

Application Area Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 15, 2013

S. Das, ed.  
ACS  
J. Malyar  
Telcordia  
D. Joslyn  
SBI  
July 14, 2012

Device to Database Protocol for White Space  
draft-das-paws-protocol-02

Abstract

This document describes the 'Protocol to Access White Space database (PAWS)' that uses HTTP/TLS as transport. The protocol is used for retrieving the necessary TV white space information (e.g., channel, frequency, transmitted power) at a given location and time from a database that is operating under a regulatory domain. The document includes the protocol functionalities, its elements, corresponding data model and recommends its encoding scheme.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 15, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Convention used in this document . . . . .	3
2. Terminology and abbreviations used in this document . . . . .	3
3. Introduction . . . . .	4
4. Protocol Description . . . . .	6
5. Protocol Functionalities and Messages . . . . .	7
5.1. Master WSD Initialization . . . . .	7
5.2. Master WSD Registration . . . . .	8
5.3. Database Query . . . . .	9
5.4. WSD Validation . . . . .	10
6. Data Objects, Elements and Attributes . . . . .	11
6.1. Data Element Definition . . . . .	17
6.2. Attribute Definition . . . . .	19
7. Example Messages with JSON Encoding . . . . .	23
8. The Digest Authentication Scheme . . . . .	29
9. IANA Considerations . . . . .	30
10. Acknowledgements . . . . .	30
11. References . . . . .	30
11.1. Normative References . . . . .	30
11.2. Informative References . . . . .	31
Authors' Addresses . . . . .	31

## 1. Convention 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 [RFC2119].

## 2. Terminology and abbreviations used in this document

Following definitions are copied from  
[I-D.ietf-paws-problem-stmt-usecases-rqmts]

**White Space** Radio spectrum which is not fully occupied at a specific location and time.

**TV White Space**

TV white space refers specifically to radio spectrum which has been allocated for TV broadcast, but is not occupied by a TV broadcast, or other assigned user (such as a wireless microphone), at a specific location and time.

**White Space Device (WSD)**

A device which opportunistically uses some part of white space spectrum. A white space device can be an access point, base station, a portable device or similar. A white space device may be required by local regulations to query a database with its location to obtain information about available spectrum.

**Database**

In the context of white space and cognitive radio technologies, the database is an entity which contains, but is not limited to, current information about available spectrum at any given location and time as required by the regulatory policies.

**Master WSD**

A device which is required to query the WS Database to find out the available operating channels.

**Slave WSD**

A device which uses the spectrum made available by a master device and cannot query the database directly.

Following definitions are copied from FCC 10-174, September, 2010

[FCC]

#### TV Bands Database (TVBD)

A database system that maintains records of all authorized services in the TV frequency bands, is capable of determining the available channels as a specific geographic location.

#### Fixed Device

A TVBD that transmits and/or receives radio communication signals at a specified fixed location.

#### Mode I personal/portable device

A personal/portable TVBD that does not use an internal geolocation capability and access to a TV bands database to obtain a list of available channels.

#### Mode II personal/portable device

A personal/portable TVBD that uses an internal geo-location capability and access to a TV bands database, either through a direct connection to the Internet or through an indirect connection to the Internet by way of fixed TVBD or another 'Mode II' TVBD, to obtain a list of available channels.

### 3. Introduction

Services offered via TV Whitespaces initiative will require a variety of devices and services each acting in accord with rules provided by the regulatory bodies and industries. Along with other things, the service architecture requires the 'Master WSD' to access the 'Database' (e.g. TV Whitespace database) to obtain the necessary information that could be utilized at their location to provide the service. In this document, we focus on this aspect of the overall system: the interface between 'Master WSD' and 'Database'. Figure 1 depicts the device-to-database interface architecture and highlights the scope of this document.

The definition of WSD may differ from one regulatory authority to another. For example, by United States (US) FCC rules, TV WSD is referred to 'Fixed' and 'Personal/Portable device' (e.g., 'Mode II' personal/portable device') [FCC]. The 'Fixed and personal/portable TV WSDs devices' may additionally support other TV WSDs (e.g. 'Mode I' personal/portable device per US FCC rules [FCC]) to establish wireless broadband services. 'Mode I' TV WSDs may also access the

database to obtain the relevant information at their location.

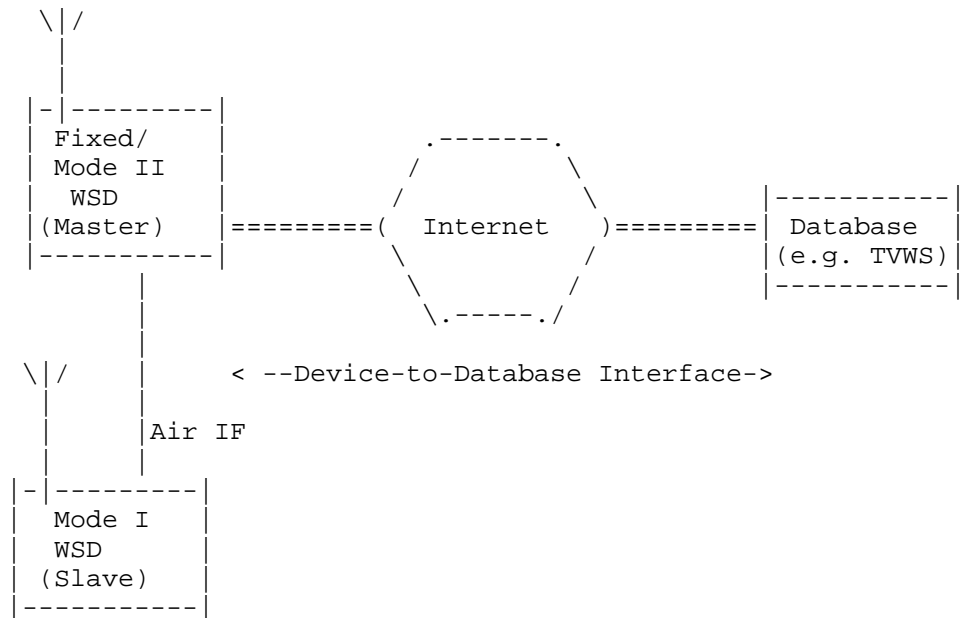


Figure 1: Device-to-Database Interface Architecture

Several use cases and requirements for use of White Space spectrum are described in document

[I-D.ietf-paws-problem-stmt-usecases-rqmts]. This document describes an interface protocol between 'Master device' and 'Database' called PAWS that uses HTTP/TLS as transport and defines the protocol functionalities, messages, its data object models and recommend the encoding scheme. This protocol can be used by the 'Master WSD' to obtain TV Whitespace information (e.g., Channel, frequency, transmitted power and so on) in a given location and time from a white space database that is operating under a regulatory domain.

This document identifies the need to support the requirements by the regulatory authorities of different countries. While some countries have published their requirements (e.g., [FCC], [OfCom]), others are expected to provide in near future. This specification attempt to

define an extensible protocol and its data models that should accommodate the need for various regulatory authorities.

#### 4. Protocol Description

Figure 2 shows the interface protocol stack. The interface protocol (PAWS) is an application protocol that uses HTTP/TLS as transport.

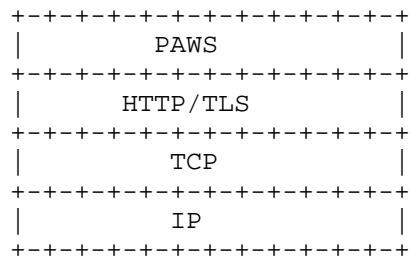


Figure 2: Example Protocol Layer

The 'Master WSD' uses the HTTPS-enabled PAWS to perform the following operations. The protocol functionalities and message flows are described in Section 5.

