

PTASS
PTASS-Draft
Intended Status: Standard Track
Version: Feb 1, 2015 (2-43)

J. Ryoo, Ed.
Stony Brook University
C. Kim
University of Washington
M. Buddhikot
Alcatel-Lucent Bell Labs
Feb 1, 2015

Protocol to Tiered Access for Shared Spectrum (PTASS)
Draft-ptass-protocol-1

Abstract

The radio spectrum in the 3.5GHz band (3550-3650MHz band) released for the commercial operation to correspond to the rapidly growing demand of wireless broadband capacity. State-of-art whitespace protocols are not capable to satisfy the requirements of the rule-making guideline such as "tiered access" and "authority control". Therefore, new and advanced wireless network architecture conform to the rule-making guideline and also maximize the utility of the spectrum sharing is needed.

A spectrum access system (SAS) is used to maintain spectrum sharing and allocate spectrum to the devices. It protects incumbent access and maximize utility of tiered accesses. To achieve interoperability among multiple devices and the SAS, a standardized protocol must be defined and implemented. This document defines such a protocol, the "Protocol to Tiered Access for Shared Spectrum (PTASS)".

Copyright and License Notice

Copyright (c) 2015 Alcatel-Lucent. All rights reserved.

Table of Contents

1	Introduction	5
2	Conventions and Terminology	5
2.1	Conventions Used in This Document	5
2.2	Terminology	5
3	Protocol Overview	6
3.1	Multi-rullset support	9
3.2	Synchronous and Asynchronous Transactions	10
3.2.1	Asynchronous transaction	10
3.2.2	Synchronous transaction	11
3.3	Message Exchange Triggered by the CSAS	12
3.4	Message Exchange between the CSAS and the FSAS	13
4	Protocol Functionalities	14
4.1	Initialization	15
4.1.1	CSAS Discovery	15
4.1.2	INIT_REQ	16
4.1.3	INIT_RESP	16
4.2	Device Registration	17
4.2.1	REGISTRATION_REQ	18
4.2.2	REGISTRATION_RESP	19
4.3	Available Spectrum Query	19
4.3.1	AVAIL_SPEC_REQ	20
4.3.2	AVAIL_SPEC_RESP	21
4.4	Spectrum Usage Query	22
4.4.1	SPECTRUM_USE_REQ	23
4.4.2	SPECTRUM_USE_RESP	23
4.5	Performance Report Notification	24
4.5.1	EventTime	25

4.6	Termination Notification	26
4.7	Ticket Renewal	26
4.7.1	TICKET_RENEW_REQ	27
4.7.2	TICKET_RENEW_RESP	28
4.8	Relocation Query	29
4.8.1	RELOCATION_REQ	29

4.8.2	RELOCATION_RESP	30
4.9	Reassignment Query	31
4.9.1	REASSIGNMENT_REQ	32
4.9.2	REASSIGNMENT_RESP	33
4.10	Deactivation Query	33
4.10.1	DEACTIVATION_REQ	34
4.10.2	DEACTIVATION_RESP	35
4.11	Performance Report Query	36
4.11.1	PERFORMANCE_REPORT_REQ	36
4.11.2	PERFORMANCE_REPORT_RESP	37
4.11.3	Performance Report Requestor	38
4.11.4	Spectrum Scan Requestor	38
4.12	CSAS Registration	39
4.12.1	REGISTRATION_REQ	40
4.12.2	REGISTRATION_RESP	40
4.13	Incumbent Notification	41
4.13.1	Schedule Information	42
4.14	Incumbent Query	43
4.14.1	INCUMBENT_REQ	43
4.14.2	INCUMBENT_RESP	44
5	Protocol Parameters	44
5.1	GeoLocation	44
5.2	StationDescriptor	47
5.3	ServerDescriptor	48
5.4	Interface Descriptor	48
5.5	ScannerDescriptor	49
5.6	TicketInfo	50
5.7	AntennaCharacteristics	50
5.8	Performance Result	51
5.9	SpectrumScanResult	52
5.10	Error Element	53
5.10.1	MISSING Error	55
6	Message Encoding	56
6.1	JSON-RPC Binding	56
6.1.1	Request Object	56
6.1.2	Response Object	57
6.2	init Method	58
6.2.1	INIT_REQ Parameters	58
6.2.2	INIT_RESP Parameters	59
6.3	register Method	60
6.3.1	REGISTRATION_REQ Parameters	60
6.3.2	REGISTRATION_RESP Parameters	62

6.4	getSpectrum Method	63
6.4.1	AVAIL_SPECTRUM_REQ Parameters	63
6.4.2	AVAIL_SPECTRUM_RESP Parameters	64

6.5	useSpectrum Method	65
6.5.1	SPECTRUM_USE_REQ Parameters	65
6.5.2	SPECTRUM_USE_RESP Parameters	66
6.6	report Method	67
6.6.1	PERF_REPORT_NOTIFY	67
6.7	termination Method	68
6.7.1	TERMINATION_NOTIFY	69
6.8	renewal Method	70
6.8.1	TICKET_RENEWAL_REQ Parameters	70
6.8.2	TICKET_RENEWAL_RESP Parameters	71
6.9	relocation Method	72
6.9.1	RELOCATION_REQ Parameters	72
6.9.2	RELOCATION_RESP Parameters	73
6.10	reassignment Method	74
6.10.1	REASSIGNMENT_REQ Parameters	74
6.10.2	REASSIGNMENT_RESP Parameters	75
6.11	deactivation Method	76
6.11.1	DEACTIVATION_REQ Parameters	76
6.11.2	DEACTIVATION_RESP Parameters	77
6.12	report Query Method	78
6.12.1	PERF_REPORT_REQ Parameters	78
6.12.2	PERF_REPORT_RESP Parameters	79
6.13	CSAS registration Method	81
6.13.1	CSAS_REGISTRATION_REQ Parameters	81
6.13.2	CSAS_REGISTRATION_RESP Parameters	82
6.14	incumbent notification Method	83
6.14.1	INCUMBENT_NOTIFY	83
6.15	incumbent access Query Method	85
6.15.1	INCUMBENT_REQ Parameters	85
6.15.2	INCUMBENT_RESP Parameters	86
7	Security Considerations	88
7.1	Assurance of Proper CSAS (to the CBSD)	89
7.2	Protection Against Modification	90
7.3	Protection Against Eavesdropping	90
8	Contributors	90
9	References	90
9.1	Normative References	90
9.2	Informative References	91
	Authors' Addresses	92

1 Introduction

This section provides some high level introductory material. Readers are strongly encouraged to read Further Notice of Proposed Rulemaking [FNPRM, 23 April 2014] for use cases, requirements, and additional background.

The guideline from FNPRM essentially requires new protocol must have following rules. First, tiered access to protect incumbent access and give priority to the regionally licensed access while guarantee floor level nationwide general access. Secondly, the forceful authorization mechanism to control tiered accesses and support variable bandwidth (up to 20MHz) to the general access and assign fixed 10MHz bandwidth to the licensee. Lastly, the new protocol must be available to both fixed and nomadic base stations.

To conform with the guideline, we propose two tiered spectrum access system, the Federal SAS which provide incumbent access related information to the Commercial SAS, and the CSAS which actually allocate the spectrum to the Citizens Broadband Radio Service Device. CSAS is the main control tower of the protocol and message exchanges mainly occurred in between CBSDs and the CSAS. The CSAS maintains geospatial database, and responses to the CBSD's spectrum request by allocating spectrum. This database driven architecture ease the complexity of the CBSD (therefore, the service device can be simplified) and concentrate to the control tower. This centralized architecture also makes policy update effortlessly by not updating all the connected devices but only the CSASs.

This document defines how to obtain operation frequency from the authorized tiered access spectrum sharing devices based on its geo-location capability, and how to activate the allocated frequency to be served as base station. The document describes the use of HTTP/TLS as transport for the protocol.

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 RFC 2119 [RFC2119].

2.2 Terminology

SAS: Spectrum Access System

FSAS: Federal SAS: A system is an entity that contains current information about protected spectrum at a given location and time for incumbent access, as well as other types of information related to protected spectrum and usage. FSAS notifies authorized CSASs about spectrum information to be vacated in terms of location and time. Also, the information can be provided to CSASs by request and response mechanism. Its operation responsibility is on the government authorities such as FCC and ETSI.

CSAS: Commercial SAS: A system is an entity that manages current information about available spectrum at a given location and time, as well as other types of information related to spectrum availability and usage. CBSD information and its engineering specification is collected when it registers. Also, CBSD's performance report may be provided to CSAS in order to compute optimized spectrum availability. This system is a main controller for the dynamic whitespace operations within 3.5GHz.

CBSD: Citizens Broadband Radio Service Device, a device that queries the CSAS, to obtain available spectrum information, and serves as a base station to client devices.

FCC: Federal Communications Commission

ETSI: European Telecommunications Standards Institute

IA: Incumbent Access

PA: Priority Access

GAA: General Authorized Access

Exclusive zone: A protection zone defined for the incumbent access.

JSON: JavaScript Object Notation is a lightweight data-interchange format.

3 Protocol Overview

A Citizens Broadband Radio Service Device (CBSD) uses the Protocol to Tiered Access for Shared Spectrum (PTASS) to obtain a schedule of available spectrum at its location and time. The security necessary to ensure the accuracy, privacy, and confidentiality of the Device's location is described in the Security Considerations (Section 7). This document assumes that the CBSD and the SAS are connected to the Internet.

A typical sequence of PTASS operations is outlined as follows. See protocol functionalities (Section 4) and protocol parameters (Section 5) for details:

1. The CBSD establishes an HTTPS session with the portal CSAS, which bridges the CBSD with a regional CSAS.
2. The CBSD optionally sends an initialization message to the portal CSAS to obtain a regional CSAS URI for its location, this step SHALL skip when the corresponding CSAS (regional CSAS unless

- noted) URI is known to the CBSD.
3. If the CSAS receives an initialization message, it responds with a JSON message for the CSAS URI in the body of the HTTP response.
 4. The CBSD MUST send a registration message to the CSAS in order to verify its accessibility to the spectrum.
 5. As the CSAS receives a registration message, it responds with an encrypted JSON message.
 6. The CBSD sends an available-spectrum request message to the CSAS.
 7. As the CSAS receives an available-spectrum request message, it responds with registration-response message which includes the result of registration and a ticket information to be used for available-spectrum request.
 8. The CBSD sends a spectrum-usage request message to the CSAS in order to inform that it operates within the assigned spectrum.
 9. As the CSAS receives a spectrum-usage request message, it responds by sending the CBSD spectrum-usage response message which includes assigned frequency range and transmit power.
 10. The CBSD periodically sends a performance report notification to the CSAS for the PTASS's performance improvement and channel allocation optimization in the form of JSON.
 11. As the CSAS receives a performance report notification, it SHOULD response in HTTP level, no designated response message for any notification messages.
 12. The CBSD MUST send a termination notification to the CSAS when it stops the operation of assigned spectrum.
 13. If the CSAS receives a termination notification, it SHOULD response in HTTP level.
 14. The CBSD MAY send a renewal request message to the CSAS before the ticket expiration time.
 15. As the CSAS receives a renewal request message, it responds by sending the ticket-renewal response message in JSON form to the CBSD.
 16. The CBSD MAY send a relocation request message to the CSAS.
 17. If the CSAS receives a relocation request message, it responds with a relocation response message in JSON form to the CBSD.

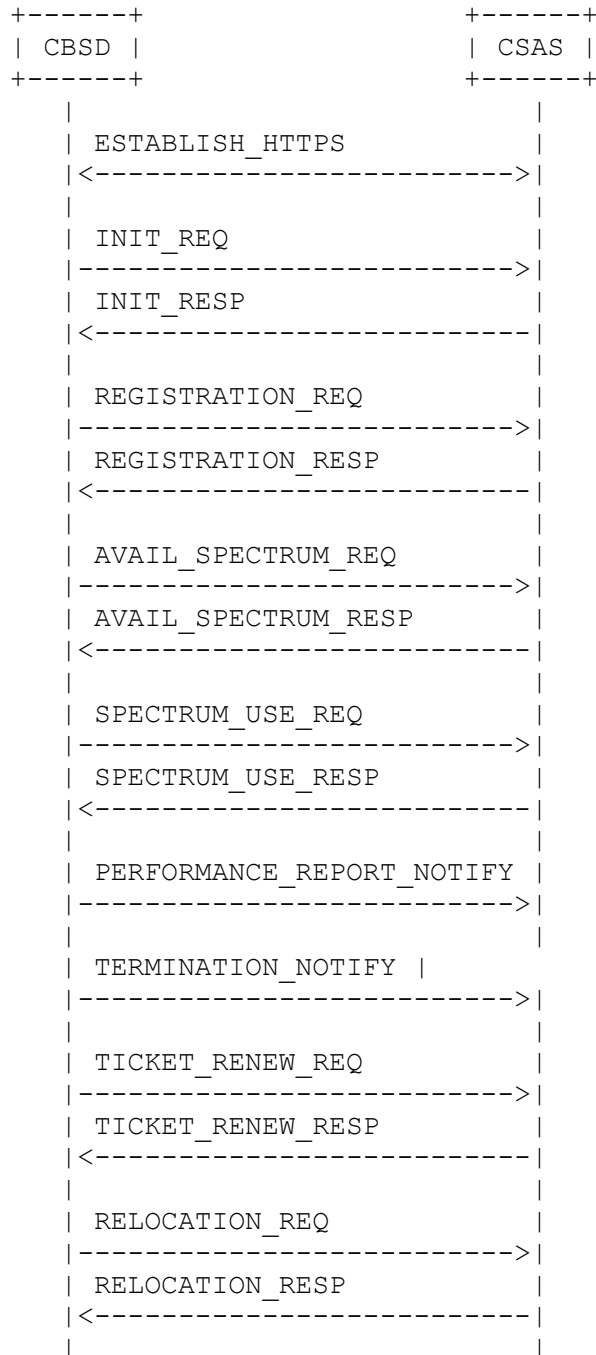


Figure 1

Note:

- o Both CSAS and CBSD MUST manage HTTPS server.
- o The parameters of PTASS messages are defined by JSON primitives or structured types in [RFC7159].
- o Any form of messages are encrypted in PTASS protocol, such as a JSON message encrypted by the HTTPS (HTTP Over TLS [RFC2818]).
- o All parameter names are case sensitive.
- o All time stamps are UTC in the format of "YYYY-MM-DDThh:mm:ss:Z" [RFC3339].
- o The security related policy is described in the Security Considerations (Section 7).

3.1 Multi-rullset support

For a CBSD that supports multiple rullsets and operates with multiple CSASs such as operated by Alcatel-Lucent Bell-labs or Google Incorporation, the PTASS protocol supports the following sequence of operations for each request by the CBSD:

1. The CBSD 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 CSAS 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 CSAS responds with an error and a list of names of the missing parameters
4. The CBSD makes the request again, adding the missing parameter values
5. The CSAS responds to the request, including the identifier of the applicable ruleset
6. The CBSD uses the indicated ruleset to determine how to interpret the CSAS response

NOTE: Regulatory rules contain many device-only requirements that govern device behavior, independent of any CSAS rules. These requirements may be complex and involve CBSD behavior that are not easily parameterized. The ruleset-id parameter provides a mechanism for the CSAS to inform the CBSD of the applicable ruleset without

having to express CBSD-side behavior within the protocol. The ruleset identifier is a string value that contains the name of the regulatory body that established the rules and version information.

