirement column lists FCC, not PAWS, requirements/optionality rules.

The FCC requires registration of "Fixed Devices". Additionally, deviceOwner is required in the registration request:

Registration Request (Section 4.4.1)

Parameter Name	Type	Requirement	Notes
deviceOwner	DeviceOwner (Section 5.5)	REQUIRED	For registering Fixed Devices

Available Spectrum Request (Section 4.5.1)

Parameter Name	Type	Requirement	Notes
deviceDesc	DeviceDescriptor (Section 5.2)	REQUIRED	

Available Spectrum Batch Request (Section 4.5.3)

Parameter Name	Type	Requirement	Notes
deviceDesc	DeviceDescriptor (Section 5.2)	REQUIRED	

DeviceDescriptor (Section 5.2)

Parameter Name	Type	Requirement	Notes
serialNumber	string	REQUIRED	Specifies a device's serial number. See [[this document]], Section 5.2.
fccId	string	REQUIRED	Specifies a device's FCC certification ID (Section 9.2.2.1).
fccTvbdDeviceType	string	REQUIRED	Specifies the FCC Device Type (Section 9.2.2.2) of TV-band White Space device, as defined by the FCC rules.

The following table lists additional requirements for DeviceOwner (Section 5.5) parameter values.

DeviceOwner (Section 5.5)

Parameter Name	Type	Additional Requirement
owner	vCard	The owner is required to contain the formatted name of an individual or organization using the "fn" property. When the name is that of an organization, the entry also is required to contain the "kind" property, with a value of "org".
operator	vCard	The operator entry is required to contain the following properties for the contact person responsible for the device's operation: "fn", "adr", "tel", and "email".

9.1.2.2. European Telecommunications Standards Institute (ETSI)

For the additional parameters that start with the "etsi" prefix, see PAWS Parameters Registry Initial Contents (Section 9.2.2) for more information.

Ruleset identifier: ETSI-EN-301-598-1.1.1

Specification document(s): This ruleset refers to the ETSI Harmonised Standard [ETSI-EN-301-598] established by ETSI.

Additional Parameter Requirements

Each of the following tables defines additional parameters for the indicated PAWS message. Note that the Requirement column lists ETSI, not PAWS, requirements/optionality rules,

DeviceDescriptor (Section 5.2)

Parameter Name	Type	Requirement	Notes
serialNumber	string	REQUIRED	Specifies a device's serial number. See [[this document]], Section 5.2.
manufacturerId	string	REQUIRED	Specifies a device's manufacturer's identifier. See [[this document]], Section 5.2.
modelId	string	REQUIRED	Specifies a device's model identifier. See [[this document]], Section 5.2.
etsiEnDeviceType	string	REQUIRED	Specifies the device's ETSI device type (Section 9.2.2.3).
etsiEnDeviceEmissionsClass	string	REQUIRED	Specifies the device's ETSI device emissions class (Section 9.2.2.4).
etsiEnTechnologyId	string	REQUIRED	Specifies the device's ETSI technology ID (Section 9.2.2.5).
etsiEnDeviceCategory	string	REQUIRED	Specifies the device's ETSI device category (Section 9.2.2.6).

AVAIL_SPECTRUM_REQ (Section 4.5.1)

Parameter Name	Type	Requirement	Notes
requestType	string	OPTIONAL	Modifies the available-spectrum request type. If specified, the only valid value is, "Generic Slave", and the Database is required to respond with generic operating parameters for any Slave Device.

Available Spectrum Batch Request (Section 4.5.3)

Parameter Name	Type	Requirement	Notes
requestType	string	OPTIONAL	Modifies the available-spectrum request type. If specified, the only valid value is, "Generic Slave", and the Database is required to respond with generic operating parameters for any Slave Device.