- o The device MUST first locate or discover the URI for the database to send the PAWS messages. The URI for the database SHOULD be obtained from an authorized and authenticated entity. The discovery of database URI details are outside the scope of this document. However, it is RECOMMENDED that the type of URI provided by database discovery method SHOULD be an HTTPS URI.
- o Once the TLS handshake between device and database server has finished, the 'Master WSD' initiates the initial service request to WS database server in the body of an HTTPS request via the POST method as described in [RFC2818] to exchange certain information including the capability exchange. The database responds with the message in the body of the HTTPS response. The database uses a server certificate issued by a well-known certificate authority. The devices can authenticate the server side certificate by configuring the trust to the issuing authority. The device authentication is performed by the database server at the PAWS layer by using 'Digest Authentication'.

- o Once authenticated, the device registers with the WS database and establishes certain operational parameters as required by the spectrum management authority. The registration messages are handled via same HTTP method as described above. The WS database returns the result of the registration.
- o After device and server are mutually authenticated and the device is authorized to obtain the service, the device sends a query message to the WS database with the required parameters for obtaining WS channel information. The channel query messages are handled via same HTTP method as described above. The WS database returns the available channel information.
- o Depending upon the regulatory and operational requirements, 'Master WSD' may need to verify the identity of the 'Slave WSDs'. 'Master WSD' sends a validation message to the WS database with the required information. The device validation messages are handled via same HTTP method as described above. The WS database returns the result of the validation.
- o The data model and its relationship with data-objects, data-elements and attributes are described in Section 6. The example messages are encoded in JSON structure and is described in Section 7. Other encodings (such as XML) may be added in the future version of the document. However, all encoding should follow the same data model.

## 5. Protocol Functionalities and Messages

Following are the protocol functionalities and message flows of the WS Application protocol.

### 5.1. Master WSD Initialization

The 'Master WSD' initialization is the process of a device establishing initial connection to the database. Each time the 'Master WSD' powers up or opens up a communication with the database, it exchanges these messages. The principle purpose of the initialization messages are: i) exchange the capability information related to regulatory domains and operations; ii) the device to request parameters that will initiate the client authentication process. In particular, this will allow the device and server to know in which regulatory domain the service is requested for, the sequence of protocol operations necessary to fulfill the regulatory requirements. In addition, PAWS client in this step will obtain the authentication parameters from the server that will enable client device to proof its authenticity and provide message integrity during

the entire protocol operation at the PAWS application layer. The confidentiality is achieved at the HTTPS layer. However, the information needed for these initial message exchanges at the PAWS layer may vary depending upon the deployment and regulatory requirements. In this document we describe the use digest of authentication scheme for device authentication similar to the use in [RFC3261], the details of which are described in Section 8.

'INIT-REQ' and 'INT-RESP' messages are used to perform the Master WSD initialization with the database. Figure 3 depicts the call flow of the initialization message.

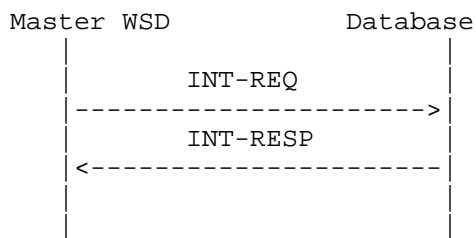


Figure 3: WSD Initialization Message Flow

## 5.2. Master WSD Registration

The 'Master WSD' registration is the process of a device establishing certain operational parameters with the database, as required by the spectrum management authority. FCC rules, for example, requires that 'Fixed Devices' register as owner and/or operator contact information. 'Fixed Devices' may also register their fixed location and antenna height parameters with the database. Registration is required upon its initial contact with the database, or when its registered parameters change (e.g., a Fixed device is redeployed at a new location, or its antenna height is adjusted,) or registration life time expires. However, this functionality may be optional in certain regulatory domains.

'REG-REQ' and 'REG-RESP' messages are used to perform the 'Master Device' registration with the database. Figure 4 depicts the call flow of the registration message.



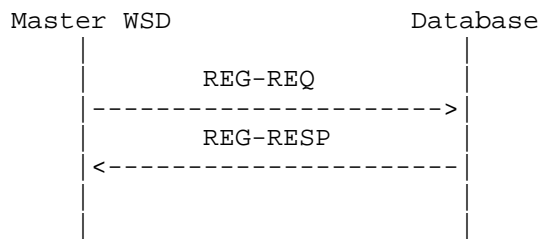


Figure 4: WSD Registration Message Flow

### 5.3. Database Query

In order to obtain the available channel and other information 'Master WSD' needs to query the 'Database'. In certain regulatory environment, it may be required to be authenticated and registered before a 'Master WSD' can query the database and in other domains, the requirements may vary. 'Master WSD' will perform 'Initialization' and 'Registration' procedures as and when required before querying the database.

When the 'Master WSD' sends a query to the 'Database', it sends its location (e.g., Geo-location) along with other parameters. 'Database' returns an array of channels within the scope of the request (e.g., VHF/UHF) and regulatory authority where the returned elements contain the channel frequency range, availability indicator, operating power, event management (indicator when channel is scheduled or is not available), and so on. It may also include other parameters depending upon the regulatory requirements.

'AVAIL-CHAN-REQ' and 'AVAIL-CHAN-RESP' messages are used by devices for querying the database in a given location. 'USE-CHAN-NOTIFY' message is used by the 'Master WSD' to notify the database which channels are anticipated to be used in that location by the master WSD or associated slave devices. 'USE-CHAN-RESP' message is used by the database to positively or negatively acknowledge (ACK/NACK) the channel availability. Figure 5 depicts the call flow of the database query message.

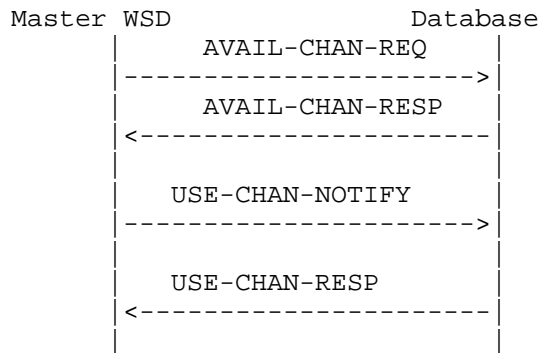


Figure 5: Query Message Flow

#### 5.4. WSD Validation

The WSD validation is the process by which slave WSDs can be validated by the database. For example, by US FCC rule, the 'TVWS Fixed or Mode II' device provides service to a Mode I device only after the Mode I is validated by the TVWS database. To facilitate this validation, 'Database' needs to support 'WSD Validation' capability. However, the requirement may vary from one regulatory domain to another.

'DEV-VALID-REQ' and 'DEV-VALID-RESP' messages are used by 'Master WSD' for 'Slave WSD' validation with the database. Figure 6 depicts the call flow of validation message.