By separating the regulatory "authority" from the "ruleset-id", it allows the protocol to support multiple regulatory authorities that

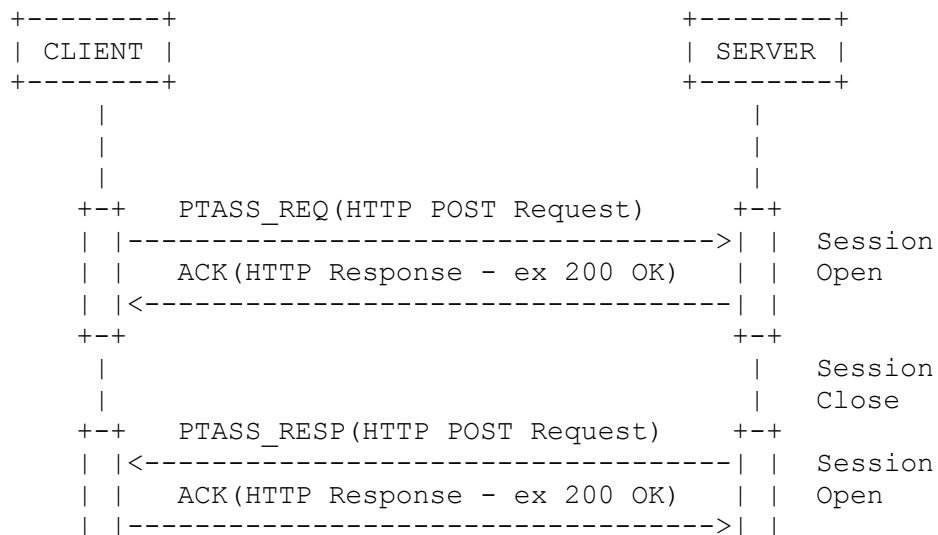
use the same device-side ruleset. It also allows support for a single authority to define multiple rulesets.

3.2 Synchronous and Asynchronous Transactions

The HTTPS asynchronous is mainly adopted for the basic PTASS transactions, requests and responses. We believe, it is more beneficial than synchronous transactions, because it can substantially reduce the number of alive TCP connections in the CSAS server, note that a CSAS handles hundreds of thousands CBSDs. To complete asynchronous based protocol, the client side SHOULD inform reply-IP-address and port-number (HTTPS) to the server side.

3.2.1 Asynchronous transaction

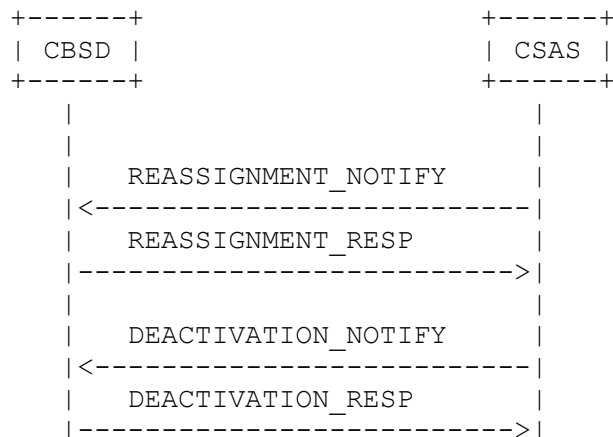
The PTASS uses the native HTTP request/response with HTTP post mechanism, it sends an acknowledgment on successful reception of a request message. Then a response message asynchronously to the client.



3.3 Message Exchange Triggered by the CSAS

As PTASS assumes the http server-service on both CBSD and CSAS side, a CSAS can send a request message to the CBSD in order to reassign spectrum or deactivate its operation for performance optimization and tiered access protection. The active request from the CSAS saves the inefficient period by assigning new spectrum to the CBSD (REASSIGNMENT_NOTIFY), protects the accesses from the harmful interference caused by lower/same tiered accesses (REASSIGNMENT_NOTIFY, DEACTIVATION_NOTIFY), also helps to achieve performance optimization by analyzing data from performance reports (PERFORMANCE_REPORT_QUERY). Sections such as REASSIGNMENT_NOTIFY (Section 4.8), DEACTIVATION_NOTIFY (Section 4.9), and PERFORMANCE_REPORT_QUERY (Section 4.10) cover in detail.

The message exchanges procedures requested from the CSAS are depicted in Figure 4.



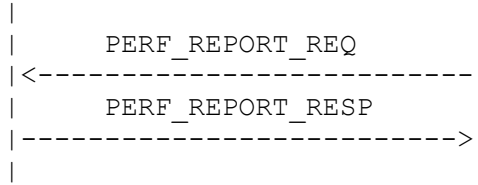


Figure 4

3.4 Message Exchange between the CSAS and the FSAS

The PTASS is tiered access protocol which obtains incumbent access related information from the federal spectrum access system, the FSAS which controlled and operated by the federal authority. The incumbent related information MUST be kept confidentially. The PTASS protocol designed to protect the incumbent access, and not giving any clues to the clients those accessed to CBSDs.

A CSAS MUST register to the FSAS (Section 4.12) in order to query incumbent access related information (Section 4.14). In the procedure of incumbent access request and response, the CSAS can configure the schedule of incumbent access activity and the CSAS MUST schedule their CBSDs to protect the incumbent access's activity. In case of the FSAS needs to update the schedule of IA's activity in the shared spectrum, the FSAS SHALL notify the CSAS with the updated schedule (Section 4.12).

The message exchanges between the CSAS and the FSAS are depicted in Figure 5.



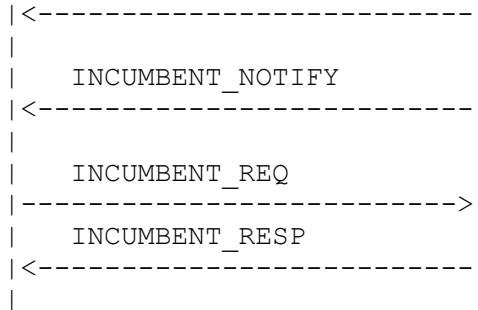


Figure 5

4 Protocol Functionalities

The PTASS protocol consists of several components:

- o CSAS Discovery (Section 4.1.1) MUST be supported by the CBSD.
- o Initialization (Section 4.1) MAY be used by the CBSD and MUST be implemented by the CSAS.
- o Device Registration (Section 4.2) MUST be used by the CBSD and MUST be implemented by the CSAS, either as a separate component or as part of the Available Spectrum Query (Section 4.3) component.
- o Available Spectrum Query (Section 4.3) MUST be supported by CBSD and the CSAS.
- o Spectrum Use Notify (Section 4.4) MUST be used by the CBSD and the CSAS.
- o Performance (Section 4.5) report MUST be implemented by CBSD for periodical report, and MUST be implemented by CSAS for report request and MUST be supported by CBSD for request respond.
- o Termination Notify (Section 4.6) MUST be implemented by CBSD, and MUST be supported by CSAS.
- o Ticket Renewal (Section 4.7) MUST be supported by the CBSD, and MUST be supported by CSAS.
- o Relocation (Section 4.8) MUST be supported by the CBSD.
- o Reassignment (Section 4.9) MUST be implemented by the CSAS, and MUST be supported by CBSD.
- o Deactivation (Section 4.10) MUST be implemented by the CSAS, and MUST be supported by CBSD.
- o Performance Report Query (Section 4.11) MUST be implemented by the

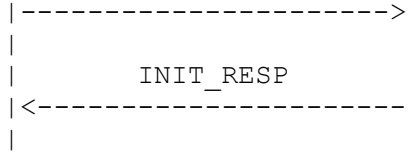


Figure 6

4.1.1 CSAS Discovery

Preconfiguration

The CBSD can be provisioned statically with the URI of one or more CSASs. For example, in a particular regulatory domain, there may be a few number of certified Commercial Spectrum Access System that any CBSD (PA and GAA) operating in that domain is permitted to connect to, and those URIs can be provisioned in the CBSD. Alternatively, a CBSD can be provisioned statically with the URI of the portal CSAS, from which it can retrieve URIs of available CSASs.

Configuration Update

To adapt to changes in the list of certified or approved CSASs, the CBSD needs to update its preconfigured list of CSASs. As we assumed that both CBSD and CSAS are connected to the Internet, the related

information retrieved through alternative connection from the portal CSAS.

The portal CSAS MUST hold regional CSAS URIs and provide available CSAS URIs to the CBSD's request.

4.1.2 INIT_REQ

The initialization request message allows the CBSD to discover regional CSAS URIs with the portal CSAS.

```

+-----+
| INIT_REQ |
+-----+-----+
| location:GeoLocation | required |
| stationDesc:stationDescriptor | required |
| ..... |
| *other:any | optional |
+-----+-----+

```


Parameters:

location: The GeoLocation (Section 5.1) for the Device is REQUIRED.

stationDesc: The stationDescriptor (Section 5.2) for the Device is REQUIRED. The portal CSAS MUST be able to find the corresponding CSAS based on the provided stationDesc and location, otherwise, it MUST return an error with the UNSUPPORTED (Table 1) code in the error response.

other: The CBSD may specify additional initialization parameters. The portal CSAS MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.1.3 INIT_RESP

The initialization response message communicates portal CSAS parameters to the requesting device. In this response message, the identification of the regional CSAS's information is included. The portal CSAS MUST check validity of the CBSD, the serial number MUST be valid in the portal CSAS's database.

```

+-----+
|INIT_RESP|
+-----+-----|
|csasDesc:ServerDescriptor | required |
|.....|
|*other:any | optional |
+-----+-----+

```

Parameters:

csasDesc: The ServerDescriptor (Section 5.3) for the CSAS is REQUIRED. The portal CSAS assigns corresponding CSAS to the CBSD's request based on the ServerDescriptor which included in the request message.

other: The portal CSAS MAY include additional handshake parameters in the INIT_RESP (Section 4.1.3) message. The CBSD MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.2 Device Registration

In the PTASS, the CBSD MUST register by specifying its identification and capabilities to the CSAS due to confidential issue of 3.5GHz spectrum before its operation. The registration is an essential component in the tiered access system, the CSAS MUST NOT assign any spectrum to unregistered devices. Once the CBSD is verified, the CSAS issues ticket(s) to the CBSD to provide enhanced security and enforcement, the ticket will be issued per network interface, and it is only valid for certain period and used for the available spectrum query. We have set it to 24 hours as default period, but it MAY change for the purpose.

The Device Registration request procedure is depicted in Figure 7.

- o REGISTRATION_REQ (Section 4.2.1) is the device-registration request message
- o REGISTRATION_RESP (Section 4.2.2) is the device-registration response message

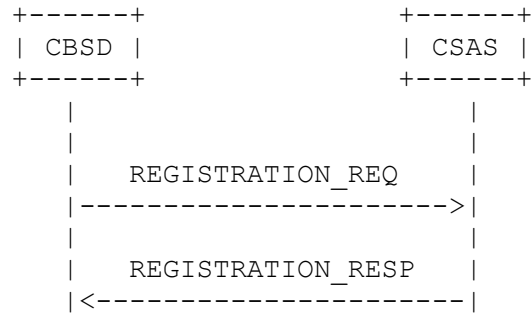
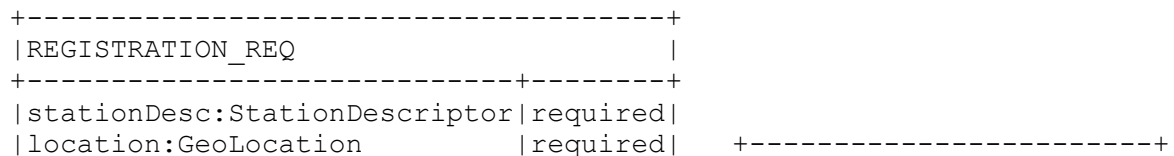


Figure 7

4.2.1 REGISTRATION_REQ

The registration request message contains REQUIRED registration parameters. A parameter marked as OPTIONAL may be required for more information gathering for some CSASs.



```

|interfaces:list          |required|-->|InterfaceDescriptor  |
|licenses:list          |optional|  +-----+-----+
|scannerDesc:ScannerDescriptor|optional|+ |id:string      |required| |
|.....| | |ant:AntennaChar|required|
|*other:any             |optional| | |txPwOutput:list|required|
+-----+-----+ | +-----+-----+
|
| +-----+-----+
+-->|ScannerDescriptor  |
+-----+-----+
|maker:string |required|
|model:string |required|
|flf2:FreqRange|required|
|InstBwHz:float|required|
|link:string  |optional|
+-----+-----+

```

Parameters:

stationDesc: The StationDescriptor for (Section 5.2) for the Device is REQUIRED.

location: The GeoLocation (Section 5.1) for the Device is REQUIRED.

interfaces: The list of InterfaceDescriptor (Section 5.4) is REQUIRED. Note that a CBSD can have multiple interfaces.

licenses: The list of licenses is OPTIONAL. The CBSD MAY have license(s) for PA. Note that a CBSD with licenses and multiple interfaces, SHALL request spectrum for PA and GAA together. The license MAY not bound to the interface.

scannerDescriptor: The ScannerDescriptor (Section 5.5) for the Device is optional. It MAY need for optimization and performance improvement on CSAS side. For some rullset, this descriptor is REQUIRED.

other: The CBSD MAY include additional handshake parameters in the REG_REQ (Section 4.2.1) message. The CSAS MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.2.2 REGISTRATION_RESP

The registration response message is one of the KEY procedures in the PTASS protocol. It assigns a TICKET to a network interface which is valid for a period of time. For the next PTASS messages such as available spectrum query, the ticket represents the interface to be processed fir both CBSD and CSAS.

```

+-----+
|REGISTRATION_RESP          |
+-----+-----+-----+
|ticketInfos:list |required |--->|TicketInfo          |
|.....|
|*other:any      |optional |   |id:string          |required |
+-----+-----+-----+   |ticket:string       |required |
                                   |expireDate:string  |required |
                                   +-----+-----+

```

Parameters:

ticketInfos: The list of TicketInfo (Section 5.6) is REQUIRED. Each entry corresponds to an interface.s

other: The CSAS MAY include additional handshake parameters in the REGISTRATION_RESP (Section 4.2.2) message. The CBSD MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.3 Available Spectrum Query

The spectrum request is REQUIRED to obtain operational spectrum, the CBSD MUST operate at the frequency that assigned by the CSAS.

The CBSD sends a request that includes a ticket to identify the device, the desired bandwidth, and the use of license. For a PA

licensee 10MHz is assigned per license, a PA MAY increase their bandwidth by requesting wider bandwidth, several licenses can be assigned to a interface, however, a GAA can request various bandwidths for its performance without limitation.

The CSAS responds to the CBSD with assigned frequency range and permissible transmit power.

The available spectrum query procedure is depicted in Figure 8.

- o AVAIL_SPEC_REQ (Section 4.3.1) is the available spectrum request message
- o AVAIL_SPEC_RESP (Section 4.3.2) is the available spectrum response message

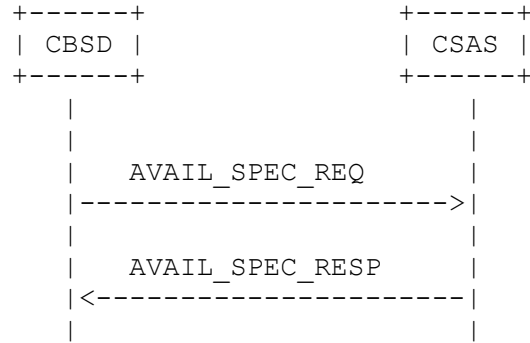


Figure 8

4.3.1 AVAIL_SPEC_REQ

The CBSD requests available frequency range to the CSAS at its location and time, the CBSD SHOULD request the spectrum before starting service in 3.5GHz spectrum.