The following tables define additional requirements for the DeviceDescriptor (Section 5.2) and RulesetInfo (Section 5.6) parameters that appear in the AVAIL_SPECTRUM_RESP (Section 4.5.2) and AVAIL_SPECTRUM_BATCH_RESP (Section 4.5.4) messages. Note that this means the Database is modifying the DeviceDescriptor sent by the Master Device to return device-specific restrictions.

DeviceDescriptor (Section 5.2)

Parameter Name	Type	Requirement	Notes
needsSpectrumReport	boolean	REQUIRED	The Database is required to set this to true to indicate that the device must report spectrum usage.
maxTotalBwHz	float	REQUIRED	Specifies a constraint on total allowed bandwidth.
maxContiguousBwHz	float	REQUIRED	Specifies a constraint on total allowed contiguous bandwidth.
etsiEnSimultaneousChannelOperationRestriction	string	REQUIRED	Specifies a constraint on simultaneous channel operation (Section 9.2.2.7). If it is not provided, the default value is "0".

RulesetInfo (Section 5.6)

Parameter Name	Type	Requirement	Notes
maxLocationChange	float	OPTIONAL	Specifies a constraint on maximum location changes.

9.2. PAWS Parameters Registry

This specification establishes the PAWS Parameters Registry.

Additional parameters for inclusion in PAWS requests, responses, or sub-messages are registered on the advice of one or more Designated Experts, with Specification Required [RFC5226].

The Designated Expert should avoid duplication, i.e., avoid adding a new parameter when an existing one suffices. When a set of parameters is added in support of a new ruleset (Section 9.1), the parameters should share a common prefix that reflects the ruleset ID. The prefix may be omitted, of course, if a parameter has more general applicability. Similarly, when a parameter is not associated with a ruleset, the Designated Expert should ensure that the parameter name does not have a prefix used by existing ruleset parameters (e.g., "fcc", "etsi") or are the initials of an organization that has not yet registered anything, but reasonably might.

The PAWS Parameters Registry will include the following: 'Parameter name', 'Parameter usage location', and 'Specification document(s)'.

9.2.1. Registration Template

Parameter name: The name of the parameter (e.g., "example").

Parameter usage location: The location(s) where the parameter can be used. The possible locations are the named structures defined in Protocol Functionalities (Section 4) and Protocol Parameters (Section 5).

Specification document(s): Reference to the document that specifies the parameter, preferably including a URI that can be used to retrieve a copy of the document. An indication of the relevant sections also may be included, but is not required.

9.2.2. Initial Registry Contents

The PAWS Parameters Registry enables protocol extensibility to support any regulatory domain and ruleset. The initial contents of the registry, however, include only FCC-specific and ETSI-specific entries, because, as of this writing, they are the only regulatory domains that have established rules. There is no intent to restrict the protocol to any particular set of authorities.

The PAWS Parameters Registry's initial contents are listed below; each section corresponds to a row of the registry.

9.2.2.1. FCC ID

Parameter name: fccId

Parameter usage location: DeviceDescriptor (Section 5.2)

Specification document(s): [[this document]] Specifies the device's FCC certification identifier. A valid FCC ID is limited to 19 characters in the ASCII value range, as proposed in FCC Administration Topics Review [FCC-Review-2012-10]. For the purposes of the PAWS protocol, the maximum length of the fccId value is 32 octets.

9.2.2.2. FCC Device Type

Parameter name: fccTvbdDeviceType

Parameter usage location: DeviceDescriptor (Section 5.2)

Specification document(s): [[this document]] Specifies the TV Band White Space device type, as defined by the FCC. Valid values are "FIXED", "MODE_1", "MODE_2".

9.2.2.3. ETSI Device Type

Parameter name: etsiEnDeviceType

Parameter usage location: DeviceDescriptor (Section 5.2)

Specification document(s): Specifies the White Space Device type, as defined by the ETSI Harmonised Standard [ETSI-EN-301-598]. Valid values are single-letter strings, such as "A", "B", etc. Consult the documentation for details about the device types.