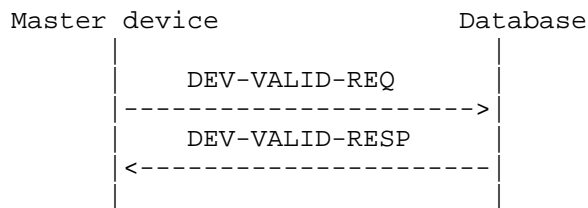
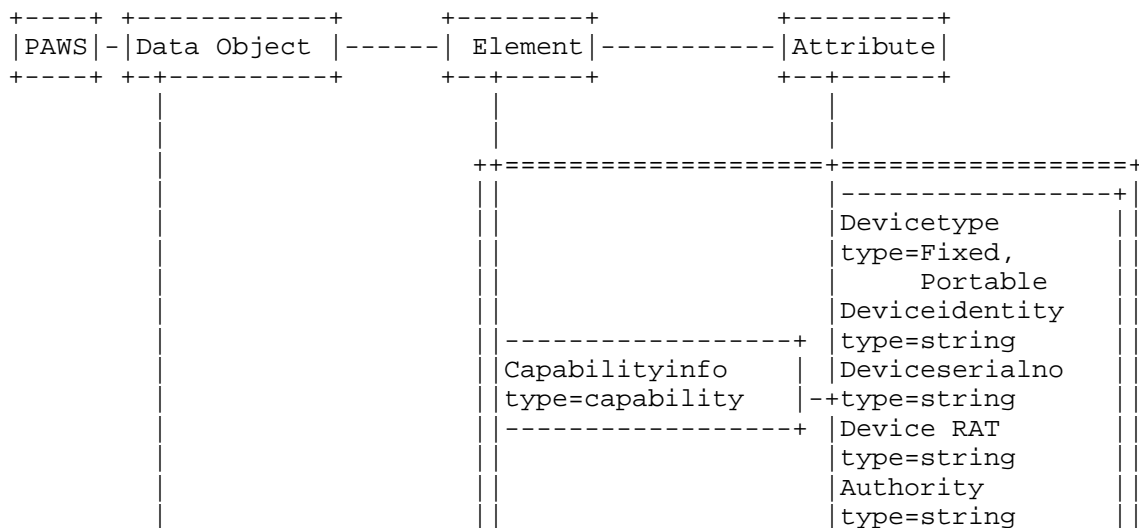


Figure 6: WSD Validation Message Flow

## 6. Data Objects, Elements and Attributes

This section presents the data objects, data elements and attribute list and show their relationships. Protocol messages are mapped to data-objects and then elements are derived based on necessary information that need to be exchanged. Parameters are listed as attributes per data elements. The model is structured in such a way that additional data-objects, elements and attributes can be added easily as new requirements emerge. In addition, there could be a need for vendor-specific attributes as things evolve.

Each data object (e.g., REQ, RESP, NOTIFY) will contain a set of data elements which will carry a set of attributes. In general, 'REQ' and 'NOTIFY' data objects are included in the body of the HTTPS Request message and is initiated by the 'Master WSD' while 'RESP' data object is included in the body of the HTTPS Response message and is generated by the 'WS Database'. The absence or presence of the data elements and attributes within a data object is mostly dictated by the regulatory domain where the device and the database are operating. However, some data elements are independent of the regulatory domain and are essential for the protocol operation. For example, the 'Capabilityinfo', 'ProtocolInfo', 'Devicelocation', 'AvailChannellist' and 'Authinformation' in this version are mandatory while others are optional and their use will depend upon the regulatory domain rules where the service is being offered.





			Resultcode type=boolean
			Errorcode type=number
			-----+
			-----+
			Latitude type=float
			longitude type=float
			longitude type=float
			Locuncertainty type=number
+-----+	Registration REQ/RESP	Deviceolocation type=geoloc,civic	Locconfidence type=number
			HAGL type=float
			HAGLuncertainty type=number
			Antennatype type=int
			Geolocationcode type=string
			-----+
			-----+
			Ownername type=string
			Address type=string
			phonenumber type=alphanumeric
			Email type=alphanumeric
			Operatorname type=string
			address type=string
			phonenumber type=alphanumeric
			Email type=alphanumeric
			-----+
			-----+
			Authscheme type=string
			-----+

		-----+	Realm type=string Nonce type=string cnonce type=string qop type=string -----+	
		+++++		
		+++++		
			-----+	Devicetype type=Fixed, Portable Deviceidentity type=string Deviceidentity type=string DeviceRAT type=string Authority type=string -----+
		-----+	Capabilityinfo type=capability -----+	
		-----+	ProtocolInfo type=proto -----+	-----+ Protoversion type=float -----+ Messagetype type=string Resultcode type=boolean Errorcode type=string -----+
				-----+ Latitude type=float longitude type=float longitude type=float longitude type=float -----+
		-----+	Devicelocation type=geoloc,civic -----+	Locuncertainty type=number -----+ Locconfidence type=number -----+
	+-----+			
-	Databasequery REQ/RESP; NOTIFY/RESP	-+		



[illegible]



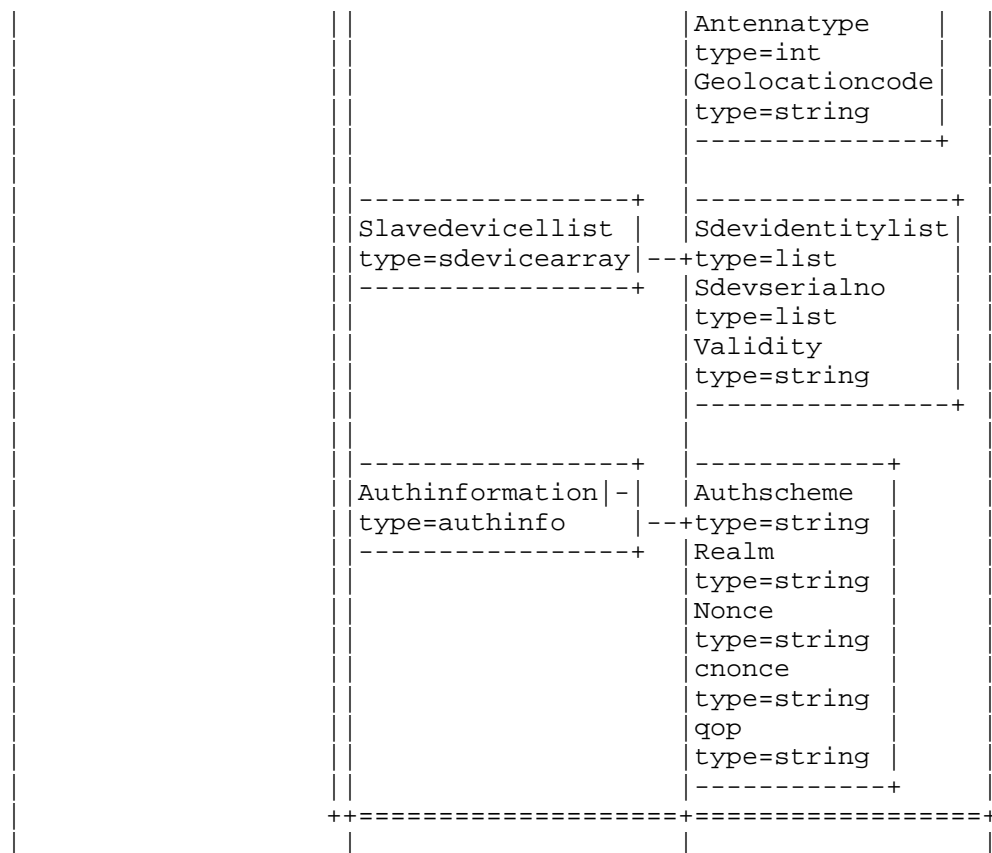


Figure 7: PAWS Data Model

### 6.1. Data Element Definition

Capabilityinfo;type=capability

This data element is used to exchange the capability information of the 'Master Device' including the regulatory domain information in which the service is requested for. This is a mandatory feature of the protocol operation. However, the list of attributes may depend upon the regulatory domain requirements.

`Protocolinfo;type=proto`

This data element is used to exchange the protocol information such, as protocol version, message type and the result code. This is a mandatory feature of the protocol operation.

`Authinformation;type=authinfo`

This data element is used to exchange authentication information that is required by the server for the client authentication and provide authentication and message integrity of the subsequent messages. This include 'Digest Authentication' parameters and it is a mandatory feature of the protocol operation.

`Devicelocation;type=geo-loc, civic`

This data element is used to provide the location information of the 'Master device' to the database. This is a mandatory feature of the protocol operation and the list of attributes for this element may depend upon the regulatory domain requirements.