The available spectrum request message contains REQUIRED parameters.

```

+-----+
|AVAIL_SPEC_REQ          |
+-----+-----+
|ticket:string          |required |
|bandwidthHz:float     |required |
|usesLicense:bool      |required |
|.....                 |         |
|*other:any            |optional |
+-----+-----+

```

Parameters:

ticket: The ticket is REQUIRED.

bandwidthHz: The bandwidth in Hz is REQUIRED. It is desired bandwidth on the CBSD side.

usesLicense: The indication of usage of license is REQUIRED. In case a CBSD is licensee and willing to use its license for the interface, the CBSD MUST set this value to "true". Therefore, CSAS considers the interface as a PA for spectrum assignment and protection. as PA.

other: The CBSD MAY include additional handshake parameters in the AVAIL_SPEC_REQ (Section 4.3.1) message. The CSAS MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.3.2 AVAIL_SPEC_RESP

The available spectrum response message is for the allocation of spectrum from the CSAS to the CBSD, it is the spectrum allocation MESSAGE in PTASS protocol, the goal of protocol. In this message the CSAS assigns the spectrum to the CBSD. The CSAS MAY assign the lowered transmit power to protect primary users and minimize interference to higher-tiered devices and devices in the same tier.

```

+-----+
|AVAIL_SPEC_RESP          |
+-----+-----+
|ticket:string           |required | +-----+
|freqRange:FrequencyRange |required |--->|FrequencyRange      |
|txPowerOutput:int       |optional | +-----+-----+
|.....|                 |          | |fc1:float |required |
|*other:any              |optional | |fc2:float |required |
+-----+-----+ +-----or-----+
|                          |          | |fc:float  |required |
|                          |          | |BW:float  |required |
+-----+-----+ +-----+-----+

```

Parameters:

ticket: The ticket which represents the requested interface is REQUIRED.

freqRange: The frequency range of the assigned spectrum is REQUIRED. The FrequencyRange is either in the form of start frequency and stop frequency in Hz or center frequency and bandwidth of the assigned

spectrum.

txPowerOutput: The transmit power output is OPTIONAL. The transmit power level from the device OPTIONALLY included to control the transmit power of the device, it is based on the REGISTRATION_REQ (Section 4.2.1).

other: The CSAS MAY include additional handshake parameters in the AVAIL_SPEC_RESP (Section 4.3.2) message. The CBSD MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.4 Spectrum Usage Query

The spectrum usage query is REQUIRED. The device MUST request the usage of the assigned spectrum for an interface to the CSAS to be served as a CBSD. Before sending the request, the spectrum MUST be assigned with AVAIL_SPEC_RESP (Section 4.3.2) from the CSAS. In the response message from the CSAS, the CSAS MAY activate the device as CBSD with the activation result "true", otherwise a CBSD MUST not be served, this mechanism forces the CBSD under the control of the CSAS.

The spectrum usage query procedure is depicted in Figure 9.

- o SPECTRUM_USE_REQ (Section 4.4.1) is the spectrum use request message
- o SPECTRUM_USE_RESP (Section 4.4.2) is the spectrum use response message

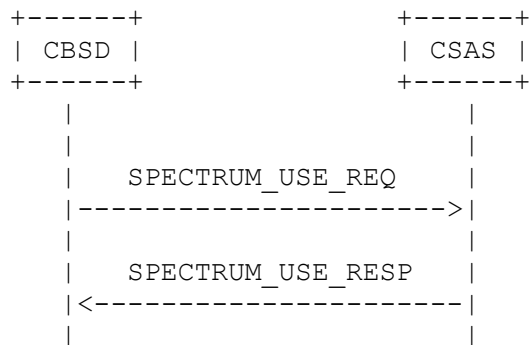


Figure 9

4.4.1 SPECTRUM_USE_REQ

The CBSD requests the usage of assigned spectrum to the CSAS for the certain period of time at its location, the CBSD MUST request its usage to obtain activation result in the SPECTRUM_USE_RESP (Section 4.4.2).

The spectrum usage request message contains REQUIRED parameter.

```
+-----+
|SPECTRUM_USE_REQ          |
+-----+-----+
|ticket:string            |required |
|.....|
|*other:any              |optional |
+-----+-----+
```

Parameters:

ticket: The ticket is REQUIRED. A ticket of the interface, the CSAS knows the assigned spectrum to its ticket.

other: The CBSD MAY include additional handshake parameters in the SPECTRUM_USE_RESP (Section 4.4.2) message. The CSAS MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.4.2 SPECTRUM_USE_RESP

The CSAS responds to the spectrum usage request, this confirms the rights of the spectrum use until expiration. Note that a CSAS MAY expire the ticket for various reasons. The CSAS updates CBSD's interface as activated state in the database, which indicates the CBSD's operation within the assigned spectrum now. The CBSD MUST start its service after reception of the response message with activation result "true".

The spectrum usage response message contains REQUIRED parameters.

```
+-----+
|SPECTRUM_USE_RESP          |
+-----+-----+
|activate:bool            |required |
|ticket:string            |required |
|.....|
|*other:any              |optional |
```


+-----+-----+

Parameters:

activate: The activation result for the CBSD's operation is REQUIRED.
Only when this boolean value is true, the CBSD MUST start its
service, otherwise, the CBSD MUST not be activated.

ticket: The ticket of the requested interface is REQUIRED.

other: The CSAS MAY include additional handshake parameters in the
SPECTRUM_USE_RESP (Section 4.4.2) message. The CBSD MUST ignore all
parameters it does not understand. Consult the PTASS Parameters
Registry for possible additional parameters.

4.5 Performance Report Notification

The performance report notification is REQUIRED. The CBSD SHOULD
notify periodic performance report to the CSAS, a CSAS can optimize
spatial and temporal spectrum usage based on this report. A CSAS MAY
reassign spectrum to CBSDs those show poor performance or hinder
neighboring CBSDs' performance. Optional spectrum sensing data based
performance report set the basis for the CSAS's spectrum
optimization. The report designed for 1) Accessibility, 2)
Retainability, 3) Integrity, 4) Availability, and 5) Mobility.

The performance report notify is one directional notification from
the CBSD to the CSAS and this message contains REQUIRED parameters.
Note that a CBSD sends a report, not every network interfaces.

PTASS-Draft

ptass-protocol

February 1, 2015

```

+-----+
|PERF_REPORT_NOTIFY      |
+-----+-----+-----+
|perfResults:list      |required|--->|PerformanceResult      |
|scanResult:SpecScan  |optional|+  +-----+-----+-----+
|.....| | |ticket:string          |required|
|*other:any           |optional| | |timeRange:EventTime*   |required|
+-----+-----+-----+ | |nClient:int             |optional|
| | |avgTputMbps:float     |optional|
| | |avgNRetrans:float    |optional|
| | |avgDelayMs:float     |optional|
| | |avgPER:float        |optional|
| | +-----+-----+-----+
| | |
| | +-----+-----+-----+
+-->|SpecScan              |
| | +-----+-----+-----+
| | |devSerialNum:string   |required|
| | |timeRange:EventTime*  |required|
| | |freqRange:FrequencyRange|required|
| | |nBins:int             |required|
| | |signalStrength:list   |required|
| | +-----+-----+-----+
| | |
| | +-----+-----+-----+
*|EventTime              |
| | +-----+-----+-----+
| | |startTime:string      |required|
| | |stopTime:string       |required|
| | +-----+-----+-----+

```

Parameters:

perfResults: The performance results are REQUIRED. A list of PerformanceResult (Section 5.8) for every activated interfaces.

scanResult: The scan result of the CBSD is OPTIONAL. A SpecScan (Section 5.9) at the location of the device.

other: The CBSD MAY include additional parameters in the message. The CSAS MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.5.1 EventTime

The event time indicates the start time and stop time of the event. It contains start time and stop time as REQUIRED. Each time is in UTC as string format.

4.6 Termination Notification

A CBSD MUST send termination-notify-message to the CSAS to inform the end of service meaning that the CBSD no longer use assigned spectrum from the CSAS. The notification includes the list of tickets those stop operating. When the CSAS receives the notify message, it updates the interface state as idle. Therefore, spectrum utilization can increase by reducing hollow allocation.

The CSAS does not return a response message.

```
+-----+
|TERMINATION_NOTIFY      |
+-----+-----+
|tickets:list  |required |
|code:int      |optional |
|.....|
|*other:any    |optional |
+-----+-----+
```

Parameters:

tickets: The list of tickets are REQUIRED. The tickets those terminate the service. A CBSD MAY terminate some of interfaces and the rest still operates. The CSAS expires the ticket validity for terminated interfaces.

code: The code which indicates the reason of termination is OPTIONAL. The error code (Section 5.10) covers list of code and corresponding explanation.

other: The CBSD MAY include additional parameters in the message. The CSAS MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.7 Ticket Renewal

The ticket renewal is REQUIRED to renew a CBSD's ticket before the expiration. Once the CBSD's ticket is about to expire (an hour ahead by default), the CBSD SHALL send ticket renewal request message, once

the CSAS receives request message, the CSAS MUST reply with the response message, it is required to have "is the ticket allowed to renew" with boolean indication, or the response message optionally

explains the reason for the decline of the renewal with error the code (Section 5.10). The other case, the CBSD is not send renewal message (and the renewal is declined), the current ticket expires as it is scheduled, therefore, the CBSD MUST restart from the registration (Section 4.2). This ticket renewal saves the message exchanges between the CBSD and the CSAS, by extending previously issued ticket, and it is only allowed its extension when the CSAS has nothing to change on currently assigned spectrum.

The ticket renewal request procedure is depicted in Figure 10.

- o TICKET_RENEW_REQ (Section 4.7.1) is the ticket renewal request message
- o TICKET_RENEW_RESP (Section 4.7.2) is the ticket renewal response message

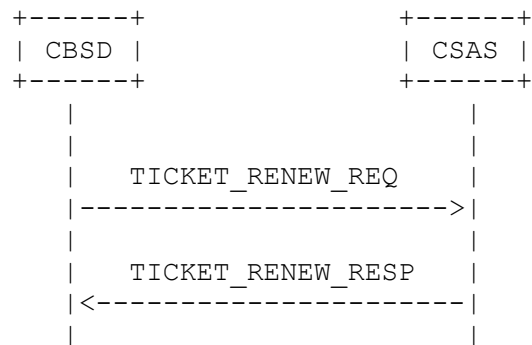


Figure 10

4.7.1 TICKET_RENEW_REQ

The CBSD requests the renewal of its current ticket to the CSAS, when the CBSD requests the ticket renewal, the request message REQUIRED to contain its current ticket and expiration date which expires within an hour.

The renewal request message contains REQUIRED parameters.

PTASS-Draft

ptass-protocol

February 1, 2015

```

+-----+
|TICKET_RENEW_REQ          |
+-----+-----+
|ticket:string             |required |
|expireDate:string        |required |
|.....                    |
|*other:any                |optional |
+-----+-----+

```

Parameters:

ticket: The ticket for renewal is REQUIRED.

expireDate: The current expiration date of the ticket is REQUIRED.

other: The CBSD MAY include additional parameters in the message. The CSAS MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.7.2 TICKET_RENEW_RESP

The ticket renewal response message is the result of the ticket's renew-ability. If the device has nothing to impact the CSAS's regional spectrum scheduling and optimization, the CSAS SHALL allow its renewal by sending "true" boolean for indication, then CBSD SHOULD continually use the spectrum until the updated expiration date without further message exchanges. Otherwise "false" boolean to the CBSD, then the ticket will expire on its scheduled expiration date, then the CBSD MUST start from the registration (Section 4.2) to obtain new spectrum.

The renewal response message contains REQUIRED parameter.

```

+-----+
|TICKET_RENEW_RESP        |
+-----+-----+
|isRenewed:bool           |required |
|ticket:string            |required |

```

```

|expireDate:string |optional |
|code:int           |optional |
+-----+-----+

```

Parameters:

isRenewed: The renew-ability boolean that simply tells the CBSD "true" or "false" is REQUIRED.

Ryoo, et al.

Version: Feb 1, 2015

[Page 28]

PTASS-Draft

ptass-protocol

February 1, 2015

ticket: The requested ticket for renewal is REQUIRED.

expireDate: The OPTIONAL expiration date when the ticket is renewed.

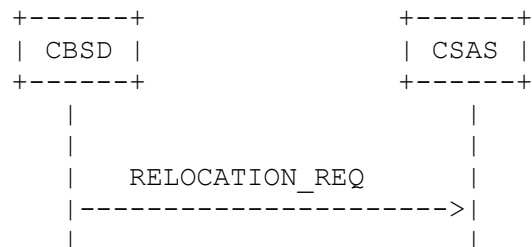
code: the code indicates the reason of decline OPTIONAL. The error code (Section 5.10) covers list of code and corresponding explanation.

4.8 Relocation Query

The relocation request is REQUIRED to inform its new location to the CSAS. The relocation OPTIONALLY explains the reason for the relocation with the error code (Section 5.10). Once the CBSD is relocated further than threshold for any reasons, it SHOULD report to the CSAS to check that the CBSD is still under the corresponding CSAS's area. In case of the CBSD needs to renew the CSAS, the CBSD MUST contact to the portal CSAS, and all processes MUST be started from the initialization (Section 4.1). The other case that the CBSD is still under the same CSAS's region, the CBSD MAY start from the available spectrum request (Section 4.3). Note that relocation MAY not frequently occurred, and this process rather be replaced with termination (Section 4.6) and registration process (Section 4.2).

The relocation request procedure is depicted in Figure 11.

- o RELOCATION_REQ (Section 4.8.1) is the relocation request message
- o RELOCATION_RESP (Section 4.8.2) is the relocation response message



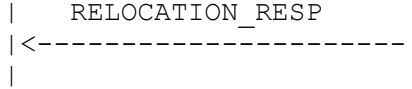


Figure 11

4.8.1 RELOCATION_REQ

The CBSD requests the validity of its new location to the CSAS. The request message MUST contain the updated location, the station

description in order to identify the CBSD, and OPTIONAL error code which explains the reason for the relocation.

The relocation request message contains REQUIRED parameters.

```

+-----+
|RELOCATION_REQ|
+-----+-----+
|location:GeoLocation|required|
|stationDesc:StationDescriptor|required|
|code:int|optional|
|.....|
|*other:any|optional|
+-----+-----+

```

Parameters:

location: The GeoLocation (Section 5.1) for the Device is REQUIRED.

stationDesc: The StationDescriptor (Section 5.2) for the Device is REQUIRED.

code: the code indicates the reason of relocation is OPTIONAL. The error code (Section 5.10) covers list of code and corresponding explanation.

other: The CBSD MAY include additional parameters in the message. The CSAS MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.8.2 RELOCATION_RESP

The relocation response message is the result of the CSAS's regional validity. If the device is still under the control region of the CSAS, the CSAS will send "true" boolean and continue to serve as CSAS to the CBSD. Otherwise "false" boolean to the CBSD, then the CBSD MUST contact the portal CSAS again.