9.2.2.4. ETSI Device Emissions Class

Parameter name: etsiEnDeviceEmissionsClass

Parameter usage location: DeviceDescriptor (Section 5.2)

Specification document(s): Specifies the White Space Device emissions class, as defined by the ETSI Harmonised Standard [ETSI-EN-301-598], that characterises the out-of-block emissions of the device. The values are represented by numeric strings, such as "1", "2", "3", etc. Consult the documentation for details about emissions classes

9.2.2.5. ETSI Technology Identifier

Parameter name: etsiEnTechnologyId

Parameter usage location: DeviceDescriptor (Section 5.2)

Specification document(s): Specifies the White Space Device technology identifier, as defined by the ETSI Harmonised Standard [ETSI-EN-301-598]. The maximum length of the string value is 64 octets. Consult the documentation for valid values.

9.2.2.6. ETSI Device Category

Parameter name: etsiEnDeviceCategory

Parameter usage location: DeviceDescriptor (Section 5.2)

Specification document(s): Specifies the White Space Device category, as defined by the ETSI Harmonised Standard [ETSI-EN-301-598]. Valid values are the strings, "master" and "slave". It is case insensitive.

9.2.2.7. ETSI Simultaneous Channel Operation Restriction

Parameter name: etsiEnSimultaneousChannelOperationRestriction

Parameter usage location: SpectrumSpec (Section 5.9)

Specification document(s): Specifies the constraint on the device maximum total EIRP, as defined by the ETSI Harmonised Standard [ETSI-EN-301-598]. The values are represented by numeric strings, such as "0", "1", etc. Consult the documentation for the specification of the power constraint corresponding to each parameter value.

9.3. PAWS Error Code Registry

This specification establishes the PAWS Error Code Registry.

Additional error codes for inclusion in PAWS error messages are registered on the advice of one or more Designated Experts, with Specification Required [RFC5226].

Error codes are intended to be used for automated error handling by devices. Before approval, the Designated Expert should consider whether a device would handle the new error code differently from an existing error code, or whether the difference could be communicated

effectively to the end-user via the "reason" parameter of the Error (Section 5.17) object.

The PAWS Error Code Registry will include the following: 'Code', 'Name', 'Description and Additional parameters'.

9.3.1. Registration Template

Code: Integer value of the error code. The value MUST be an unassigned value in the range -32768 to 32767, inclusive.

Name: Name of the error.

Description and Additional parameters: Description of the error and its associated parameters, if any. It also lists additional parameters that are returned in the data portion of the error (See Section 5.17). New parameters MUST be registered separately in the PAWS Parameters Registry, as described by Section 9.2.

9.3.2. Initial Registry Contents

Initial registry contents are defined in the Table of Error Codes (Table 1).

The registry will also include the error-code categories describing -100s, -200s, and -300s as a note (see Error Codes (Section 5.17)).

10. Security Considerations

PAWS is a protocol whereby a Master Device requests a schedule of available spectrum at its location (or location of its Slave Devices) before it (they) can operate using those frequencies. Whereas the information provided by the Database must be accurate and conform to the applicable ruleset, the Database cannot enforce, through the protocol, that a client device uses only the spectrum it provided. In other words, devices can put energy in the air and cause interference without asking the Database. Hence, PAWS security considerations do not include protection against malicious use of the White Space spectrum. For more detailed information on specific requirements and security considerations associated with PAWS, see "Protocol to Access White Space database: PAWS Use Cases and Requirements" [RFC6953].

By using PAWS, the Master Device and the Database expose themselves to the following risks:

- Accuracy: The Master Device receives incorrect spectrum-availability information.

- Privacy:

-

  - An unauthorized entity intercepts identifying data for the Master Device or its Slave Devices, such as serial number and location.

  - Where databases are required to take device registrations and/or maintain request logs, unauthorized access to such information.