`Deviceowner;type=ownerinfo`

This data element is used to provide the owner information (as applicable) of the 'Master WSD' to the database operator. This is an optional feature of the protocol operation and the list of attributes for this element may depend upon the regulatory domain requirements.

`Deviceowner;type=operatorinfo`

This data element is used to provide the operator information (as applicable) of the 'Master WSD' to the database operator. This is an optional feature of the protocol operation and the list of attributes for this element may depend upon the regulatory domain requirements.

`AvailChannellist;type= channelarray`

This data element is used by the 'Master WSD' to query the available channels in a given location and time and by the 'WS database' for corresponding response. This is a mandatory feature of the protocol operation and the list of attributes for this element may depend upon the regulatory domain requirements.

UsedChannellist;type= channelarray

This data element is used by the 'Master WSD' to notify the channels used to the 'WS database'. This is an optional feature of the protocol operation and the list of attributes for this element may depend upon the regulatory domain requirements.

Slavedevicellist;type= sdevicearray

This data element is used by the 'Master WSD' to validate the 'Slave WSDs' with the 'WS Database'. This is an optional feature of the protocol operation and the list of attributes for this element may depend upon the regulatory domain requirements.

## 6.2. Attribute Definition

Devicetype;type=string

The type of the whitespace device is being used (fixed or portable). Additionally, this could include regulatory type name for example, in US FCC, device type can be called as Mode II portable/personal device as mentioned in Section 2.

Deviceidentity;type=string

The identity of the whitespace devices (Master WSD and Slave WSD). This could be a regulatory domain id (e.g., FCCid in US).

Deviceserialno;type=string

The manufacturer serial number of WSD (e.g., Master WSD and Slave WSD).

DeviceRAT;type=string

This attribute represents the Radio Access Technology (RAT) for the master device.

Authority;type=string

The regulatory domain where the WS service is needed

Protoversion;type=float

This attribute represents the Version number of the protocol

Msgtype;type=string

Type of the WS application protocol message. An enumerated type. Allowed messages are INIT-REQ, INIT-RESP, REG-REQ, REG-RESP, AVAIL-CHAN-REQ, AVAIL-CHAN-RESP, USE-CHAN-NOTIFY, USE-CHAN-RESP, DEV-VALID-REQ and DEV-VALID-RESP

Resultcode; type=boolean

This attribute indicates the result of the message transaction.

Errorcode; type=number

If Resultcode is 'failure', the value of 'Errorcode' represents a specific reason for failure. Error codes are TBD.

Authscheme;type=string

The name of the authentication scheme used to authenticate the device at the PAWs layer.

Realm;type=string

A human readable string identifying the security realm of the authentication scheme.

Nonce;type=string

A random string data that the server (database) sends to the device.

Cnonce;type=string

A random string data that the client (device) sends to the server.

qop;type=string

A string of one or more tokens indicating the 'quality of protection' values supported by the server. This is optional per [RFC2617].

Latitude;type=float

This attribute represents the location co-ordinate; [RFC3825] based representation including resolution can be used. While disclosing such information, privacy and user preferences are important, [RFC4119] based representation should be used.

Longitude;type=float

This attribute represents the location co-ordinate. [RFC3825] based representation including resolution can be used. Where disclosing such information, privacy and user preferences are important, [RFC4119] based representation should be used.

Locuncertainty;type=number

This attribute represents the location uncertainty in meters. The value is assumed to be 0 when not provided.

Locconfidence;type=number

This attribute represents the location confidence in percentage. The value is assumed to be 100 when not provided.

HAGL;type=float

The antenna height above the ground level or average terrain.

HAGLuncertainty;type=float

This attribute represents the HAGL Uncertainty in meters. The value is assumed to be 0 when not provided.

Antennatype;type=int

The identity of antenna used for transmission. The identity of the antenna can be regulatory domain specific.

Geolocationcode;type=string

A value indicating whether the device location components (latitude, Longitude) have been calculated according to another set of parameters, for example civic address. For example, if geo-coding or reverse geo-coding algorithms are used.

Ownername;type=string

This attribute represents the name of the device owner.

Operatorname;type=string

This attribute represents the name of the device operator.

Address;type=string

The civic address of the device owner or operator of the device.

Email;type=alphanumeric

The email address of the device owner or operator of the device.

Phonenumber;type=alphanumeric

The phone number of the device owner or the operator of the device.

Requesttype;type=allchannels, availableonly

This attribute provides the ability for the Master WSD to request either available and unavailable channels or available channels only. When requested as 'allchannels' all available and unavailable channels will be returned. When 'availableonly' is requested, only available channel will be returned.

Channelno;type=string

The list of channels that are available or selected in a given location and time.

Minfreq;type=string

The minimum frequency of the indicated channel represented in MHz.

Maxfreq;type=string

The maximum frequency of the indicated channel represented in MHz.

MaxEIRP;type=float

Maximum effective radiated power measured in dBW.

Datetime;type=string

This attribute represents the time that the channel is available until when availability flag is 'True'. When the availability flag is 'False' it indicates when the channel may become available. However, the Master WSD needs to query again after the time expires. The format of this representation is a datetime stamp.

`Duration;type=sequence`

This attribute represents the time that the channel is available until when availability flag is 'True'. When the availability flag is 'False' it indicates when the channel may become available. However, the Master WSD needs to query again after the time expires. The format of this representation is a sequence of ticks and its units. The value of the Unit can be seconds, minutes or hours.

`Availability;type=string`

This attribute indicates the availability of the channel (true or false).

`Usedchannelno;type=list`

The list of channels that are used by the Master WSD or its associated slave devices in a given location and time.

`Sdeviceidentitylist;type=list`

The list of the slave devices' identity that needs the validation with the database.

`Sdevserialno;type=list`

The list of the slave devices' manufacturer serial numbers that needs the validation with the database.

`vailidity;type=list`

This attribute indicates the validity of the slave devices (true or false).

## 7. Example Messages with JSON Encoding

The examples in this section show the HTTPS messages that include few PAWs messages. The message is encoded in JSON [JSON] structure.

The following example shows the request for a registration. the POST includes the 'REG-REQ' object.

```
POST/REG_REQ HTTPS/1.1
Host:WSP.example.com:443
Content-Type:application/PAWS+json; charset=utf-8
content length: XX
{
  "Protoversion": "1.0",
  "messagetype": "REG_REQ",
  "Authority": "US",
  "Devicetype": "F",
  "Deviceidentity": "TTT1234",
  "Deviceserialno": "01AB23CD45EF",
  "Latititude": "40.5414",
  "Longitude": "-74.4750",
  "Locuncertainty": "2",
  "LocConfidence": "95",
  "HAGL": "25",
  "HAGLuncertainty": "1",
  "Antennatype": "MM",
  "Geolocationcode": "DEFAULT",
  "Ownername": "XYZ",
  "Address": "444 Hoes Lane, US, 08854",
  "Phonenumber": "17326992000",
  "Email": "XYZ@gmail.com",
  "Operatorname": "XYZ",
  "Address": "444 Hoes Lane, US, 08854",
  "Phonenumber": "17326992000",
  "Email": "XYZ@gmail.com",
  "Authscheme": "DIGEST",
  "Realm": "PAWS-DDI",
  "Nonce": "7b52009b64fd0a2a49e6d8a939753077792b0554",
  "Cnonce": "bd307a3ec329e10a2cff8fb87480823da114f8f4",
  "qop": "auth",
  "resp": "4dfb972d427b4100c821d53b8bea9b2c33b74a7e",
}
```