The relocation response message contains REQUIRED parameter.

```
+-----+
|RELOCATION_RESP      |
+-----+-----+
|isValid:bool      |required |
+-----+-----+
```

Parameters:

isValid: The validity boolean that simply tells the CBSD "true" or "false".

4.9 Reassignment Query

The CSAS MAY request reassignment to the CBSD. The CSAS MAY request parameter-update to the CBSD such as operational spectrum update or transmit power adjust with the time to use new parameters. The reassignment is caused by various reasons such as presence of incumbents or performance optimization of the CBSDs. The error code OPTIONALLY explains the reason for the reassignment.

The design choice of the time stamp (rather than update instantly) is

- o To adapt various network environment over the national wide, there is possibility that some CBSDs under the harsh network environment. For example, a home installed GAA based CBSD with economic Internet plan, the connection between CBSD-CSAS is not guaranteed and the number of routers and gateways between CBSD-CSAS are inestimable. We believe that the CSAS's request of the parameter-update beforehand helps robust and reliable operation.

- o To guarantee the evacuation, in case of multiple exclusive zones are assigned to the CSAS from the FSAS with the schedule. The number of CBSDs related to that exclusive zones could be tens of thousands for a single CSAS, to guarantee the parameter-update for all related CBSDs, the reassignment beforehand with the time is more efficient and better for robust operation than waits for the last minute and requests the reassignment.

By setting the time as of now, the CSAS MAY request assignment immediately, however, this kind of case rarely occurred, generally the IA runs in schedule.

The reassignment request procedure depicted in Figure 12.

- o REASSIGNMENT_REQ (Section 4.9.1) is the reassignment request message
- o REASSIGNMENT_RESP (Section 4.9.2) is the reassignment response message

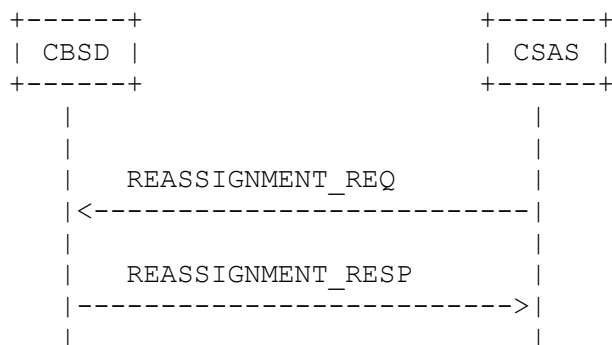


Figure 12

4.9.1 REASSIGNMENT_REQ

The CSAS requests the parameter-update of the CBSD. The purpose of reassignment is to protect higher accesses from harmful interface and optimize CBSDs' performance. The request contains a ticket of the network interface to be assigned, new spectrum, and new transmitting power output those MUST be applied by the indicated time. An optional code MAY explain the reason for the event. To update multiple interfaces, the request MUST be sent multiple times.

The reassignment request message contains REQUIRED parameters.

+-----+

REASSIGNMENT_REQ	
ticket:string	required
timestamp:string	required
freqRange:FrequencyRange	required
txPowerOutput:int	required
code:int	optional
.....	
*other:any	optional

Parameters:

ticket: The ticket is REQUIRED. A ticket of the interface.

timestamp: the timestamp is REQUIRED. The specific time that parameters MUST be updated. The date and time in the format of UTC timestamp.

freqRange: The frequency range of the newly assigned spectrum is REQUIRED. The FrequencyRange is either in the form of start frequency and stop frequency in Hz or center frequency and bandwidth of the

assigned spectrum.

txPowerOutput: The transmit power output is REQUIRED. The new transmit power level of the device. The possible power values of the CBSD are known to the CSAS through the device registration.

code: the code indicates the reason of reassignment is OPTIONAL. The error code (Section 5.10) covers list of code and corresponding explanation.

other: The CSAS MAY include additional handshake parameters in the REASSIGNMENT_REQ (Section 4.9.1) message. The CBSD MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.9.2 REASSIGNMENT_RESP

The reassignment response message MUST be returned after parameter-update is physically reassigned. Therefore it is sent after the timestamp which indicated in the request message. If the device is updated with the new parameters, the CBSD will send "true" boolean, and continue to serve with the updated parameters. The other cases, when a CBSD sends "false" or no response, the CSAS will send DEACTIVATION_REQ (Section 4.10) message to terminate the CBSD's

operation, and automatically expires the corresponding ticket.

The reassignment response message contains REQUIRED parameter.

```
+-----+
|REASSIGNMENT_RESP          |
+-----+-----+
|ticket:string              |required |
|isReassigned:bool         |required |
+-----+-----+
```

Parameters:

ticket: The ticket is REQUIRED. A ticket of the updated interface.

isReassigned: The boolean that simply tells the device's parameters are updated ("true") or not ("false") is REQUIRED.

4.10 Deactivation Query

Ryoo, et al.

Version: Feb 1, 2015

[Page 33]

PTASS-Draft

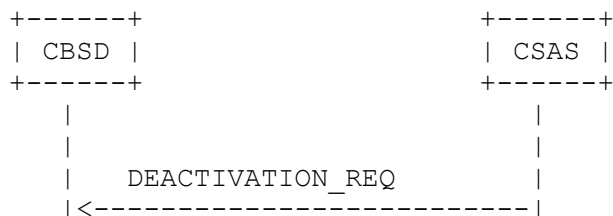
ptass-protocol

February 1, 2015

The CSAS MAY request deactivation to the CBSD. The deactivation can be caused by various reasons such as presence of incumbents, or to protect higher (or equal) tiered devices from the harmful interference. The CBSD returns a response that includes the result of deactivation.

The deactivation request procedure depicted in Figure 13.

- o DEACTIVATION_REQ (Section 4.10.1) is the deactivation request message
- o DEACTIVATION_RESP (Section 4.10.2) is the deactivation response message



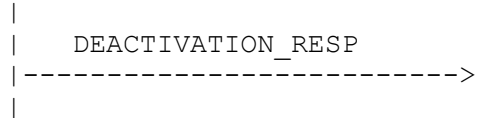


Figure 13

4.10.1 DEACTIVATION_REQ

The request message for deactivation contains a ticket of the interface to be deactivated, the deactivation time, and the code which indicates a reason of deactivation. To deactivate multiple (or all) interfaces of the CBSD, the request message from the CSAS MUST be sent multiple times.

The deactivation request message contains REQUIRED parameters.

```

+-----+
|DEACTIVATION_REQ          |
+-----+-----+
|ticket:string             |required |
|timeRange:EventTime       |required |
|code:int                  |optional |
|.....                    |
|*other:any                |optional |
+-----+-----+

```

Parameters:

ticket: The ticket is REQUIRED. A ticket of the interface.

timeRange: The time period of the device MUST be deactivated is REQUIRED. It is EventTime which indicates start and stop time of the deactivation.

code: the code indicates the reason of deactivation is OPTIONAL. The error code (Section 5.10) covers list of code and corresponding

other: The CSAS MAY include additional handshake parameters in the DEACTIVATION_REQ (Section 4.10.1) message. The CBSD MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.10.2 DEACTIVATION_RESP

The deactivation response message MUST be returned after the interface is deactivated.

Therefore it is sent after the timestamp which indicated in the request message. If the interface is deactivated, the CBSD will send "true" boolean, and continue to serve with the rest of activated interfaces. The other cases, when a CBSD sends "false" or no response, the CSAS will send DEACTIVATION_REQ (Section 4.10.1) message repeatedly to deactivate the CBSD's operation, and automatically expires the corresponding ticket.

The deactivation response message contains REQUIRED parameter.

```
+-----+
|DEACTIVATION_RESP          |
+-----+-----+
|ticket:string              |required |
|isDeactivated:bool        |required |
+-----+-----+
```

Parameters:

ticket: The ticket is REQUIRED. The corresponding ticket of the CBSD.

isDeactivated: The boolean that simply tells the device's interface is deactivated ("true") or not ("false") is REQUIRED.

4.11 Performance Report Query

A CSAS MAY request performance report to the CBSD. When the CSAS needs additional performance report for optimization or monitoring purpose, or OPTIONAL spectrum scanning data for detail spectrum environment. The response report has the same structure with the PERF_REPORT_NOTIFY (Section 4.5), note that it is request/response message unlike the notification. Therefore, response message contains transaction ID. The CBSD SHOULD report every repetition time for those requested interfaces and optionally requested scanResult.

The performance report request procedure depicted in Figure 14.

- o PERFORMANCE_REPORT_REQ (Section 4.11.1) is the performance report request message
- o PERFORMANCE_REPORT_RESP (Section 4.11.2) is the performance report response message

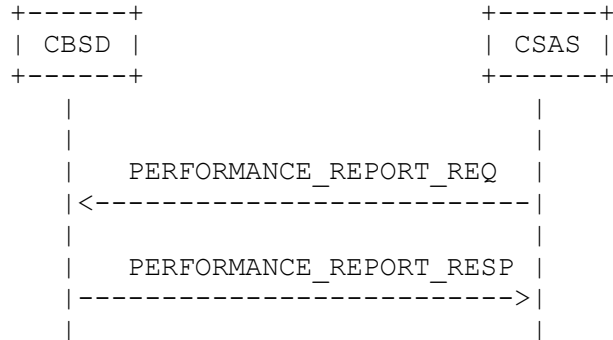


Figure 14

4.11.1 PERFORMANCE_REPORT_REQ

A CSAS MAY request detailed performance report on specific interface. When the CSAS decides the report is needed for optimization or for monitoring purpose, it can request the report to the CBSD.

The performance report request message contains REQUIRED parameters. Note that the CSAS requests a report, with a(multiple) performance report requestor(s) (Section 4.11.3).

```

+-----+
| PERFORMANCE_REPORT |
+-----+-----+
| repetitionMin:int |required| +-----+
| perfReportReqs:list|required| ---> |PerformanceReportRequestor |
| scanReq:SpecScanReq|optional| -+ +-----+-----+
| ..... | | ticket:string |required|
| *other:any |optional| | reqTput:bool |optional|
+-----+-----+ | reqNRetrans:bool |optional|
| | reqDelay:bool |optional|
| | reqPER:bool |optional|
| | ..... |

```

```

| |*other:any |optional|
| +-----+-----+
|
| +-----+-----+
+->|SpecScanRequestor |
| +-----+-----+
|freqRange:FrequencyRange|required|
|nbins:int |required|
|.....|
|*other:any |optional|
+-----+-----+

```

Parameters:

repetitionMin: The repetition minute in integer is REQUIRED. The CSAS MAY request report every specified minute.

perfReportReqs: The list of performance report requests is REQUIRED. A list of PerformanceReportRequestor(Section 4.11.3) for every requested interfaces.

scanReq: The spectrum scan request is OPTIONAL. A SpecScanRequestor(Section 4.11.4) at the location of the device.

other: The CSAS MAY include additional parameters in the message. The CBSD MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.11.2 PERFORMANCE_REPORT_RESP

The performance report response message contains the list of perfResult (Section 4.11.5), and OPTIONALLY sends scanResult(Section 4.11.6).

The performance report response message contains REQUIRED parameter.

```

+-----+-----+
|PERF_REPORT_RESP |
| +-----+-----+
|perfResults:list |required |
|scanResult:SpecScan |optional |
+-----+-----+

```

Parameters:

perfResults: The list of performance result report is REQUIRED. A list of PerformanceResult (Section 5.8).

scanResult: The scan result of requested frequency is OPTIONAL. The CBSD MAY response SpecScan (Section 5.9).

4.11.3 Performance Report Requestor

The performance report requestor is a report-requestor that CSAS asks for report of the interface. This report-requestor contains parameters to find out how a interface performs.

The PerformanceReportRequestor contains REQUIRED parameters.

ticket: The ticket is REQUIRED. A ticket of the interface.

reqTput: The request of throughput in boolean is OPTIONAL. An interface's average throughput MAY be requested.

reqNRetrans: The request of number of retransmission in boolean is OPTIONAL. An interface's average number of retransmission MAY be requested.

reqDelay: The request of delay in boolean is OPTIONAL. An interface's average delay MAY be requested.

reqPER: The request of packet error rate in boolean is OPTIONAL. An interface's average PER MAY be requested.

other: The CSAS MAY include additional parameters in the message. The CBSD MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.11.4 Spectrum Scan Requestor

The spectrum scan requestor is a report-requestor that CSAS asks for report of the device's spectrum environment. This report-requestor contains parameters to find out device's neighboring spectrum environment.

The SpecScanRequestor contains REQUIRED parameters.

freqRange: The frequency range of the scanning device is REQUIRED. The FrequencyRange is either in the form of start frequency and stop frequency in Hz or center frequency and bandwidth of the assigned spectrum.

nbins: The number of bins for the scanning is REQUIRED. The nbins decides how finely the spectrum is scanned.

other: The CSAS MAY include additional parameters in the message. The CBSD MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.12 CSAS Registration

In the tiered access protocol, protecting primary user is substantial. To obtain incumbent access information, the device MUST be registered before any transactions. The registration request message contains CSAS description and managing geographical region. When federal spectrum access system identifies the CSAS, the FSAS will issue the ticket for valid upcoming transactions. Since incumbents such military radars require high-level confidentiality, and leak of related information causes legal responsibility. How we protect the incumbent access activities from clients or CBSDs are dealt in Security Consideration (Section 7).

The CSAS registration request procedure is depicted in Figure 15.

- o REGISTRATION_REQ (Section 4.12.1) is the CSAS-registration request message
- o REGISTRATION_RESP (Section 4.12.2) is the CSAS-registration response message

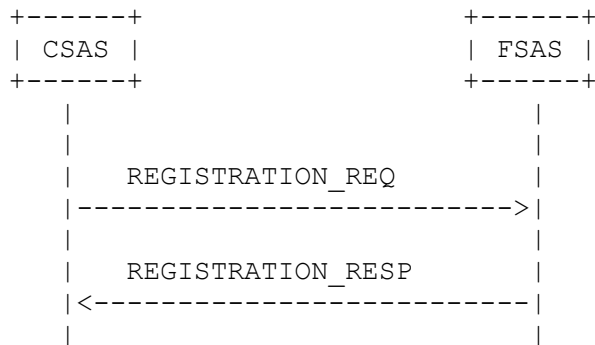


Figure 15

4.12.1 REGISTRATION_REQ

The CSAS MUST register to be served as a CSAS. Therefore, the CSAS

can be provided information related to IA from FSAS. The request message contains a descriptor that describes the CSAS's information, and the coverage region that defines governing region of the CSAS.

The CSAS registration request message contains REQUIRED parameters.

```
+-----+
|REGISTRATION_REQ          |
+-----+-----+
|csasDesc:ServerDescriptor  |required |
|coverageRegion:GeoLocation |required |
|.....|
|*other:any                 |optional |
+-----+-----+
```

Parameters:

csasDesc: The ServerDescriptor (Section 5.3) for the CSAS is REQUIRED. The FSAS identifies the CSAS's validity based on the ServerDescriptor which included in the request message.

coverageRegion: The GeoLocation (Section 5.1) of the CSAS's governing region is REQUIRED.

other: The CSAS may require additional initialization parameters. The FSAS MUST ignore all parameters it does not understand. Consult the PTASS Parameters Registry for possible additional parameters.

4.12.2 REGISTRATION_RESP

The FSAS checks the requested CSAS's validity, then confirms registration of the CSAS by sending a response message to the CSAS. In the response message, the FSAS issues a ticket for the CSAS.