Protection from these risks depends on the success of the following steps:

1. The Master Device must determine the address of a proper database.

2. The Master Device must connect to the proper database.

3. The Database must determine or compute accurate spectrum-availability information.

4. PAWS messages must be transmitted unmodified between the Database and the Master Device.

5. PAWS messages must be encrypted between the Database and the Master Device to prevent exposing private information.

6. For a Slave Device, the spectrum-availability information also must be transmitted unmodified and secure between the Master Device and the Slave Device.

7. When a Listing Server is required, any attack that would prevent reaching a Listing Server would result in all devices relying on that Listing Server ceasing their use of any Whitespace.

8. No future extensions to PAWS can allow the return of sensitive information, such as device information or logs

9. The Database must not allow unauthorized access to device information and request logs and should publish and implement privacy policies regarding their use.

Of these, only steps 1, 2, 4, 5 and 8 are within the scope of this document. This document addresses Step 1 by allowing static provisioning of one or more trusted Databases; dynamic provisioning is out of scope. Step 3 is dependent on specific database

implementations and rulesets and is outside the scope of this document. Step 6 requires a protocol between master and slave devices and is thus outside the scope of this document.

Use of "HTTP Over TLS" [RFC2818], assuming the PKI used is not compromised, ensures steps 2, 4, and 5, as detailed in the following sections:

- Assurance of Proper Database (Section 10.1)
- Protection Against Modification (Section 10.2)
- Protection Against Eavesdropping (Section 10.3)

Any specification for an alternate transport MUST define mechanisms that ensure each of these steps.

In addition to the privacy risks described above, information provided in DeviceDescriptor (Section 5.2) and DeviceOwner (Section 5.5), along with device location, may allow a Database administrator to track the activity and location of a device and its user over time. Risks of secondary use of such tracking information, including sharing with third parties, require out-of-band mitigation, such as public statements or contractual terms. Furthermore, while it is understandable that regulators require DeviceOwner information for higher-power fixed white space devices, for privacy concerns, regulators should not require DeviceOwner information for mobile devices. Similarly, regulators should require, and implementations should provide, device location at a level of granularity only as precise as necessary to support accurate database responses.

10.1. Assurance of Proper Database

This document assumes that the Database is contacted using a domain name or an IP address. Using HTTP over TLS, the Database authenticates its identity, either as a domain name or IP address, to the Master Device by presenting a certificate containing that identifier as a "subjectAltName" (i.e., as a dNSName or IP address). If the Master Device has external information as to the expected identity or credentials of the proper database (e.g., a certificate fingerprint), checks of the subjectAltName MAY be omitted. Note that in order for the presented certificate to be valid at the client, the client must be able to validate the certificate. In particular, the validation path of the certificate must end in one of the client's trust anchors, even if that trust anchor is the Database certificate itself. A Master Device should allow for the fact that a Database can change its certificate authorities (CAs) over time.

10.2. Protection Against Modification

To prevent a PAWS response message from being modified en route, messages must be transmitted over an integrity-protected channel. Using HTTP over TLS, the channel will be protected by appropriate cypher suites.

10.3. Protection Against Eavesdropping

Using HTTP over TLS, messages protected by appropriate cypher suites are also protected from eavesdropping or otherwise unrestricted reading by unauthorized parties en route.

10.4. Client Authentication Considerations

Although the Database can inform a device of available spectrum it can use, the Database cannot enforce that the Master Device uses any/only those frequencies. Indeed, a malicious device can operate without ever contacting a database. Note also that, whereas a malicious device may send fraudulent SPECTRUM_USE_NOTIFY (Section 4.5.5) messages, in the regulatory domains that have established rules, such notifications do not change the available-spectrum answers, so no harm can result from such messages. Consequently, client authentication is not required for the core PAWS (although it may be required by specific regulatory domains).

Depending on a prior relationship between a Database and Master Device, the Database MAY require client authentication. TLS provides client authentication, but there are some considerations:

- The Database must nominate acceptable CAs and the Master Device must have a certificate rooted at one of those CAs.