The following example shows the response for a registration. the POST includes the 'REG-RESP' object.



```
POST/REG_RESP HTTPS/1.1 200 OK
Server: Example PAWS
Date: Fri, 12 June 2012 06:24:27 GMT
Expires: Fri, 30, June 2012, 23:59:00, GMT
Cache-control : private
Content-Type:application/PAWS+json; charset=utf-8
content length: YY
{
  "Protoversion": "1.0",
  "Messagetype": "REG_RESP",
  "Authority": "US",
  "Resultcode": "success",
  "Authscheme": "DIGEST",
  "Realm": "PAWS-DDI
  "Nonce": "7b52009b64fd0a2a49e6d8a939753077792b0554"}
  "qop": "auth"
}
```

The following example shows the available channel request. The POST includes the 'AVAIL-CHAN-REQ' object.

```
POST/AVAIL-CHAN-REQ HTTPS/1.1
Host:WSP.example.com:443
Content-Type:application/PAWS+json; charset=utf-8
content length: XX
{
  "Protoversion": "1.0",
  "messagetype": "AVAIL_CHAN_REQ",
  "Authority": "US",
  "Devicetype": "F",
  "Deviceidentity": "TTT1234",
  "Deviceserialno": "01AB23CD45EF",
  "Latititude": "40.5414",
  "Longitude": "-74.4750",
  "Locuncertainty": "2",
  "LocConfidence": "95",
  "HAGL": " 25",
  "HAGLuncertainty": "1",
  "Antennatype": "MM",
  "Geolocationcode": "DEFAULT",
  "Requesttype": "allchannels",
  "Authscheme": "DIGEST",
  "Realm": "PAWS-DDI",
  "Nonce": "7b52009b64fd0a2a49e6d8a939753077792b0554",
  "Cnonce": "bd307a3ec329e10a2cff8fb87480823da114f8f4",
  "qop": "auth",
  "resp": "4dfb972d427b4100c821d53b8bea9b2c33b74a7e",
}
```

The following example shows the response for available channel request. the POST includes the 'AVAIL-CHAN-RESP' object.

```
POST/AVAIL_CHAN_RESP HTTPS/1.1 200 OK
Server: Example PAWS
Date: Fri, 12 June 2012 06:24:27 GMT
Expires: Fri, 12 June, June 2012, 20:30:00, GMT
Cache-control : private
Content-Type:application/PAWS+json; charset=utf-8
content length: YY
{
  "Protoversion": "1.0",
  "Messagetype": "AVAIL_CHAN_RESP",
  "Authority": "US",
  "Resultcode": "success",
```

```
"Authscheme": "DIGEST",
"Realm": "PAWS-DDI
"Nonce": "7b52009b64fd0a2a49e6d8a939753077792b0554"}
  "qop": "auth",
  "Channellist": [
    {
      "Channelno": 2,
      "Minfreq": 54,
      "Maxfreq": 60
      "MaxEIRP": 4000,
      "Datetime": "20120612T235959Z",
      "Duration": "1440, mins",
      "Availability": true
    },
    {
      "Channelno": 3,
      "Minfreq": 60,
      "Maxfreq": 66,
      "MaxEIRP": 0,
      "Datetime": "20120612T235959Z",
      "Duration": "1440, mins",
      "Availability": false
    },
    .
    .
    .
    {
      "Channelno": 51,
      "Minfreq": 692,
      "Maxfreq": 698,
      "MaxEIRP": 4000,
      "Datetime": "20120612T120000Z",
      "Duration": "720, mins ",
      "Availability": true
    }
  ]
}
```

The following example shows the request for a device validation. the POST includes the 'DEV\_VALID-REQ' object.

```
POST/DEV_VALID_REQ HTTPS/1.1
Host:WSP.example.com:443
Content-Type:application/PAWS+json; charset=utf-8
content length: XX
{
  "Protoversion": "1.0",
  "messagetype": "DEV-VALID-REQ",
  "Authority": "US",
  "Devicetype": "F",
  "Deviceidentity": "TTT1234",
  "Deviceserialno": "01AB23CD45EF",
  "Latititude": "40.5414",
  "Longitude": "-74.4750",
  "Locuncertainty": "2",
  "LocConfidence": "95",
  "HAGL": " 25",
  "HAGLuncertainty": "1",
  "Antennatype": "MM",
  "Geolocationcode": "DEFAULT",
  "ownername": "XYZ",
  "Address": "444 Hoes Lane, US, 08854",
  "Phonenumber": "17326992000",
  "Email": "XYZ@gmail.com",
  "Operatorname": "XYZ",
  "Address": "444 Hoes Lane, US, 08854",
  "Phonenumber": "17326992000",
  "Email": "XYZ@gmail.com",
  "Authscheme": "DIGEST",
  "Realm": "PAWS-DDI",
  "Nonce": "7b52009b64fd0a2a49e6d8a939753077792b0554",
  "Cnonce": "bd307a3ec329e10a2cff8fb87480823da114f8f4",
  "qop": "auth",
  "resp": "4dfb972d427b4100c821d53b8bea9b2c33b74a7e",
  "Sdevicelist": [
    "sdevidentity": "1234",
    "sdevserialno": "4321",

    "sdevidentity": "5678",
    "sdevserialno": "8765",

    "sdevidentity": "91011",
    "sdevserialno": "11109",
  ]
}
```

The following example shows the response for a device validation. The POST includes the 'DEV-VALID-RESP' object.

```
POST/DEV_VALID_RESP HTTPS/1.1 200 OK
Server: Example PAWS
Date: Fri, 12 June 2012 06:24:27 GMT
Expires: Fri, 30, June 2012, 23:59:00, GMT
Cache-control : private
Content-Type:application/PAWS+json; charset=utf-8
content length: YY
{
  "Protoversion": "1.0",
  "Messagetype": "DEV_VALID_RESP",
  "Authority": "US",
  "Resultcode": "success",
  "Authscheme": "DIGEST",
  "Realm": "PAWS-DDI",
  "Nonce": "7b52009b64fd0a2a49e6d8a939753077792b0554"}
  "qop": "auth",
  "Sdevicelist": [
    {
      "sdevidentity": "1234",
      "sdevserialno": "4321",
      "Validity": true
    },
    {
      "sdevidentity": "5678",
      "sdevserialno": "8765",
      "Validity": true
    },
    {
      "sdevidentity": "91011",
      "sdevserialno": "11109",
      "Validity": false
    }
  ]
}
```

## 8. The Digest Authentication Scheme

This section describes the modifications and clarifications required to apply the HTTP Digest authentication scheme to PAWS. The PAWS scheme usage is almost completely identical to that for HTTP [RFC2617] and in particular SIP [RFC3261] except MD5 is replaced by SHA-1 and with the following differences. Future version may support

SHA-2 (.e.g., SHA-256, SHA-384 and SHA-512).

- o The URI and method included in the challenge are empty. Therefore to the calculation of the A2 value for message integrity assurance in the Digest authentication scheme, the hash of the entity-body resolves to the SHA-1 hash of an empty string;  $H(\text{entity-body}) = \text{SHA1}(" ") = 654e0aaee80e38636c503629d32225db31a616de$
- o The realm represents one 'security realm' and the value can be chosen arbitrarily but the realm field from the challenge must be used in the calculation
- o The device's serial number should be mapped to 'username' and the device's shared secret is mapped to 'password'. The shared secret MUST have a binding with the device serial number. This document does not describe on how to establish a shared secret.

## 9. IANA Considerations

TBD

## 10. Acknowledgements

Authors would like to acknowledge Dan Harasty, Anthony Triolo, Joel Halpern and Peter Stanforth for their constructive input and feedback on this document.

## 11. References

### 11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [I-D.ietf-paws-problem-stmt-usecases-rqmts] Probasco, S. and B. Patil, "Protocol to Access White Space database: PS, use cases and rqmts", draft-ietf-paws-problem-stmt-usecases-rqmts-06 (work in progress), July 2012.
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261,

June 2002.