The CSAS registration response message contains REQUIRED parameter.

```
+-----+
|REGISTRATION_RESP        |
+-----+-----+
|ticketInfo:TicketInfo    |required |
+-----+-----+
```

Parameter:

ticketInfo: A TicketInfo (Section 5.6) is REQUIRED. A ticket is issued to the CSAS.

4.13 Incumbent Notification

The FSAS MAY notify newly scheduled or updated IA's activity to the corresponding CSASs. The FSAS SHALL notify CSASs to evacuate lower-tiered devices from their exclusive zones for the purpose of IA's activity protection. This notification message contains information related to exclusive zones, operating frequency ranges, period of the activities, and additional information. If the schedule occurring repeatedly, the FSAS OPTIONALLY includes the weekly or monthly schedule for future activities. The schedule information MAY be complex due to the characteristics of the antenna, such as rotational antenna. The optimization of the sub-tiered devices is rely on the CSAS's optimization performance, the FSAS simply updates the newly added schedule(s) to the sole purpose of IA's protection.

The incumbent notify is one directional notification from the FSAS to the CSAS and this message contains REQUIRED parameters.

```

+-----+
|INCUMBENT_NOTIFY          |
+-----+-----+
|id:string                 |required|
|exclusiveZone:GeoLocation|required|
|freqRange:FrequencyRange |required| +-----+-----+
|scheduleInfo:ScheduleInfo|required|->|ScheduleInfo          |
|description:string        |optional| +-----+-----+
|.....|                    |timeRange:EventTime |required|
|*other:any                |optional| |expireDate:timestamp|required|
+-----+-----+ |repetition:bool     |optional|
|weekly:list               |optional|
|monthly:list              |optional|
+-----+-----+

```

Parameters:

id: The identification of the incumbent is REQUIRED. The incumbent ID MUST not carry any of its identity.

exclusizeZone: The exclusive zone defined by the GeoLocation(Section 5.1) is REQUIRED.

freqRange: The frequency range of the scheduled IA's activity is

REQUIRED. The FrequencyRange is either in the form of start frequency and stop frequency in Hz or center frequency and bandwidth of the assigned spectrum.

scheduleInfo: The schedule information of the newly updated ScheduleInfo(Section 4.13.1) of IA's activity is REQUIRED.

description: The description OPTIONALLY explains the newly updated schedule(s).

4.13.1 Schedule Information

The ScheduleInfo is designed to provide IA's scheduled activity. It is REQUIRED to contain timeRange in EventTime (Section 4.5.3) information which indicates initial schedule of the new schedule. The expireDate date in UTC time format is REQUIRED which indicates the expiration date of the new schedules even it repeats or not. It contains boolean that OPTIONALLY indicates this event is repeating or not, if "true", either "weekly" or "monthly" SHOULD indicate how this event is repeating. For example, the event that occurs every Monday, Wednesday, and Friday, the "weekly" can be set to the list of "Monday, Wednesday, Friday", or the event that occurs every 10th, 25th date of the month, the "monthly" can be set to the list of "10,25". The "weekly" or "monthly" can represent recurring event.

The ScheduleInfo contains REQUIRED parameters.

timeRange: The time range which indicates when this event is initially scheduled in the format of EventTime (Section 4.5.3) is REQUIRED.

expireDate: The expiration date is REQUIRED. The expire date and time in the format of UTC timestamp of this schedule.

repetition: The boolean repetition-indication is OPTIONAL. It MAY set to "true" to indicate this event is repeating, otherwise set to "false" for one time event.

weekly: If the schedule is repeating based on the weekly matter, the repetition schedule can be set to the list of "DAYS". For example, if the event is scheduled every Monday and Wednesday, the "weekly" can be set to "Monday, Wednesday".

monthly: If the schedule is repeating based on the monthly matter, the repetition schedule can be set to list of "DATES". For example, if the event is scheduled every 1st date and 15th date, the "monthly" can be set to "1,15".

4.14 Incumbent Query

The CSAS MAY ask exclusive region of the FSAS for the time. It is rarely occurred due to the schedule of IA's activity is synchronized between CSAS and FSAS. The FSAS returns a response that contains the list of IA information.

The incumbent request procedure is depicted in Figure 16.

- o INCUMBENT_REQ (Section 4.14.1) is the incumbent request message
- o INCUMBENT_RESP (Section 4.14.2) is the incumbent response message

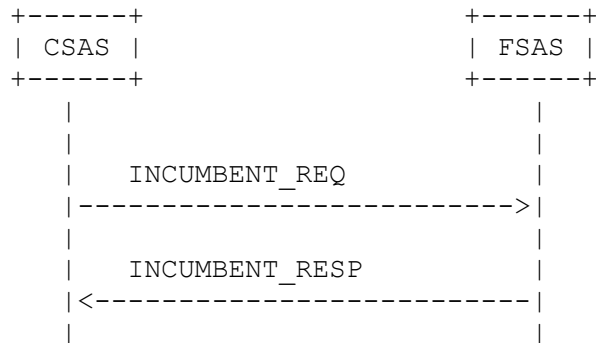
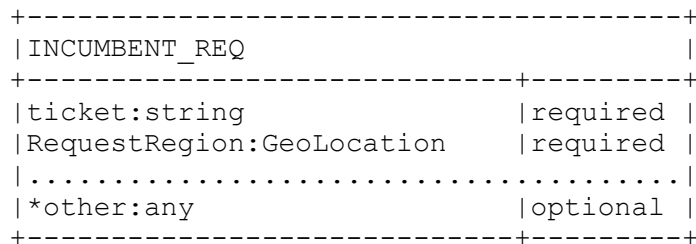


Figure 16

4.14.1 INCUMBENT_REQ

The incumbent request message contains REQUIRED parameters.



Parameters:

ticket: The ticket is REQUIRED. The corresponding ticket of the CSAS.

RequestRegion: The request region that the CSAS wants to know the IAs' information within this region (Section 5.1) is REQUIRED.

4.14.2 INCUMBENT_RESP

The FSAS responses with a message that contains the list of incumbent information within the requested area. Each entry has same parameters with INCUMBENT_NOTIFY (Section 4.13).

The incumbent response message contains REQUIRED parameter.

```

+-----+
|INCUMBENT_RESP          |
+-----+-----+ +-----+
|incumbentInfos:list|required|-->|IncumbentInfo          |
+-----+-----+ +-----+-----+
|                                     |required| |
|                                     |exclusizeZone:GeoLocation|required|
|                                     |freqRange:FrequencyRange |required|
|                                     |scheduleInfo:ScheduleInfo|required|
|                                     |description:string       |optional|
+-----+-----+ +-----+-----+
    
```

Parameters:

incumbentInfos: The list of IncumbentInfo is REQUIRED. Note that it is the same format to the INCUMBENT_NOTIFY (Section 4.13).

5 Protocol Parameters

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

5.1 GeoLocation

This parameter is used to specify the geo-location of the Device. It may be used to specify one of the following:

- o A single point with optional uncertainty
- o A region described by a polygon
- o A single 5+4 digit of zip+4 postal code, this MAY work for US only

The coordinates are expressed using the WGS84 datum [WGS-84], and units are degrees or meters. The parameter MAY also include a confidence level, expressed as a percentage. The confidence and uncertainty parameters may be required by some rulesets.

The data model for GeoLocation is illustrated below:

Ryoo, et al.

Version: Feb 1, 2015

[Page 44]

PTASS-Draft

ptass-protocol

February 1, 2015

```
+-----+
|GeoLocation          |
+-----+-----+
|point:Ellipse       |see description |
|region:Polygon      |see description |
|zip4code:string     |see description |
|confidence:int      |optional       |
+-----+-----+
```

Note: point, region and zip4code are mutually exclusive. Exactly one MUST be present.

```
+-----+
|Ellipse             |
+-----+-----+
|center:Point |required |----->|Point          |
+-----+-----+
|latitude:float |required |
|longitude:float |required |
+-----+-----+
```

```
+-----+
|Polygon            |
+-----+-----+
|exterior:list |required |----->|Point          |
+-----+-----+
|latitude:float |required |
|longitude:float |required |
+-----+-----+
```

Parameters:

point: If present, it indicates that the GeoLocation represents 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.

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].

exterior: When GeoLocation describes a region, the "exterior" field

Ryoo, et al.

Version: Feb 1, 2015

[Page 45]

PTASS-Draft

ptass-protocol

February 1, 2015

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.
- * 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).

zip4code: zip+4 postal code, according to the A950 Coding Accuracy Support System (CASS), it will file 100% of zip+4 code in near future, the 98% achieved at the time of writing this document, such as 07974-1276,g this locate exact building location. Note that zip code itself bears location information. Note that some devices MAY not equipped with the GPS or unable to operate the GPS due to the obstacles. This zip4code is an alternative of the GPS and we believe, it is the most appropriate locating method to the protocol and operation of the CBSD.

confidence: The location confidence level, as a percentage, MAY be provided. When the 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 StationDescriptor

The station descriptor contains parameters that identify the specific station, such as serialNumber, authority, owner of the device and ipAddress, portNumber to receive response message.

```

+-----+
|StationDescriptor          |
+-----+-----+
|authority:string          |required |
|owner:string              |required |
|serialNumber:string       |required |
|ipAddress:string          |required |
|portNumber:int            |required |
|.....|.....|
|*other:any                 |optional |
+-----+-----+

```

Parameters:

authority: The string of authority is REQUIRED, FCC as authority in US, alternatively ETSI or other.

owner: The string of owner is REQUIRED, owner of the device such as communication corporation which operates this station.

serialNumber: The manufacturer's device serial number is REQUIRED, the portal CSAS can check the validity of the device based on this serial number. The length of the value MUST NOT exceed 64 octets, conforming to the X.520 [ITUT.X520.2008] recommendations.

ipAddress: The string of IP address is REQUIRED, IP address to receive response message.

portNumber: The integer of port number is REQUIRED, port number associated with the IP address.

other: Depending on the ruleset, other parameters may be REQUIRED. The CBSD MUST ignore all parameters in the message it does not understand. Consult PTASS Parameters Registry for additional valid parameters and for the process for extending the message with more parameters.

Ryoo, et al.

Version: Feb 1, 2015

[Page 47]

PTASS-Draft

ptass-protocol

February 1, 2015

5.3 ServerDescriptor

The server descriptor of commercial spectrum access system contains parameters that identify the CSAS, such as id, authority, operator, ipAddress, and portNumber.

```
+-----+
|ServerDescriptor          |
+-----+-----+
|id:string                |required |
|authority:string         |required |
|operator:string          |required |
|ipAddress:string         |required |
|portNumber:int           |required |
|.....|.....|
|*other:any               |optional |
+-----+-----+
```

Parameters:

id: The ID for the CSAS is REQUIRED.

authority: The authority code for spectrum usage authority is REQUIRED.
The length of the value MUST NOT exceed 5 digit.

operator: The operator for the CSAS, the rullset operator.

ipAddress: The URL or IP address for CSAS connection is REQUIRED.

portNumber: The port number for corresponding CSAS connection is REQUIRED.

other: Depending on the ruleset, other parameters may be REQUIRED. The CBSD MUST ignore all parameters in the message it does not understand. Consult PTASS Parameters Registry for additional valid parameters and for the process for extending the message with more parameters.

5.4 Interface Descriptor

The interface descriptor of the citizens broadband radio service device contains parameters that identify the interfaces of the CBSD, such as id, antenna, and transmitting power output of the interface.

Ryoo, et al.

Version: Feb 1, 2015

[Page 48]

PTASS-Draft

ptass-protocol

February 1, 2015

```
+-----+
|InterfaceDescriptor      |
+-----+-----+
|id:string                |required |
|ant:AntennaChar         |required |
|txPwOutput:list         |required |
+-----+-----+
```

Parameters:

id: The ID for the interface is REQUIRED.

ant: The AntennaChar(Section 5.7), the antenna characteristics of interface in REQUIRED.

txPwOutput: The list of transmitting power level is REQUIRED. The list of integer for the transmitting power level is the controllable power level of the interface, therefore actual EIRP (Effective Isotropic Radiated Power) is this power level plus the antenna gain. The CSAS SHALL control the power level for its own purpose.

5.5 ScannerDescriptor

The scanner descriptor provides additional information to the CSAS, such as maker of the scanner, which model from the manufacturer, the minimum and maximum frequency for spectrum scanning, the instantaneous bandwidth Hz, and available web link for more information.

```

+-----+
|ScannerDescriptor      |
+-----+-----+
|maker:string          |required|
|model:string          |required|
|flf2:FreqRange       |required|
|InstBwHz:float       |required|
|link:string           |optional|
+-----+-----+

```

Parameters:

maker: The name of scanner provider is REQUIRED.

model: The model from the manufacturer is REQUIRED.

flf2: The minimum/maximum frequency supported by the scanner is REQUIRED. The supported frequency determines the capability of the

Ryoo, et al.

Version: Feb 1, 2015

[Page 49]

PTASS-Draft

ptass-protocol

February 1, 2015

device.

InstBwHz: The instantaneous bandwidth hertz of the scanner is REQUIRED. This value determines the speed of the device and it is two main parameters of the scanner on the market with the scanner's minimum and maximum frequency.

link: The web link about the scanner is OPTIONAL. When the scanner information is not established on the CSAS side, the CSAS MAY look it up this link for detail functionality.

5.6 TicketInfo

The ticket is issued by the CSAS based on the CBSD's interface information. It is essential to the CSAS to control the CBSD on interface level. The CBSD only can activate its service when this ticket is valid, and the CSAS can cease the CBSD's service by terminating this ticket. The ticket is only issued at the registration (Section 4.2), the terminated ticket is not reusable, and the interface MUST request the new ticket for terminated ticket by any reasons.

```

+-----+
|TicketInfo            |
+-----+-----+

```


- o antenna direction
- o antenna radiation pattern
- o antenna gain
- o antenna polarization

These are not defined by the base protocol, but may be added to the PTASS Parameters Registry, as needed.

5.8 Performance Result

The performance result is designed to provide a CBSD's performance to the CSAS, the service performance of a interface consists of multiple factors to reflect the up-to-date performance.

The PerformanceResult contains REQUIRED parameters.

```

+-----+
|PerformanceResult          |
+-----+-----+
|ticket:string              |required|
|timeRange:EventTime*      |required|
|nClient:int                |optional|
|avgTputMbps:float          |optional|
|avgNRetrans:float          |optional|
|avgDelayMs:float           |optional|
|avgPER:float               |optional|
+-----+-----+

```

Parameters:

ticket: The ticket is REQUIRED. A ticket of the interface.

timeRange: The time period of this report is REQUIRED. It is EventTime (Section 4.5.3) which indicates start and stop time of this report. Therefore a CSAS knows how long this report is collected from the start time.

nClient: The number of clients is OPTIONAL. The number of clients for being connected for the EventTime period MAY provide valuable information to the CSAS.

avgTputMbps: The average throughput in Mega bps is OPTIONAL. This provides average service quality of the interface.

avgNRetrans: The average number of retransmission is OPTIONAL.

avgDelayMs: The average delay in milliseconds is OPTIONAL.

avgPER: The average packet error rate is OPTIONAL.

The parameters of performance result can be defined differently by each CSAS in order to optimize performance and protect devices. It is difficult to define unified entries to cover a variety of underlying technology of CBSDs.

5.9 SpectrumScanResult

A spectrum scan result includes sensing data and related information. This report provides up-to-date spectrum environment around the device.