- As indicated in Section 3.2 of "HTTP Over TLS" [RFC2818], the TLS client authentication procedure only determines that the device has a certificate chain rooted in an appropriate CA (or a self-signed certificate). The database would not know what the client identity ought to be, unless it has some external source of information. Distribution and management of such information, including revocation lists, are outside the scope of this document.

- Authentication schemes are secure only to the extent that secrets or certificates are kept secure. When there are a vast number of deployed devices using PAWS, the possibility that device keys will not leak becomes small. Implementations should consider how to manage the system in the eventuality that there is a leak.

11. Contributors

This document draws heavily from the following Internet Draft documents, draft-das-paws-protocol and draft-wei-paws-framework. The editor would like to specifically call out and thank the contributing authors of these two documents.

Donald Joslyn
Spectrum Bridge Inc.
1064 Greenwood Blvd.
Lake Mary, FL 32746
U.S.A.
Email: d.joslyn at spectrumbridge dot com

Xinpeng Wei
Huawei
Phone: +86 13436822355
Email: weixinpeng@huawei.com

12. Acknowledgments

The authors gratefully acknowledge the contributions of: Gabor Bajko, Ray Bellis, Teco Boot, Nancy Bravin, Rex Buddenberg, Gerald Chouinard, Stephen Farrell, Michael Fitch, Joel M. Halpern, Daniel Harasty, Michael Head, Jussi Kahtava, Warren Kumari, Kalle Kulsmanen, Paul Lambert, Andy Lee, Anthony Mancuso, Basavaraj Patil, Scott Probasco, Brian Rosen, Andy Sago, Peter Stanforth, John Stine, and Juan Carlos Zuniga.

13. References

13.1. Normative References

[I-D.ietf-uta-tls-bcp]
Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of TLS and DTLS", draft-ietf-uta-tls-bcp-02 (work in progress), August 2014.

[JSON-RPC]
"JSON-RPC 2.0 Specification", <http://www.jsonrpc.org/specification>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, January 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5491] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", RFC 5491, March 2009.

[RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, August 2011.

[RFC7095] Kewisch, P., "jCard: The JSON Format for vCard", RFC 7095, January 2014.

[RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.

[RFC7231] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, June 2014.

13.2. Informative References

[ETSI-EN-301-598]
European Telecommunication Standards Institute (ETSI), "ETSI EN 301 598 (V1.1.1): White Space Devices (WSD); Wireless Access Systems operating in the 470 MHz to 790 MHz frequency band; Harmonized EN covering the essential requirements of article 3.2 of the R&TTE Directive", April 2014, <http://www.etsi.org/deliver/etsi_en/301500_301599/301598/01.01.01_60/en_301598v010101p.pdf>.

[FCC-CFR47-15H]
U. S. Government, "Electronic Code of Federal Regulations, Title 47, Part 15, Subpart H: Television Band Devices", December 2010, <http://www.ecfr.gov/cgi-bin/text-idx?rgn=div6&view=text&node=47:1.0.1.1.16.8>.

[FCC-Review-2012-10]
Federal Communications Commission, "Administration Topics Review", October 2012, <http://transition.fcc.gov/bureaus/oet/ea/presentations/files/oct12/2b-TCB-Admin-Issues-Oct-2012-GT.pdf>.

[I-D.ietf-geopriv-uncertainty]
Thomson, M. and J. Winterbottom, "Representation of Uncertainty and Confidence in PIDF-LO", draft-ietf-geopriv-uncertainty-02 (work in progress), August 2014.

[ISO3166-1]
"Country Codes", <http://www.iso.org/iso/country_codes.htm>.

[ITUT.X520.2008]
International Telecommunication Union, "ITU-T Recommendation X.520: Information technology - Open Systems Interconnection - The Directory: Selected attribute types", November 2008, <http://www.itu.int/rec/T-REC-X.520-200811-I>.

[RFC6953] Mancuso, A., Probasco, S., and B. Patil, "Protocol to Access White-Space (PAWS) Databases: Use Cases and Requirements", RFC 6953, May 2013.