- [RFC3825] Polk, J., Schnizlein, J., and M. Linsner, "Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information", RFC 3825, July 2004.
- [RFC4119] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, December 2005.
- [RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

#### 11.2. Informative References

- [FCC] FCC, "Second Memorandum Opinion and Order", FCC 10-174, 2010.
- [OfCom] OFCom, "Implementing Geolocation: <http://stakeholders.ofcom.org.uk/consultations/geolocation/statement/>", September, 2011.
- [JSON] JSON, "<http://www.json.org/>".

#### Authors' Addresses

Subir Das, ed.  
Applied Communication Sciences  
444 Hoes Lane  
Piscataway, NJ 08854  
U.S.A.

Email: sdas at appcomsci dot com

John Malyar  
Telcordia Technologies Inc.  
1 Ericsson Drive  
Piscataway, NJ 08854  
U.S.A.

Email: jmalyar at telcordia dot com

Donald Joslyn  
Spectrum Bridge Inc.  
1064 Greenwood Blvd.  
Lake Mary, FL 32746  
U.S.A.

Email: d.joslyn at spectrumbridge dot com





Working Group Draft  
Internet-Draft  
Intended status: Informational  
Expires: January 17, 2013

S. Probasco, Ed.  
B. Patil  
Nokia  
July 16, 2012

Protocol to Access White Space database: Discovery  
draft-probasco-paws-discovery-01

Abstract

A white space master device needs to query a white space database and obtain information about available spectrum/channels prior to operation. White space databases which contain information about available spectrum/channels are associated with a regulatory domain and hence specific to a country or region. A white space master device needs to discover the relevant white space database(s) given its current location and the regulatory domain that it is operating in. The white space database discovery is the preliminary step that a white space master device has to perform.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 17, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	4
3. Overview . . . . .	4
4. Specification . . . . .	6
4.1. Protocol Description . . . . .	6
4.2. Protocol Messages . . . . .	7
4.3. Data Model . . . . .	7
5. IANA Considerations . . . . .	7
6. Security Considerations . . . . .	7
7. Summary and Conclusion . . . . .	8
8. Acknowledgements . . . . .	8
9. References . . . . .	8
9.1. Normative References . . . . .	8
9.2. Informative References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

White space database discovery is preliminary to creating a radio network using white space. The radio network is created by a master device that must contact a trusted database to learn if any radio frequencies or channels are available for use before the master device transmits in white space spectrum. Discovery is a necessary service for PAWS protocol, see PAWS: problem statement, use cases and requirements [PAWS RQMTS].

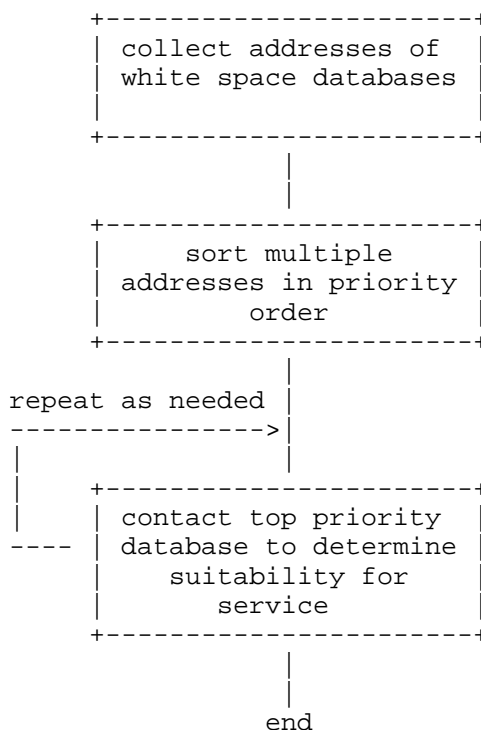


Figure 1: High level view of white space database discovery

Figure 1 shows at a high level how white space master devices discover a suitable trusted white space database. In this document we describe how the master device may collect the addresses of one or more white space database. Steps and criteria to sort multiple addresses into a priority order is left to implementation and not specified. Procedures to contact a white space database are specified in (ED NOTE: insert reference to PAWS standard, when available). Steps and criteria to determine the suitability of a

particular white space database are left to implementation.

## 2. Terminology

The terminology from PAWS: problem statement, use cases and requirements [PAWS RQMTS] is applicable to this document.

### White Space Database (WSDB)

In the context of white space and cognitive radio technologies, the database is an entity which contains, but is not limited to, current information as required by the regulatory policies about available spectrum at any given location and time, and other types of related (to the white space spectrum) or relevant information.

### White Space Database Discovery Server (WSDB DS)

A server function provided to a white space device, the client. The white space device contacts a white space database discovery server to receive the service of discovering or identifying one or more white space databases. The white space database discovery server is a known entity to the white space device, which knows at least a useable internet address for the white space database discovery server. The white space database discovery server takes as input positioning information from the white space device and returns both address information which allows the white space device to contact a trusted, regulatory-authorized white space database, suitable for service at the white space device's current location and indication of the regulatory domain governing at the white space device's current location. A single white space database discovery server may have global scope, serving clients located globally.

## 3. Overview

Before the WSD can query a trusted WSDB for a list of available frequencies or channels for use in the white space spectrum, the WSD must first discover the available databases and addresses serving the regulatory domain in which the device is currently located. At power-up the WSD does not reliably know the regulatory domain corresponding to its current location, and therefore does not reliably know with which white space database(s) it can communicate. Furthermore it is essential that the WSD connect with a trusted WSDB for proper operation and indeed regulatory compliance. By including contact information of a trusted WSDB DS in the WSD's programmed

instructions or firmware, the WSD can reliably determine the address of a trusted database or database listing server, as appropriate for its current physical location.

While it possible that a WSD knows its location, or information which may be used to derive its location, it is not reasonable for every WSD to be capable to translate this information into the current regulatory domain, i.e. the WSD needs assistance to know what is the regulatory environment with jurisdiction at its current location. A WSDB Discovery Server (DS) takes as input location information from the WSD and returns to the WSD one or more addresses of WSDBs (or WSDB listing servers as appropriate) to the WSD. If the address or addresses of these WSDB DSs are included in the WSD firmware, a secure starting point for a trusted relationship is established. Because the WSDB DS is selected by the WSD manufacturer, a foundation is set to ensure the WSD will be able to discover a trusted WSDB in every regulatory domain where the manufacturer expects the WSD to be used.

When the WSD does not have the address of a serviceable WSDB (e.g. at power-up), the WSD sends a Discovery Request message to a WSDB DS. The address of at least one WSDB DS is included in the WSD operating instructions or firmware by the manufacturer for example or provisioned using device configuration mechanisms. The WSD includes in the Discovery Request information about its current location. The WSDB DS uses this location information to determine the regulatory domain where the WSD is located, and returns a Discovery Response message which includes the address of one or more WSDBs (or WSDB listing server as appropriate) to the WSD. See Figure 2.