The SpecScan contains REQUIRED parameters.

```
+-----+-----+
|SpecScan                               |
+-----+-----+
|devSerialNum:string                    |required|
|timeRange:EventTime*                   |required|
|freqRange:FrequencyRange               |required|
|nbins:int                               |required|
|signalStrength:list                    |required|
+-----+-----+
```

Parameters:

devSerialNum: The device serial number is REQUIRED. The CSAS can check the scanning device to the registered scanning device in the device registration (Section 4.2).

timeRange: The time period of this report is REQUIRED.

freqRange: The frequency range of the scanning device is REQUIRED. The FrequencyRange is either in the form of start frequency and stop frequency in Hz or center frequency and bandwidth of the assigned spectrum.

nbins: The number of bins for the scanning is REQUIRED. The nbins decides how finely the spectrum is scanned.

signalStrength: The signal strength in dBm is REQUIRED. The list of scanned value is a set of floating numbers which indicates sensed signal strength for frequency bins.

5.10 Error Element

If the SAS (includes all type of SAS, FSAS and (portal)CSAS) or CBSD respond to a PTASS 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 and contain UTF-8 characters. The length of the value MUST NOT exceed 128 octets.

data: The SAS or CBSD MAY include additional data. For some errors, additional data may be required (see Table 1). The SAS or CBSD 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, consult PTASS Error Code Registry.

Code	Name	Description & Additional parameters
0	(reserved)	
-100	(reserved)	Indicates compatibility issues
-101	VERSION	The Database does not support the specified version of the message.
-102	UNSUPPORTED	The Database does not support the Device. For example, it does not support the ruleset specified in the request.
-103	UNIMPLEMENTED	The Database does not implement the optional request or optional feature.
-104	OUTSIDE_COVERAGE	The specified GeoLocation is outside the coverage area of the Database.
-105	DATABASE_CHANGE	The Database has changed its URI.
-200	(reserved)	Indicates that the device request contains an error
-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 5.17.1) 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.
-203	INVALID_MSG_TYPE	Invalid PTASS message type.
-300	(reserved)	Indicates authorization-related issues
-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.
-302	NOT_REGISTERED	Device registration required, but the Device is not registered.
-303	PREV_REGISTERED	Device already registered

-304	PREV_ACTIVE	Device already active
-305	NOT_REQUESTED	Spectrum not requested
-306	NO_AVAIL_LICENSE	No available license
-307	STORAGE_ERROR	Server storage error
-32000	(reserved)	Reserved for JSON-RPC error codes.
-32600	INVALID_REQUEST	Invalid request
-32601	METHOD_NOT_FOUND	Method not found
-32602	INVALID_PARAMS	Invalid parameters
-32603	INTERNAL_ERROR	Internal error
-32700	PARSE_ERROR	Parse error

Table 1: Error Codes

5.10.1 MISSING Error

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

```

+-----+
|Error          |
+-----+-----+
|code:int      |required |
|message:string|optional | +-----+
|data:ErrorData|required |--->|ErrorData          |
+-----+-----+ +-----+-----+ 1..*
                        |parameters:list |required |--+
                        +-----+-----+ |
                                                v
                                                string

```

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".

The PTASS protocol 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 provides the JSON schema for each of the protocol messages and parameters defined in sections Protocol Functionalities (Section 4) and Protocol Parameters (Section 5). JSON schemas are expressed using the format described by JSON Schema [I-D.zyp-json-schema], but are not intended to be used for formal validation.

NOTE: In general, all messages defined in this section are extensible by adding additional properties to support ruleset-specific and database-specific requirements. In all cases, the CBSD or SAS MUST ignore any parameter it does not understand.

NOTE: The JSON examples in this section 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:

- o The JSON-RPC Request object encapsulates a PTASS functionality (operation) and the request message
- o The JSON-RPC Response object encapsulates a PTASS response message and Error element

The SAS and CBSD MUST support JSON-RPC 2.0 encoding. A JSON-RPC is stateless, light-weighted remote procedure call (RPC) protocol [RFC 4627]. The PTASS uses request and response objects.

6.1.1 Request Object

The request object MUST have four members, the version of protocol and it MUST be "2.0", the name of method to be invoked as string, parameter values for the method, and an identifier for request-response transaction. A request object without an "id" member is the

notification, such as PERF_REPORT_NOTIFY, TERMINATION_NOTIFY in the PTASS.

The example form of PTASS request message

```
{
```

```

    "jsonrpc" : "2.0",
    "method" : "spectrum.ptass.methodName",
    "params" : <PTASS_REQ>,
    "id" : "idString"
}

```

where "method" is the name of a PTASS functionality (operation), and <PTASS_REQ> represents one of the PTASS request objects associated with the method. Method names are defined with the prefix, for example "spectrum.ptass.init".

6.1.2 Response Object

The response object consists of four members and only three members MUST be used. The members are, the version of protocol and it MUST be "2.0", result values of the invoked method which REQUIRE on success, and MUST not exist on error, an indication of error during the method process which REQUIRE on error, and MUST not on success, an identifier for request-response transaction which MUST be the same with the id in the request object.

The form of PTASS response message

```

{
    "jsonrpc" : "2.0",
    "result" : <PTASS_RESP>,
    "id" : "idString"
}
or
{
    "jsonrpc" : "2.0",
    "error" : {
        "code" : -102,
        "message" : "corresponding error message"
    },
    "id" : "idString"
}

```

where the Error object and error codes are described by Error Element (Section 5.10). The CSAS SHOULD attempt to use the most specific applicable PTASS 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 CSAS receives invalid JSON from the Device, it should respond with "-32700", signifying a parse

error. As a last resort, the CSAS MAY send a suitable HTTP 5xx response. Depending on prior arrangement between a SAS and CBSD, the Request and Response objects MAY contain additional parameters. The SAS or CBSD MUST ignore all parameters it does not understand.

6.2 init Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.init" method that represents the Initialization functionality (Section 4.1).

6.2.1 INIT_REQ Parameters

The JSON encoding of the Initialization request message INIT_REQ (Section 4.1.2) is described by the following schema:

```
{
  "name" : "INIT_REQ",
  "type" : "object",
  "properties" : {
    "type" : "INIT_REQ",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "stationDesc" : {
      "type" : "stationDescriptor",
      "required" : True
    },
    "location" : {
      "type" : "GeoLocation",
      "required" : True
    }
  }
}
```

Example "init" JSON-RPC request:

```
{
  "jsonrpc" : "2.0",
  "method" : "spectrum.ptass.init",
  "params" : {
    "type" : "INIT_REQ",
    "version" : "1.0",
    "stationDesc" : {
```

```

        "authority" : "FCC",
        "owner" : "companyName",
        "serialNumber" : "IDC1710001",
        "ipAddress" : "xxx.xxx.xxx.xxx",
        "portNumber" : xxxx
    },
    "location" : {
        "point" : {
            "center" : {
                "latitude" : 40.686911,
                "longitude" : -74.397187
            }
        }
    }
},
"id" : "XXXXXX"
}

```

6.2.2 INIT_RESP Parameters

The JSON encoding of the Initialization response message INIT_RESP (Section 4.1.3) is described by the following schema:

```

{
  "name" : "INIT_RESP",
  "type" : "object",
  "properties" : {
    "type" : "INIT_RESP",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "csasDesc" : {
      "type" : "ServerDescriptor",
      "required" : True
    }
  }
}

```

Example "init" JSON-RPC response:

```

{
  "jsonrpc" : "2.0",
  "result" : {
    "type" : "INIT_RESP",
    "version" : "1.0",
  }
}

```

```
    "csasDesc" : {
      "id" : "YYYYYY",
      "authority" : "FCC",
      "operator" : "Alcatel-Lucent Bell-Labs",
      "ipAddress" : "xxx.xxx.xxx.xxx",
      "portNumber": xxxx
    }
  },
  "id" : "XXXXXX"
}
```

6.3 register Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.register" method that represents the Registration functionality (Section 4.2).

6.3.1 REGISTRATION_REQ Parameters

The JSON encoding of the Registration request message REG_REQ (Section 4.2.1) is described by the following schema:

```
{
  "name" : "REGISTRATION_REQ",
  "type" : "object",
  "properties" : {
    "type" : "REGISTRATION_REQ",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "stationDesc" : {
      "type" : "stationDescriptor",
      "required" : True
    },
    "location" : {
      "type" : "GeoLocation",
      "required" : True
    },
    "interfaces" : {
      "type" : "InterfaceDescriptor",
      "required" : True
    },
    "licenses" : {
      "type" : "string",
      "required" : False
    }
  }
}
```

PTASS-Draft

ptass-protocol

February 1, 2015

```

    },
    "scannerDesc" : {
      "type" : "ScannerDescriptor",
      "required" : False
    }
  }
}

```

Example "register" JSON-RPC request:

```

{
  "jsonrpc" : "2.0",
  "method" : "spectrum.ptass.register",
  "params" : {
    "type" : "REGISTRATION_REQ",
    "version" : "1.0",
    "stationDesc" : {
      "authority" : "FCC",
      "owner" : "companyName",
      "serialNumber" : "IDC1710001",
      "ipAddress" : "xxx.xxx.xxx.xxx",
      "portNumber" : xxxx
    },
    "location" : {
      "point" : {
        "center" : {
          "latitude" : 40.686911,
          "longitude" : -74.397187
        }
      }
    },
    "interfaces" : [
      {
        "id" : "ALUIDC1710001",
        "antenna" : {"height" : 10.2, "base" : AGL},
        "txPowerOutputs" : [10,20,30]
      },
      {
        "id" : "ALUIDC1710002",
        "antenna" : {"height" : 10.2 , "base" : AGL},
        "txPowerOutputs" : [10,15,20,25,30]
      },
      .....
    ],
    "licenses" : ["FCCPALIDC1710001", "FCCPALIDC1710002"],
    "scannerDesc" : {
      "maker" : "thinkrf",
      "model" : "WSA4000",

```


PTASS-Draft

ptass-protocol

February 1, 2015

```

        "flf2" : {
            "startHz" : 100000.0,
            "stopHz" : 1.0E+10
        },
        "InstBwHz" : 100E+6
        "link" : "http://thinkrf.com/wsa4000/"
    }
},
"id" : "XXXXXX"
}

```

6.3.2 REGISTRATION_RESP Parameters

The JSON encoding of the Registration response message REG_RESP (Section 4.2.2) is described by the following schema:

```

{
  "name" : "REGISTRATION_RESP",
  "type" : "object",
  "properties" : {
    "type" : "REGISTRATION_RESP",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "ticketInfos" : {
      "type" : "TicketInfo",
      "required" : True
    }
  }
}

```

Example "register" JSON-RPC response:

```

{
  "jsonrpc" : "2.0",
  "result" : {
    "type" : "REGISTRATION_RESP",
    "version" : "1.0",
    "ticketInfos" : [
      {
        "id" : "ALUIDC1710001",
        "ticket" : "ab9d5af",

```

```
    "expireDate" : "2014-09-03T10:56:35Z"
  },
  {
```

Ryoo, et al.

Version: Feb 1, 2015

[Page 62]

PTASS-Draft

ptass-protocol

February 1, 2015

```
    "id" : " ALUIDC1710002",
    "ticket" : "574f614",
    "expireDate" : "2014-09-03T10:56:35Z"
  },
  .....
]
},
"id" : "XXXXXX"
}
```

6.4 getSpectrum Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.getSpectrum" method that represents the Available Spectrum Query functionality (Section 4.3).

6.4.1 AVAIL_SPECTRUM_REQ Parameters

The JSON encoding of the Available spectrum request message AVAIL_SPECTRUM_REQ (Section 4.3.1) is described by the following schema:

```
{
  "name" : "AVAIL_SPECTRUM_REQ",
  "type" : "object",
  "properties" : {
    "type" : "AVAIL_SPECTRUM_REQ",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "ticket" : {
      "type" : "string",
      "required" : True
    },
    "bandwidthHz" : {
      "type" : "float",
      "required" : True
    },
    "useLicense" : {
      "type" : "boolean",
```

```
        "required" : True
    }
}
}
```

Ryoo, et al.

Version: Feb 1, 2015

[Page 63]

PTASS-Draft

ptass-protocol

February 1, 2015

Example "getSpectrum" JSON-RPC request:

```
{
  "jsonrpc" : "2.0",
  "method" : "spectrum.ptass.getSpectrum",
  "params" : {
    "type" : "AVAIL_SPECTRUM_REQ",
    "version" : "1.0",
    "ticket" : "ab9d5af",
    "bandwidthHz" : 10000000,
    "usesLicense" : True
  },
  "id" : "XXXXXX"
}
```

6.4.2 AVAIL_SPECTRUM_RESP Parameters

The JSON encoding of the Available spectrum response message AVAIL_SPECTRUM_RESP (Section 4.3.2) is described by the following schema:

```
{
  "name" : "AVAIL_SPECTRUM_RESP",
  "type" : "object",
  "properties" : {
    "type" : "AVAIL_SPECTRUM_RESP",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "ticket" : {
      "type" : "string",
      "required" : True
    },
    "freqRange" : {
      "type" : "FrequencyRange",

```

```

        "required" : True
    },
    "txPowerOutput" : {
        "type" : "integer",
        "required" : False
    }
}
}
}

```

Ryoo, et al.

Version: Feb 1, 2015

[Page 64]

PTASS-Draft

ptass-protocol

February 1, 2015

Example "getSpectrum" JSON-RPC response:

```

{
  "jsonrpc" : "2.0",
  "result" : {
    "type" : "AVAIL_SPECTRUM_RESP",
    "version" : "1.0",
    "freqRange" : {
      "startHz" : 3.55E+9,
      "stopHz" : 3.56E+9
    },
    "txPowerOutput" : 30,
    "ticket" : "ab9d5af"
  },
  "id" : "XXXXXXX"
}

```

6.5 useSpectrum Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.useSpectrum" method that represents the Spectrum usage Query functionality (Section 4.4).

6.5.1 SPECTRUM_USE_REQ Parameters

The JSON encoding of the Spectrum usage request message SPECTRUM_USE_REQ (Section 4.4.1) is described by the following schema:

```

{
  "name" : "SPECTRUM_USE_REQ",
  "type" : "object",
  "properties" : {
    "type" : "SPECTRUM_USE_REQ",
    "version" : {

```

```

        "type" : "string",
        "required" : True
    },
    "ticket" : {
        "type" : "string",
        "required" : True
    }
}
}

```

Example "useSpectrum" JSON-RPC request:

Ryoo, et al.

Version: Feb 1, 2015

[Page 65]

PTASS-Draft

ptass-protocol

February 1, 2015

```

{
  "jsonrpc" : "2.0",
  "method" : "spectrum.ptass.useSpectrum",
  "params" : {
    "type" : "SPECTRUM_USE_REQ",
    "version" : "1.0",
    "ticket" : "ab9d5af"
  },
  "id" : "XXXXXX"
}

```

6.5.2 SPECTRUM_USE_RESP Parameters

The JSON encoding of the Spectrum usage response message SPECTRUM_USE_RESP (Section 4.4.2) is described by the following schema:

```

{
  "name" : "SPECTRUM_USE_RESP",
  "type" : "object",
  "properties" : {
    "type" : "SPECTRUM_USE_RESP",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "activate" : {
      "type" : "boolean",
      "required" : True
    },
    "ticket" : {
      "type" : "string",

```

```

        "required" : True
    }
}

```

Example "useSpectrum" JSON-RPC response:

```

{
  "jsonrpc" : "2.0",
  "result" : {
    "type" : "SPECTRUM_USE_RESP",
    "version" : "1.0",
    "activate" : True,
    "ticket" : "ab9d5af"
  }
}

```

Ryoo, et al.