[WGS-84] National Imagery and Mapping Agency, "Department of Defense World Geodetic System 1984, Its Definition and Relationships with Local Geodetic Systems, NIMA TR8350.2 Third Edition Amendment 1", January 2000, <http://earth-info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html>.

Appendix A. Database Listing Server Support

As discussed in Database Discovery (Section 4.1), some regulatory domains support the preconfiguration of devices with the URI of a listing server, to which devices can connect to obtain a list of databases certified by the regulatory domain. Regulatory domains may require the device to periodically contact the Database Listing Server to validate and/or update its list of certified databases. If the device is unable to validate its list of certified databases within the required period, regulatory rules may require the device to treat this inability as equivalent to the device having no available spectrum.

A sample JSON response from a Database Listing Server might be represented as follows:

```
{
  "lastUpdateTime": "2014-06-28T10:00:00Z",
  "maxRefreshMinutes": 1440
  "dbs": [
    {
      "name": "Some Operator",
      "uris": [
        {
          "uri": "https://example.some.operator.com",
          "protocol": "paws"
        },
        ...
      ]
    },
    ...
  ]
}
```

In parameters in this sample message are:

lastUpdateTime The time at which the database entries were last updated.

maxRefreshMinutes The maximum interval, expressed in minutes, that is allowed between device requests to the Database Listing Server.

dbs A list of entries for certified databases, each containing the following

name The name of the database operator.

uris One or more URIs for each database, allowing a database to support more than one protocol.

uri, protocol Each protocol supported by a certified database is associated with a separate URI (PAWS protocol URI shown).

Appendix B. Changes / Author Notes.

[RFC Editor: Remove this section before publication.]

Changes from 15:

- More precise language: "supports none of the ruleset" instead of "does not support any ruleset", where it makes sense.

- Batch request: Change MAY to MUST when some locations are acceptable

- Remove explicit mention of OCSP and leave it up the the TLS best practices draft
- Define power levels are EIRP for consistency

Changes from 14:

- Clarified why spectrum-notify is "informational"
- Clarified that device registration is typically only required for fixed devices
- Global statement about timestamp format and must be UTC
- Global statement about MISSING error returned, whether it's required by PAWS or ruleset
- Clarified UNSUPPORTED error
- Mandate that Database-change must be included in all responses a minimum of 2 weeks before change
- Clarified that preconfigured values are ruleset specific (INIT_RESP)
- Added reference to FCC ruleset for registration of Fixed Devices
- Make deviceOwner and serialNumber optional at PAWS level and required on a per-ruleset basis
- Update description for "location" to be where device intends to operate, rather than "current location"
- For REGISTRATION_RESP, add clarification that when it is returned, it will have at least one RulesetInfo. Otherwise, it's an UNSUPPORTED error.
- Clarified that, when a Master Device asks for spectrum on behalf of a Slave Device, there are 2 locations in the message and changed masterDeviceLocation to be required
- Indicate that power levels are typically EIRP (as opposed to conducted power to the antenna)
- Added description for a "schedule"
- Add intro to DEVICE_VALID_REQ

- TLS: Follow best practices to improve security and interop. Reference draft-ietf-uta-tls-bcp
- TLS: Use OCSP for better performance; RFC6960
- TLS: When using client auth, Database determines acceptable root CAs
- Extensibility: Add statement that no extensions that return device information will not be accepted
- Clarify IANA instructions for the Ruleset ID Registry
- Security: Acknowledge that unauthorized access to device registration, other sensitive device info is a risk, and indicate that privacy policies must be published and implement to control access.
-

Changes from 13:

- Clarification in IANA Section 9
- Use full method name in description of the JSON examples in Section 6
- Ask RFC Editor to give full iconectiv name in the Addresses section
- Add URI to ETSC and FCC

Changes from 12:

- Define primitive types in Section 4, specifying UTF-8 for strings and remove UTF-8 references elsewhere.
- Replace or rephrase "parameter" when reference is to higher-level structure.
- Replace "MUST" with non-2119 language (maximum length is xxx octets.
- Rephrase statements that reflected regulatory, not PAWS requirements (e.g., deletion of "MUST" or "MAY" and rephrasing as a regulatory requirement).
- Replace "Paws protocol (redundant) to "PAWS"

- Change "geo-location" to "geolocation"
- General references to "Database" changed to "database."
- Simplify Section 6 by showing several JSON encoding examples.
- Change "Device" to "device"
- Moved treatment of List Server as discovery mechanism to Appendix and added JSON-encoded DB List Server response example.

Changes from 11:

- Change "regulatory rules" to "ruleset"
- Change "regulatory specifics" to "ruleset specifics"
- Change "regulatory-specific" to "ruleset-specific"
- Change "regulatory domain" to "ruleset" where appropriate
- Replace "depends on regulatory domain" to "optional" in tables
- Add "optional" to "*other" in tables
- Change "regulators" to "regulatory domains"
- Change "REQUIRED" error name to "MISSING"
- Changed the IANA instructions and added DE instructions to each section
- Reformat and reorder IANA sections

Changes from 10:

- Ruleset Name change: ETSI-EN-301-598-1.0.9 and update reference to PDF
- Add new ETSI parameter: etsiEnSimultaneousChannelOperationRestriction
- Separate protocol requirements from regulatory requirements

Changes from 09:

- Updated format of the IANA section

Changes from 08:

- Fix JSON typos.
- Added note that JSON schema is not intended to be formally validated
- Finalize paws-iana-review@ietf.org as the email for updating the PAWS IANA registries
- URLs to URIs
- Typo fixes

Changes from 07:

- Propose ruleset ID name for ETSI: ETSI-EN-301-598-1.0.0-draft
- Change TBD email address to paws-iana-review@ietf.org for proposing changes to the PAWS IANA registries
- Moved discussion of required vCard properties to regulatory-specific sections
- Fixed vCard examples for organization names: Use "fn" property, but set "kind" to "org".
- Shorten parameter names:
  - freqHz -> hz
  - powerDbmPerBw -> dbm

Changes from 06:

- Multi-ruleset support:
  - Changed RulesetInfo to have single ruleset ID
  - Changed INIT_RESP to return a list of RulesetInfo parameters, rather than a single one
  - Changed REGISTRATION_RESP to return a list of RulesetInfo parameters to indicate the regulatory domains for which registration was accepted
  - Added SpectrumSpec (Section 5.9) parameter to represent available-spectrum specification for one regulatory domain,

allowing AVAIL_SPECTRUM_RESP and AVAIL_SPECTRUM_BATCH_RESP to include answers for multiple regulatory domains

* Changed GeoSpectrumSchedule to GeoSpectrumSpec (Section 5.15) for supporting batch responses to represent available spectrum for multiple regulatory domains at a location

- To avoid ambiguity or redundant information, clarified that:
  * Event-time intervals within a single set of schedules MUST be disjoint
  * A single Spectrum element MUST cover the entire range of frequencies governed by a ruleset, rather than splitting them to present a "channelized" view
- Add "ruleset" to Terminology section
- Sync Terminology section with Use Case document
- Add "masterDeviceDesc" to Device Validate request
- Add "masterDeviceLocation" to the AVAIL_SPECTRUM requests and the SPECTRUM_NOTIFY message. Change "location" to be the location of the Slave Device, if the request is made by a Master Device on behalf of a Slave Device
- Update vCard reference and example
- Add jsonrpc 2.0 to all sample messages
- Clarify that Listing Servers may be preconfigured in a device
- Clarify meaning of maximum power levels vs bandwidth, including renaming parameter names:
-
  * maxPowerDBm -> powerDbmPerBw
  * bandwidth -> resolutionBwHz
- Explicitly allowed generic JSON-RPC error codes as possible codes.
- Replace SHALL with MUST for consistency
- Replace URI with URL for consistency

- Reduce clutter in JSON encoding examples by removing string-concatenation characters

- Changed "depends" to "depends on regulatory rules" in several places

Changes from 05:

- Remove requirement for JSON-RPC 1.0

- More typo fixes and clarifications

Changes from 04:

- Add "masterDeviceDesc" parameter to the available-spectrum requests to allow both Master and Slave device descriptors when the Master is making the request on behalf of a Slave.