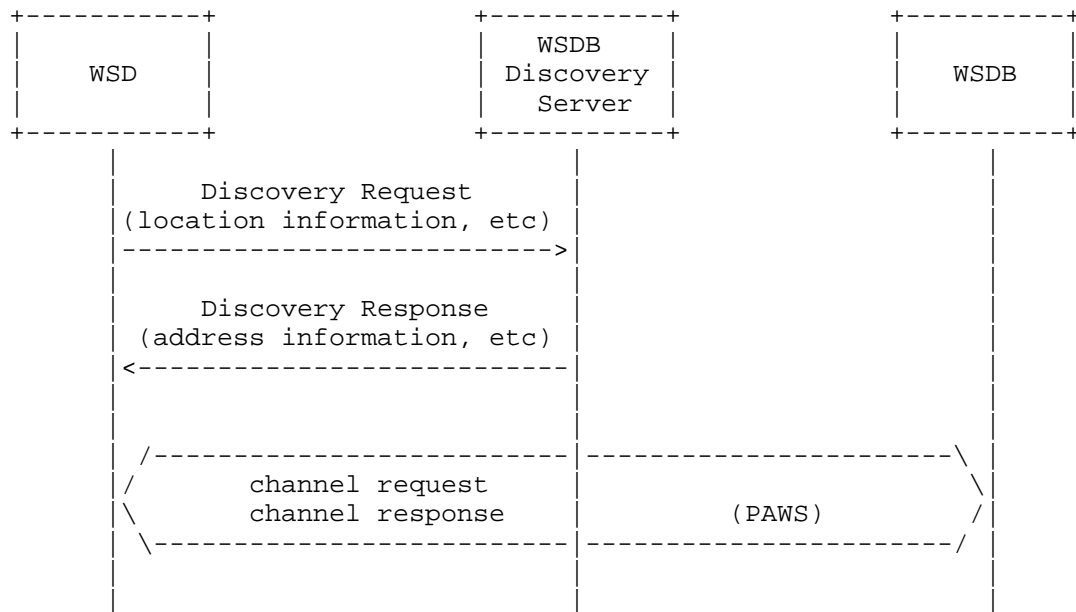


Figure 2: Example illustration of registration of the discovery process using PAWS: Discovery

The discovery procedure fulfills requirements P.1, P.2 and P.3 from PAWS: problem statement, use cases and requirements [PAWS RQMTS].

#### 4. Specification

##### 4.1. Protocol Description

PAWS: Discovery is an application protocol that uses HTTP/TLS as transport. See Figure 3.

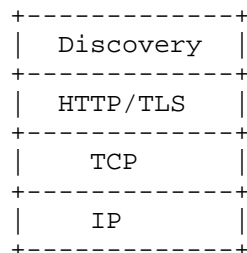


Figure 3: Protocol stack

#### 4.2. Protocol Messages

The WSD sends Discovery-REQ to the WSDB DS and return receive Discovery-RSP, see Figure 4.

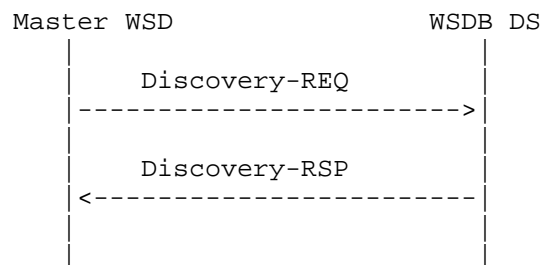


Figure 4: Discovery message flow

1. The Discovery-REQ message contains information to allow the WSDB DS to identify and determine the location of the Master WSD.
2. The Discovery-RSP message contains the regulatory domain and either the address of a listing server or the address of one or more WSDB authorized to provide service where the WSD is physically located. If spectrum access is not authorized at the WSD physical location, the response will contain an error code and no address information.

#### 4.3. Data Model

#### 5. IANA Considerations

This document has no requests to IANA.

#### 6. Security Considerations

The white space database provides a critical service to white space master devices in the form of query responses about available spectrum/channels for use at a specific location and time. The white space database is specific to a regulatory domain. A white space master device querying a database needs to ensure that it is communicating with a valid and authorized entity. The master device performs database discovery prior to establishing a session with a white space database for querying spectrum/channel availability. The database discovery process needs to be secured in order to ensure that the master device is provided with the address of a valid and authorized database for the specific regulatory domain. There is a



trust relationship that needs to be established between the master device and the entity which aids it in database discovery.

## 7. Summary and Conclusion

White space database discovery is a preliminary step in the process of creating a radio network using white space by devices. A simple and secure means to discover valid and authorized database(s) within the scope of a regulatory domain by a WSD is specified in this document. A trust relationship between the WSD and the WSDB discovery server ensures security w.r.t the list of databases provided to the WSD.

## 8. Acknowledgements

The authors would like to acknowledge Brian Rosen, Peter Stanforth and Andy Sago for their comments which have helped improve this document.

## 9. References

### 9.1. Normative References

[PAWS RQMTS]

IETF, "Protocol to Access White Space database: PS, use cases and rqmts;", December 2012.

### 9.2. Informative References

## Authors' Addresses

Scott Probasco (editor)  
Nokia  
6021 Connection drive  
Irving, TX 75039  
USA

Email: [scott.probasco@nokia.com](mailto:scott.probasco@nokia.com)

Basavaraj Patil  
Nokia  
6021 Connection drive  
Irving, TX 75039  
USA

Email: basavaraj.patil@nokia.com



Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: January 10, 2013

X. Wei  
L. Zhu  
P. McCann  
Huawei  
July 9, 2012

PAWS Framework  
draft-wei-paws-framework-00

## Abstract

Portions of the radio spectrum that are allocated to a licensed, primary use but are unused or unoccupied at specific locations and times are defined as "white space". White space devices can make use of this spectrum; however, they must first determine which spectrum is unused or unoccupied by a primary user at their current location. A white space database can be consulted that holds information about primary users of the spectrum and that returns information about white space. In this document we introduce a Protocol for Access to WhiteSpace database (PAWS) which is for use between a white space device and a white space database. We give a framework for PAWS, a protocol stack that defines the interface between the white space device and the white space database, the parameters of the protocol, an XML schema that can encode the parameters, and example messages. The realization of the database and the calculation of protected contour are not considered in this framework draft.

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 10, 2013.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology and Abbreviation . . . . .	5
2.1. Conventions Used in This Document . . . . .	5
2.2. Terminology . . . . .	5
3. Overview of PAWS . . . . .	6
4. Protocol Stack . . . . .	9
5. Protocol Framework and Interface of PAWS . . . . .	10
5.1. Database Discovery . . . . .	11
5.2. Device Registration with Trusted Database . . . . .	11
5.3. White Space Channel Query . . . . .	14
5.4. Validation Procedure . . . . .	18
5.5. White Space Channel Update . . . . .	20
5.6. Result Codes . . . . .	20
6. Message Encoding . . . . .	22
6.1. XML Schema Definition . . . . .	22
6.2. HTTP Encoding . . . . .	27
7. Security Considerations . . . . .	33
8. IANA Considerations . . . . .	34
9. Acknowledgements . . . . .	35
10. References . . . . .	36
10.1. Normative References . . . . .	36
10.2. Informative References . . . . .	36
Authors' Addresses . . . . .	37

## 1. Introduction

"White Space" means the radio spectrum that has been allocated for some primary use, but is not fully occupied by that primary use at a specific location and time. Currently the white space in television bands (which is called TV white space (TVWS)) is widely discussed; TVWS has some good characteristics such as propagation characteristics and low power consumption. The regulatory bodies in some countries have created rules allowing secondary white space access; the secondary user must ensure it does not interfere with the primary user when using white space. The purpose of white space study is to design a mechanism that enables the secondary user to use the white space resource without interfering with the primary user. The widely accepted scheme of utilizing white space is by querying a database. This document defines a protocol over which such a database may be queried, called the "Protocol to Access White Space database (PAWS)". The use cases and requirements of PAWS have been discussed in another document [2].

The master devices may be produced by different manufacturers and there may be multiple databases serving a geographic area administered by different administrators. To ensure interoperability between these devices and databases, a standard interface needs to be defined. This document defines that interface.

Spectrum management rules of different spectrum regulatory bodies are different, so the white space spectrum databases may be designed to implement different spectrum policies in different regulatory domains. In order to satisfy the needs of these disparate regulatory domains, the database query protocol MUST be independent of different spectrum management rules. PAWS is a protocol between a master device and a database that carries information about white space spectrum from the database to a master device. The master device could act as a WiFi AP or a cellular base station (e.g. 3GPP LTE eNodeB) in the whitespace spectrum; the PAWS protocol is agnostic to the technology used by the master device. A slave device is the device which uses the spectrum made available by a master device. After the master device has obtained information about white space spectrum from the database and formed a wireless access network, the slave device can access it.