Version: Feb 1, 2015

[Page 66]

PTASS-Draft

ptass-protocol

February 1, 2015

```

    },
    "id" : "XXXXXX"
}

```

6.6 report Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.report" method that represents the performance report Notify functionality (Section 4.5).

6.6.1 PERF_REPORT_NOTIFY

The JSON encoding of the Performance report notification message PERF_REPORT_NOTIFY (Section 4.5.1) is described by the following schema:

```

{
  "name" : "PERF_REPORT_NOTIFY",
  "type" : "object",
  "properties" : {
    "type" : "PERF_REPORT_NOTIFY",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "perfResults" : {
      "type" : "PerformanceResult",
      "required" : True
    },
    "scanResult" : {

```

```

        "type" : "SpecScan",
        "required" : False
    }
}
}

```

Example "report" JSON-RPC notification:

```

{
  "jsonrpc" : "2.0",
  "method" : "spectrum.ptass.report",
  "params" : {
    "type" : "PERF_REPORT_NOTIFY",
    "version" : "1.0",
    "perfResults" : [

```

Ryoo, et al.

Version: Feb 1, 2015

[Page 67]

PTASS-Draft

ptass-protocol

February 1, 2015

```

{
  "ticket" : "ab9d5af",
  "timeRange" : {
    "startTime" : "2014-08-29T14:25:00Z",
    "stopTime" : "2014-08-29T14:55:00Z",
  },
  "nClient" : 5,
  "avgTputMbps" : 1.5,
  "avgNRetrans" : 1.0,
  "avgDelayMs" : 100.0,
  "avgPER" : 0.02
},
{
  "ticket" : "574f614",
  "timeRange" : {
    "startTime" : "2014-08-29T14:25:00Z",
    "stopTime" : "2014-08-29T14:55:00Z",
  },
  "nClient" : 12,
  "avgTputMbps" : 1.7,
  "avgNRetrans" : 1.2,
  "avgDelayMs" : 80.0,
  "avgPER" : 0.01
},
.....
],
"scanResult" : {

```

```

    "devSerialNum" : "IDC1710001WSAK"
    "timeRange" : {
      "startTime" : "2014-08-29T14:25:00Z",
      "stopTime" : "2014-08-29T14:55:00Z",
    },
    "freqRange" : {
      "startHz" : 3.55E+9,
      "stopHz" : 3.65E+9,
    },
    "nbins": 100000,
    "signalStrength" : [
      -77.0, -77.2, -77.1, ...
      -77.0, -77.3, -77.0
    ]
  }
}
}
}

```

6.7 termination Method

This section describes the encoding for the JSON-RPC

Ryoo, et al.

Version: Feb 1, 2015

[Page 68]

PTASS-Draft

ptass-protocol

February 1, 2015

"spectrum.ptass.notifyTermination" method that represents the termination Notify functionality (Section 4.6).

6.7.1 TERMINATION_NOTIFY

The JSON encoding of the Termination notification message TERMINATION_NOTIFY (Section 4.6.1) is described by the following schema:

```

{
  "name" : "TERMINATION_NOTIFY",
  "type" : "object",
  "properties" : {
    "type" : "TERMINATION_NOTIFY",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "tickets" : {
      "type" : "string",
      "required" : True
    },
    "code" : {

```



```

        "type" : "integer",
        "required" : False
    }
}

```

Example "notifyTermination" JSON-RPC notification:

```

{
  "jsonrpc" : "2.0",
  "method" : "spectrum.ptass.notifyTermination",
  "params" : {
    "type" : "TERMINATION_NOTIFY",
    "version" : "1.0",
    "code" : KKK,
    "tickets" :
    [
      "ab9d5af",
      "574f614",
      "4338526"
    ]
  }
}

```

6.8 renewal Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.requestRenewal" method that represents the ticket renewal Query functionality (Section 4.7).

6.8.1 TICKET_RENEWAL_REQ Parameters

The JSON encoding of the Ticket renewal request message TICKET_RENEW_REQ (Section 4.7.1) is described by the following schema:

```

{
  "name" : "TICKET_RENEW_REQ",
  "type" : "object",
  "properties" : {
    "type" : "TICKET_RENEW_REQ",
    "version" : {
      "type" : "string",
      "required" : True
    }
  },

```

```

        "ticket" : {
            "type" : "string",
            "required" : True
        },
        "expireDate" : {
            "type" : "string",
            "required" : True
        }
    }
}

```

Example "requestTicketRenew" JSON-RPC request:

```

{
    "jsonrpc" : "2.0",
    "method" : "spectrum.ptass.requestRelocation",
    "params" : {
        "type" : "RELOCATION_REQ",
        "version" : "1.0",
        "ticket" : "ab9d5af",
        "expireDate" : "2014-10-07T16:20:00Z"
    },
    "id" : "XXXXXX"
}

```

6.8.2 TICKET_RENEWAL_RESP Parameters

The JSON encoding of the Ticket renewal response message TICKET_RENEW_RESP (Section 4.7.2) is described by the following schema:

```

{
    "name" : "TICKET_RENEW_RESP",
    "type" : "object",
    "properties" : {
        "type" : "TICKET_RENEW_RESP",
        "version" : {
            "type" : "string",
            "required" : True
        },
        "isRenewed" : {
            "type" : "boolean",
            "required" : True
        }
    }
}

```

```

    },
    "ticket" : {
        "type" : "string",
        "required" : True
    },
    "expireDate" : {
        "type" : "string",
        "required" : False
    },
    "code" : {
        "type" : "integer",
        "required" : False
    }
}
}
}

```

Example "requestTicketRenew" JSON-RPC response:

```

{
    "jsonrpc" : "2.0",
    "result" : {
        "type" : "TICKET_RENEW_RESP",
        "version" : "1.0",
        "isRenewed" : True,
        "ticket" : "ab9d5af",
        "expireDate" : "2014-10-08T16:20:00Z"
    },
    "id" : "XXXXXXX"
}

```

6.9 relocation Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.requestRelocation" method that represents the relocation Query functionality (Section 4.8).

6.9.1 RELOCATION_REQ Parameters

The JSON encoding of the Relocation request message RELOCATION_REQ (Section 4.8.1) is described by the following schema:

```

{
    "name" : "RELOCATION_REQ",
    "type" : "object",
    "properties" : {

```

```

    "type" : "RELOCATION_REQ",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "location" : {
      "type" : "GeoLocation",
      "required" : True
    },
    "stationDesc" : {
      "type" : "stationDescriptor",
      "required" : True
    },
    "code" : {
      "type" : "integer",
      "required" : False
    }
  }
}

```

Example "requestRelocation" JSON-RPC request:

```

{
  "jsonrpc" : "2.0",
  "method" : "spectrum.ptass.requestRelocation",
  "params" : {
    "type" : "RELOCATION_REQ",
    "version" : "1.0",
    "location" : {
      "point" : {

```

```

      "center" : {
        "latitude" : 40.629911,
        "longitude" : -74.373187
      }
    },
    "stationDesc" : {
      "authority" : "FCC",
      "owner" : "companyName",
      "serialNumber" : "IDC1710001",
      "ipAddress" : "xxx.xxx.xxx.xxx",
      "portNumber" : xxxx
    }
  }
}

```

```

    },
    "code" : KKK
  },
  "id" : "XXXXXX"
}

```

6.9.2 RELOCATION_RESP Parameters

The JSON encoding of the Relocation response message RELOCATION_RESP (Section 4.8.2) is described by the following schema:

```

{
  "name" : "RELOCATION_RESP",
  "type" : "object",
  "properties" : {
    "type" : "RELOCATION_RESP",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "isValid" : {
      "type" : "boolean",
      "required" : True
    }
  }
}

```

Example "requestRelocation" JSON-RPC response:

```

{
  "jsonrpc" : "2.0",
  "result" : {

```

```

    "type" : "RELOCATION_RESP",
    "version" : "1.0",
    "isValid" : False,
  },
  "id" : "XXXXXX"
}

```

6.10 reassignment Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.reassignment" method that represents the reassignment Query functionality (Section 4.9).

6.10.1 REASSIGNMENT_REQ Parameters

The JSON encoding of the Reassignment request message REASSIGNMENT_REQ (Section 4.9.1) is described by the following schema:

```
{
  "name" : "REASSIGNMENT_REQ",
  "type" : "object",
  "properties" : {
    "type" : "REASSIGNMENT_REQ",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "tickets" : {
      "type" : "string",
      "required" : True
    },
    "timestamp" : {
      "type" : "string",
      "required" : True
    },
    "freqRange" : {
      "type" : "FrequencyRange",
      "required" : True
    },
    "txPowerOutput" : {
      "type" : "integer",
      "required" : True
    },
    "code" : {
      "type" : "integer",
      "required" : False
    }
  }
}
```

```
}
}
```

Example "reassignment" JSON-RPC request:

```
{
```

```

"jsonrpc" : "2.0",
"method" : "spectrum.ptass.reassignment",
"params" : {
  "type" : "REASSIGNMENT_REQ",
  "version" : "1.0",
  "ticket" : "ab9d5af",
  "timestamp" : "2014-10-07T16:20:00Z",
  "freqRange" : {
    "startHz" : 3.55E+9
    "stopHz" : 3.56E+9
  },
  "txPowerOutput" : 30,
  "code" : KKK
},
"id" : "XXXXXX"
}

```

6.10.2 REASSIGNMENT_RESP Parameters

The JSON encoding of the Reassignment response message REASSIGNMENT_RESP (Section 4.9.2) is described by the following schema:

```

{
  "name" : "REASSIGNMENT_RESP",
  "type" : "object",
  "properties" : {
    "type" : "REASSIGNMENT_RESP",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "ticket" : {
      "type" : "string",
      "required" : True
    },
    "isReassigned" : {
      "type" : "boolean",
      "required" : True
    }
  }
}

```

Example "reassignment" JSON-RPC response:

```

{
  "jsonrpc" : "2.0",
  "result" : {
    "type" : "REASSIGNMENT_RESP",
    "version" : "1.0",
    "isReassigned" : True,
    "ticket" : "ab9d5af"
  },
  "id" : "10021231"
}

```

6.11 deactivation Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.deactivation" method that represents the deactivation Query functionality (Section 4.10).

6.11.1 DEACTIVATION_REQ Parameters

The JSON encoding of the Deactivation request message DEACTIVATION_REQ (Section 4.10.1) is described by the following schema:

```

{
  "name" : "DEACTIVATION_REQ",
  "type" : "object",
  "properties" : {
    "type" : "DEACTIVATION_REQ",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "ticket" : {
      "type" : "string",
      "required" : True
    },
    "timestamp" : {
      "type" : "string",
      "required" : True
    },
    "code" : {
      "type" : "integer",

```



```

        "required" : False
    }
}

```

Example "deactivation" JSON-RPC request:

```

{
  "jsonrpc" : "2.0",
  "method" : "spectrum.ptass.deactivation",
  "params" : {
    "type" : "DEACTIVATION_REQ",
    "version" : "1.0",
    "ticket" : "ab9d5af",
    "timestamp" : "2014-10-07T16:20:00Z",
    "code" : KKK
  },
  "id" : "XXXXXX"
}

```

6.11.2 DEACTIVATION_RESP Parameters

The JSON encoding of the Deactivation response message DEACTIVATION_RESP (Section 4.10.2) is described by the following schema:

```

{
  "name" : "DEACTIVATION_RESP",
  "type" : "object",
  "properties" : {
    "type" : "DEACTIVATION_RESP",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "ticket" : {
      "type" : "string",
      "required" : True
    },
    "isDeactivated" : {

```

```
        "type" : "boolean",
        "required" : True
    }
}
```

Example "deactivation" JSON-RPC response:

```
{
  "jsonrpc" : "2.0",
  "result" : {
    "type" : "DEACTIVATION_RESP",
    "version" : "1.0",
    "isDeactivated" : True,
    "ticket" : "ab9d5af"
  },
  "id" : "XXXXXX"
}
```

6.12 report Query Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.getReport" method that represents the Performance report Query functionality (Section 4.11).

6.12.1 PERF_REPORT_REQ Parameters

The JSON encoding of the Performance report request message PERF_REPORT_REQ (Section 4.11.1) is described by the following schema:

```
{
  "name" : "PERF_REPORT_REQ",
  "type" : "object",
  "properties" : {
    "type" : "PERF_REPORT_REQ",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "repetitionMin" : {
      "type" : "integer",
      "required" : True
    }
  },
}
```

```

    "perfReportReqs" : {
      "type" : "PerformanceReportRequestor",
      "required" : True
    },
    "scanReq" : {
      "type" : "SpecScanRequestor",
      "required" : False
    }
  }
}

```

Example "getReport" JSON-RPC request:

```

{
  "jsonrpc" : "2.0",
  "method" : "spectrum.ptass.getReport",
  "params" : {
    "type" : "PERF_REPORT_REQ",
    "version" : "1.0",
    "repetitionMinute" : 60,
    "perfReportReqs" : [
      {
        "ticket" : "ab9d5af",
        "reqTput" : True,
        "reqNRetrans" : False,
        "reqDelay" : False,
        "reqPER" : True
      },
      .....
    ],
    "scanReq" : {
      "freqRange" : {
        "startHz" : 3.55E+9,
        "stopHz" : 3.65E+9,
      },
      "nbins": 10000
    }
  },
  "id" : "XXXXXXX"
}

```

6.12.2 PERF_REPORT_RESP Parameters

The JSON encoding of the Performance report response message

PERF_REPORT_RESP (Section 4.11.2) is described by the following schema:

```
{
  "name" : "PERF_REPORT_RESP",
  "type" : "object",
  "properties" : {
    "type" : "PERF_REPORT_RESP",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "perfResults" : {
      "type" : "PerformanceResult",
      "required" : True
    },
    "scanResult" : {
      "type" : "SpecScan",
      "required" : False
    }
  }
}
```

Example "getReport" JSON-RPC response:

```
{
  "jsonrpc" : "2.0",
  "result" : {
    "type" : "PERF_REPORT_RESP",
    "version" : "1.0",
    "perfResults" : [
      {
        "ticket" : "ab9d5af",
        "timeRange" : {
          "startTime" : "2014-08-29T14:25:00Z",
          "stopTime" : "2014-08-29T14:55:00Z",
        },
        "nClient" : 5,
        "avgTputMbps" : 1.5,
        "avgNRetrans" : 1.0,
        "avgDelayMs" : 100.0,
        "avgPER" : 0.02
      },
    ],
  },
}
```

```
{
  "ticket" : "574f614",
  "timeRange" : {
```

Ryoo, et al.

Version: Feb 1, 2015

[Page 80]

PTASS-Draft

ptass-protocol

February 1, 2015

```
    "startTime" : "2014-08-29T14:25:00Z",
    "stopTime" : "2014-08-29T14:55:00Z",
  },
  "nClient" : 12,
  "avgTputMbps" : 1.7,
  "avgNRetrans" : 1.2,
  "avgDelayMs" : 80.0,
  "avgPER" : 0.01
},
.....
],
"scanResult" : {
  "devSerialNum" : "IDC1710001WSAK"
  "timeRange" : {
    "startTime" : "2014-08-29T14:25:00Z",
    "stopTime" : "2014-08-29T14:55:00Z",
  },
  "freqRange" : {
    "startHz" : 3.55E+9,
    "stopHz" : 3.65E+9,
  },
  "nbins": 100000,
  "signalStrength" : [
    -77.0, -77.2, -77.1, ...
    -77.0, -77.3, -77.0
  ]
}
},
"id" : "XXXXXXX"
}
```

6.13 CSAS registration Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.csasRegister" method that represents the CSAS registration Query functionality (Section 4.12).