- Add "requestType" parameter to the available-spectrum requests to support requesting generic operating parameters for any Slave Device.

- Add DbUpdateSpec as optional parameter to all response messages and to the error response to allow a Device to detect a database change at any stage of the control flow.

- For the OUTSIDE_COVERAGE error, added ability to return a list of alternate databases

- Explicitly allow JSON-RPC v2.0 and v1.0 encodings

- Relaxed language that state, "MUST stop operation" to "MUST cease use of spectrum under rules for database-managed spectrum". I.e., the device may have other fallback strategies allowed by regulators.

Changes from 03:

- Expanded the Database Discovery mechanism to describe in more detail pre-configuration with URLs of databases and database-listing servers, including mechanisms for updating the configurations when things change

  * Add database-change field to Available Spectrum Response (Section 4.5.2)

- Added fields that are anticipated to be needed by the ETSI harmonized standard for White Space Devices:

    * Added bandwidth constraints to the Available Spectrum Response (Section 4.5.2)

    * Updated Available Spectrum Response to return RulesetInfo, rather than just a ruleset identifier

    * Added optional device-manufacturer and device-model IDs to the DeviceDescriptor (Section 5.2) message. Also moved fccId from this message to the IANA section.

    * Expanded IANA (Section 9) sections

- Clarified restrictions on the specification of the vertices of a Polygon.

- Changed default confidence level to 95% for a point with uncertainty

- Clarified how devices without absolute time source can use the timestamps in the response messages

- Change method names to start with "spectrum.paws." prefix

- Added maximum string lengths

- Updated author contact info

- More typo fixes

Changes from 02:

- Added timestamp to the AVAIL_SPECTRUM_RESP (Section 4.5.2) and AVAIL_SPECTRUM_BATCH_RESP (Section 4.5.4) data models to serve as a reference for the event times in the response. This was accidentally omitted (but was specified in their JSON encodings (Section 6)).

- Fixed typos throughout the JSON encoding (Section 6) sections, typically adding missing commas.

Changed from 01:

- Added a description of message sequences to support multiple rulesets and multiple jurisdictions Section 3.1.

- Modified DeviceDescriptor (Section 5.2) to add rulesetIds parameter

- Modified RulesetInfo (Section 5.6), AvailableSpectrumResponse (Section 4.5.2) to add rulesetId parameter.

- Add Extensibility (Section 8) section.

- Filled in IANA (Section 9) section.

- Removed blank Example Messages section

Changes from 00:

- Add JSON encoding

- Adopt RFC5491 for GeoLocation

- Adopt vCard for contact information

- Add Response Code section and update text referencing the defined response codes

- Change DeviceIdentifier to be DeviceDescriptor, allowing identifiers and device-characteristic fields to be included.

Authors' Addresses

Vincent Chen (editor)
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: vchen@google.com

Subir Das
Applied Communication Sciences
150 Mount Airy Road
Basking Ridge, NJ 07920
U.S.A.

Email: sdas at appcomsci dot com

Lei Zhu
Huawei

Phone: +86 13910157020
Email: lei.zhu@huawei.com

John Malyar
iconectiv
444 Hoes Lane/RRC 4E1106
Piscataway, NJ 08854
U.S.A.

Email: jmalyar at iconectiv dot com

Peter J. McCann
Huawei
400 Crossing Blvd, 2nd Floor
Bridgewater, NJ 08807
USA

Phone: +1 908 541 3563
Email: peter.mccann@huawei.com