6.13.1 CSAS_REGISTRATION_REQ Parameters

The JSON encoding of the CSAS registration request message CSAS_REGISTRATION_REQ (Section 4.12.1) is described by the following schema:

```
{
  "name" : "CSAS_REGISTRATION_REQ",
  "type" : "object",
  "properties" : {
    "type" : "CSAS_REGISTRATION_REQ",
```

Ryoo, et al.

Version: Feb 1, 2015

[Page 81]

PTASS-Draft

ptass-protocol

February 1, 2015

```
    "version" : {
      "type" : "string",
      "required" : True
    },
    "csasDesc" : {
      "type" : "ServerDescriptor",
      "required" : True
    },
    "coverageRegion" : {
      "type" : "GeoLocation",
      "required" : True
    }
  }
}
```

Example "csasRegistration" JSON-RPC request:

```
{
  "jsonrpc" : "2.0",
  "method" : "spectrum.ptass.csasRegister",
  "params" : {
    "type" : "CSAS_REGISTRATION_REQ",
    "version" : "1.0",
    "csasDesc" : {
      "id" : "csas_XXXXXX",
      "authority" : "FCC",
      "operator" : "Alcatel-Lucent Bell-Labs",
    },
    "ipAddress" : "xxx.xxx.xxx.xxx",
    "portNumber": XXXX
  },
  "coverageRegion" : {
    "region":{
      "exterior":[
        {"latitude":43.25742136,"longitude":-70.0636},
        {"latitude":43.25728263,"longitude":-70.04204977},
        {"latitude":43.25686576,"longitude":-70.02050640},
        .....
        {"latitude":43.25728263,"longitude":-70.08515022}
      ]
    }
  }
}
```

```

        ]
      }
    },
    "id" : "XXXXXXX"
  }
}

```

6.13.2 CSAS_REGISTRATION_RESP Parameters

Ryoo, et al.

Version: Feb 1, 2015

[Page 82]

PTASS-Draft

ptass-protocol

February 1, 2015

The JSON encoding of the CSAS registration response message CSAS_REGISTRATION_RESP (Section 4.12.2) is described by the following schema:

```

{
  "name" : "CSAS_REGISTRATION_RESP",
  "type" : "object",
  "properties" : {
    "type" : "CSAS_REGISTRATION_RESP",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "ticketInfos" : {
      "type" : "TicketInfo",
      "required" : True
    }
  }
}

```

Example "csasRegistration" JSON-RPC response:

```

{
  "jsonrpc" : "2.0",
  "result" : {
    "type" : "CSAS_REGISTRATION_RESP",
    "version" : "1.0",
    "ticketInfo" : {
      "id" : "csas_xxxxxx",
      "ticket" : "e51ba6",
      "expireDate" : "2014-09-03T10:56:35Z",
    }
  },
}

```

```
    "id" : "XXXXXX"
  }
```

6.14 incumbent notification Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.notifyIncumbent" method that represents the incumbent Notify functionality (Section 4.13).

6.14.1 INCUMBENT_NOTIFY

Ryoo, et al.

Version: Feb 1, 2015

[Page 83]

PTASS-Draft

ptass-protocol

February 1, 2015

The JSON encoding of the Incumbent notification message INCUMBENT_NOTIFY (Section 4.13.1) is described by the following schema:

```
{
  "name" : "INCUMBENT_NOTIFY",
  "type" : "object",
  "properties" : {
    "type" : "INCUMBENT_NOTIFY",
    "version" : {
      "type" : "string",
      "required" : True
    },
    "id" : {
      "type" : "string",
      "required" : True
    },
    "exclusiveZone" : {
      "type" : "GeoLocation",
      "required" : True
    },
    "freqRange" : {
      "type" : "FrequencyRange",
      "required" : True
    },
    "scheduleInfo" : {
      "type" : "ScheduleInfo",
      "required" : True
    },
    "description" : {
      "type" : "string",
      "required" : False
    }
  }
}
```



```
}
}
```

Example "notifyIncumbent" JSON-RPC notification:

```
{
  "jsonrpc": "2.0",
  "method": "spectrum.ptass.notifyIncumbent",
  "params": {
    "type": "INCUMBENT_NOTIFY",
    "version": "1.0",
    "id": "xxxxx",
    "exclusiveZone": {
      "region": {
        "exterior": [
```

Ryoo, et al.

Version: Feb 1, 2015

[Page 84]

PTASS-Draft

ptass-protocol

February 1, 2015

```
    {"latitude":43.25742136,"longitude":-70.0636},
    {"latitude":43.25728263,"longitude":-70.04204977},
    {"latitude":43.25686576,"longitude":-70.02050640},
    .....
    {"latitude":43.25728263,"longitude":-70.08515022}
  ]
}
},
"freqRange":{
  "startHz":3.55E+9,
  "stopHz":3.6E+9
},
"scheduleInfo":{
  "timeRange":{
    "startTime":"16:00:00",
    "stopTime":"20:00:00"
  },
  "expireDate": "2015-09-03T10:56:35Z",
  "repetition": True,
  "weekly": {"Monday", "Wednesday"},
  "monthly": {}
},
"description":"NEW INCUMBENT ACTIVITY SCHEDULED"
}
}
```

6.15 incumbent access Query Method

This section describes the encoding for the JSON-RPC "spectrum.ptass.getIncumbent" method that represents the incumbent access Query functionality (Section 4.14).

6.15.1 INCUMBENT_REQ Parameters

The JSON encoding of the incumbent access request message INCUMBENT_REQ (Section 4.14.1) is described by the following schema:

```
{
  "name" : "INCUMBENT_REQ",
  "type" : "object",
  "properties" : {
    "type" : "INCUMBENT_REQ",
    "version" : {
      "type" : "string",
      "required" : True
    }
  }
}
```

Ryoo, et al.

Version: Feb 1, 2015

[Page 85]

PTASS-Draft

ptass-protocol

February 1, 2015

```
    },
    "id" : {
      "type" : "string",
      "required" : True
    },
    "RequestRegion" : {
      "type" : "GeoLocation",
      "required" : True
    }
  }
}
```

Example "getIncumbent" JSON-RPC request:

```
{
  "jsonrpc": "2.0",
  "method": "spectrum.ptass.getIncumbent",
  "params": {
    "type": "INCUMBENT_REQ",
    "version": "1.0",
    "RequestRegion": {
      "region": {
        "exterior": [
          {"latitude": 43.25742136, "longitude": -70.0636},

```

```

        {"latitude":43.25728263,"longitude":-70.04204977},
        {"latitude":43.25686576,"longitude":-70.02050640},
        .....
        {"latitude":43.25728263,"longitude":-70.08515022}
    ]
    }
    },
    "ticket":"a53fa6"
},
"id" : "XXXXXX"
}

```

6.15.2 INCUMBENT_RESP Parameters

The JSON encoding of the incumbent access response message INCUMBENT_RESP (Section 4.14.2) is described by the following schema:

```

{
  "name" : "INCUMBENT_REQ",

```

```

    "type" : "object",
    "properties" : {
      "type" : "INCUMBENT_REQ",
      "version" : {
        "type" : "string",
        "required" : True
      },
      "incumbentInfos" : {
        "type" : "IncumbentInfo",
        "required" : True
      }
    }
  }
}

```

Example "getIncumbent" JSON-RPC response:

```

{
  "jsonrpc":"2.0",
  "result":{
    "type":"INCUMBENT_REQ",
    "version":"1.0",
    "id":"xxxxx",

```

```

    "exclusiveZone":{
      "region":{
        "exterior":[
          {"latitude":43.25742136,"longitude":-70.0636},
          {"latitude":43.25728263,"longitude":-70.04204977},
          {"latitude":43.25686576,"longitude":-70.02050640},
          .....
          {"latitude":43.25728263,"longitude":-70.08515022}
        ]
      }
    },
    "freqRange":{
      "startHz":3.55E+9,
      "stopHz":3.6E+9
    },
    "scheduleInfo":{
      "timeRange":{
        "startTime":"16:00:00",
        "stopTime":"20:00:00"
      },
      "expireDate": "2015-09-03T10:56:35Z",
      "repitition": True,
      "weekly": {"Monday", "Wednesday"},
      "monthly": {},

```

Ryoo, et al.

Version: Feb 1, 2015

[Page 87]

PTASS-Draft

ptass-protocol

February 1, 2015

```

    },
    "description":"BE CAREFUL"
  },
  "id" : "XXXXXXX"
}

```

7 Security Considerations

PTASS is a protocol whereby a CBSD requests a schedule of available spectrum at its location before it can operate using those frequencies. Unlike loosely controlled protocols such as PAWS [RFC 6953], a PTASS enforces its operation through activation code (Section 4.4.2). When malicious or greedy General Authorized User declines to obey the protocol, the PTASS MAY expire the ticket and able to send deactivation request (Section 4.10.1), an unidentified or greedy CBSD can be pinpointed based on the periodic performance report from the CBSDs in vicinity (Section 4.5), especially from the spectrum sensing report. Note that a greedy user MAY impact the performance of CBSDs in vicinity and violate the hierarchical design of the 3.5GHz spectrum sharing, and this greedy user and outlaw CBSD

make the spectrum sharing defective, especially in the area where the CBSDs are congested.

The PTASS includes sensitive schedules those MUST be kept secured [FCC 14-49]. The connection to the FSAS is only permitted to the CSAS and the message exchange between CSAS and FSAS is rarely occurred. The Incumbent Access's schedules are prohibited to the any type of CBSDs. The CBSD only knows available or not. The reason for the unavailability MAY vary such as congested CBSDs, interference in its vicinity. To distinguish the difference between those mentioned or similar reasons from the IA's activity, the chaser (who wants to know the IA's activity for any reason) MUST collect spectrum sensing data from location to location, the scanning 3.5GHz without permission is subject to be illegal[FCC 14-49].

The transaction among the CBSD, CSAS, and FSAS is depicted in the Figure 17.

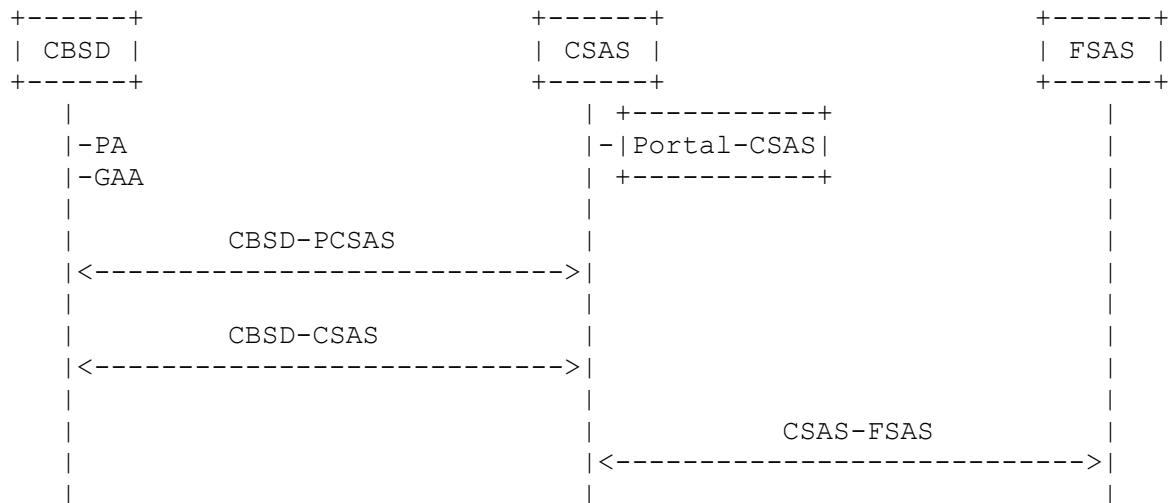


Figure 17

By using the PTASS protocol, the CBSD and the CSAS expose themselves to the privacy risk:

* An unauthorized entity intercepts identifying data for the CBSD or the CSAS, such as serial number, location, ipaddress and port number.

The ticket concept we adopt in the entire scope of the protocol minimize the exposure of the identification and related information, and the use of "HTTP over TLS" [RFC2818] (technically latest SSL for HTTP, briefly HTTPS), assuming the PKI used is not compromised.

The detailed security consideration is dealt in the following sections:

- o Assurance of proper CSAS (Section 7.1)
- o Protection against modification (Section 7.2)
- o Protection against eavesdropping Using (Section 7.3)

7.1 Assurance of Proper CSAS (to the CBSD)

To initiate the service, a CBSD MUST be connected to the corresponding CSAS, a local CSAS of the CBSD's region. The initialization procedure (Section 4.1) covers detail. A local CSAS is hidden from the public access unless it is preconfigured manually. Only the portal CSAS is open to the public and the CBSD MUST be

Ryoo, et al.

Version: Feb 1, 2015

[Page 89]

PTASS-Draft

ptass-protocol

February 1, 2015

communicate to this portal CSAS to obtain (or update) proper CSAS, the assurance of the proper CSAS connection is dealt on the portal CSAS, and it assigns corresponding CSAS based on the geo-location on the request message (Section 4.1.2). In case of unavailability of the CSAS (include error, attack, malfunction), the portal CSAS MAY reassign alternative CSAS.

7.2 Protection Against Modification

To prevent a PTASS 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.

Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", RFC 5491, March 2009.

9.2 Informative References

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

[draft-ietf-paws-protocol-20]
V. Chen, Ed., "Protocol to Access White-Space (PAWS) Databases", November 2014,
<<http://tools.ietf.org/pdf/draft-ietf-paws-protocol-20.pdf>>.

[President's Spectrum Plan Report]
Locke, G., Secretary and Strickling, L. E., "An Assessment of the Near-Term Viability of Accommodating Wireless Broadband Systems in the 1675-1710 MHz, 1755-1780 MHz, 3500-3650 MHz, and 4200-4220 MHz, 4380-4400 MHz Bands", October 2010.

[FCC14-49] FURTHER NOTICE OF PROPOSED RULEMAKING, "Amendment of the Commission's Rules with Regard to Commercial Operations in the 3550-3650 MHz Band", April 2014.

[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://earthinfo.nga.mil/GandG/publications/tr8350.2/tr8350_2.html>.

[ITU-T X.520.2008] International Telecommunication Union, "ITU-T Recommendation X.520: Information technology - Open

Ryoo, et al.

Version: Feb 1, 2015

[Page 91]

PTASS-Draft

ptass-protocol

February 1, 2015

Systems Interconnection - The Directory: Selected attribute types", November 2008,
<<http://www.itu.int/rec/T-REC-X.520-200811-I>>.

Authors' Addresses

Jihoon Ryoo (editor)
Stony Brook University
1500 Stony Brook Road
Stony Brook, NY 11794-6040
US

Email: jiryoo@cs.stonybrook.edu

Chang-Wook Kim
University of Washington
185 Stevens Way
Paul Allen Center
Seattle, WA 98195-2500
US

Email: hyperkcw@u.washington.edu

Milind Madhav Buddhikot
Alcatel-Lucent Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974-0636
US

Email: milind.buddhikot@alcatel-lucent.com