In this document we introduce a framework for the PAWS protocol, a protocol stack defining an interface between a master device and a whitespace database, a set of messages and their associated parameters, and an XML schema encoding the messages and parameters. Co-existence of multiple whitespace devices in the same geographic area and interference avoidance between white space devices within the same spectrum are out of scope of the current protocol.

Provisioning and how databases store the white space information are also out of scope of the protocol.

There is much sensitive information, such as location and identity, which MAY be transmitted between the master device and the database when PAWS is used. Attackers may attempt to obtain such information during the operation of the protocol. Therefore, the messaging interface between the master device and the database needs to be secured. Meanwhile, the two entities SHALL be the authorized and mutually authenticated. This document assumes that PAWS can be run over an HTTPS connection, but details of how security credentials are issued, managed, and validated among the various entities (databases, master devices and slave devices) are out of scope of the basic protocol and should be specified in a different document.

Given that multiple databases may serve a given region, and that a master device may move from region to region, a mechanism to discover the proper database to query must be provided in the master device. This document provides an overview of possible mechanisms that can be used for this purpose, but does not define any new protocols in this area.

## 2. Terminology and Abbreviation

### 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 [1].

### 2.2. Terminology

The Terminology Section of the latest version of the PAWS Problem Statement and Use Cases draft [2] shall be included by reference.

#### WS Interface

The interface between master device and whitespace database, including the data model and protocol messages defined in this document.

#### RAT

Radio Access Technology.



### 3. Overview of PAWS

We first define the entities of Master Device and Database, and the common interface between these two entities.

Figure 1 shows a common system model consisting of Master Device and Database. The Master Device connects to the database directly using the WS interface.

This document defines the data model and protocol messaging procedures of the WS interface. The messages of WS are encoded in XML, with security provided by HTTPS, and reliable in-order delivery provided by TCP. More details about the protocol of WS see section "protocol stack".

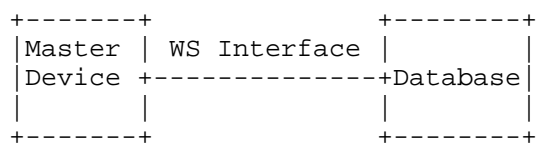


Figure 1: System Model of PAWS

The master device in Figure 1 queries for a list of available channels from the database so it can provide radio access for user equipments. It can be a WiFi access point, a base station of 3GPP WCDMA or 3GPP LTE, or some other RAT. The master device can send its own information (such as device ID, geo-location etc.) to the database to query white space spectrum for itself, or it can send the information from other devices to the database to query white space spectrum for other devices.

The database in Figure 1 is in charge of storing and maintaining white space channel information for certain area(s). There may be one or more databases providing white space information for a given area. The main function of the database is to provide suitable white space spectrum information to master devices. The databases are assumed to be on the Internet and can be accessed by the master devices via any Internet connection. When the database receives a request for white space spectrum from the master device, it will respond with a list of available white space channels to the master device if there are available channels. How the database stores the white space spectrum information and the policies for which white space spectrum can be returned to a master device is outside the scope of this document.

As shown in Figure 2 in order to provide wireless access based on white space, there are several procedures involved. These include

database discovery, secure connection establishment, device registration, and white space channel query.

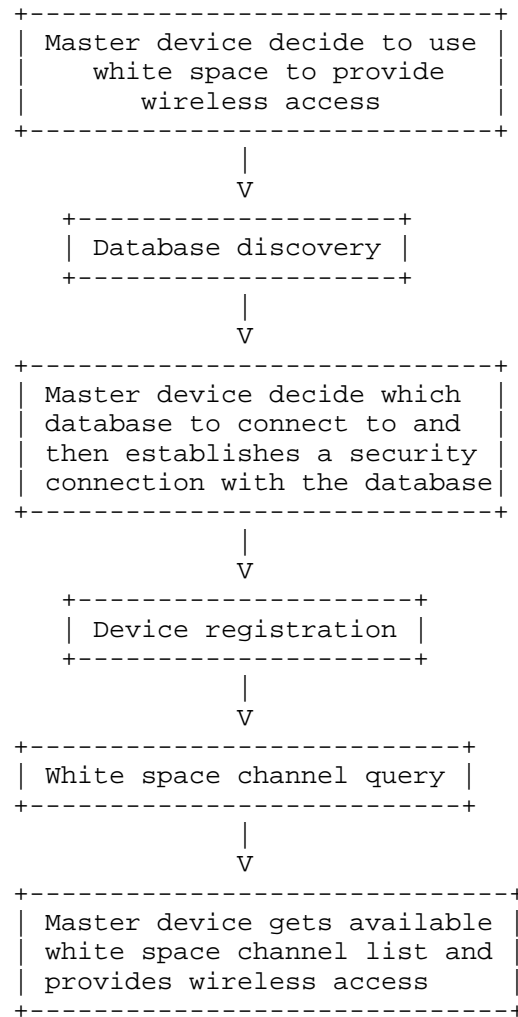


Figure 2: The procedure for getting white space channel for master device

When the master device is to provide radio access service, it is required to execute the following steps:

1. Database discovery. When the master device needs to connect to database, it must know the Internet address of the database and it has to decide to which database to connect when there is more

than one database. A database discovery mechanism is needed. There may be several mechanism for database discovery, for example, DNS.

2. Registration. Registration is an optional procedure of PAWS. In particular, requirements for registration come from individual regulatory domains and can be different depending on the regulator's individual requirements. When registration is used, before the database provides information on available radio channels, the master device MUST register with the trusted database. In the registration procedure, the information may include but is not limited to device ID, device owner's name, device owner's email address, device owner's phone number etc.
3. White space channel query. The white space channel query procedure from master device to database is based on a client-server model. When a master device is to create a radio network using white space, it queries available white space channel information from the database by sending a query message and receiving a response containing available whitespace channel(s).
4. White space channel update. The white space channel returned to master device is available for a limited duration of time, which means that when this time expires the channel can not be used by the master device any more, and then the master device must obtain updated white space channel information from the database. There are also some requirements from regulatory bodies that the white space channel information MUST be updated periodically. The update mechanism is necessary and is needed to avoid interfering with the primary user or other secondary user. A mechanism to update the whitespace information is provided in this draft.

Considering the security aspects, there is a trust relationship between the database and master devices. There SHOULD be corresponding authentication, integrity protection, and confidentiality protection mechanisms between the master device. Security considerations are given in Section 7; details of the security procedure at the beginning of an HTTPS connection is not included in this document.

#### 4. Protocol Stack

The protocol specified here is an application protocol that depends on a lower-layer transport protocol, which must provide the necessary features and security properties for use as the building blocks for communication between location-aware devices and white space databases. The service model between master device and database is client-server using request/response messages.

A protocol stack model is proposed here, shown in Figure 3. The transport layer is TCP and the application runs over HTTPS.

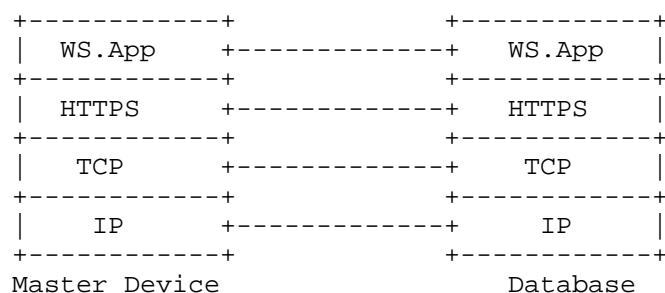


Figure 3: Protocol Stack of PAWS

WS.App is the white space spectrum application protocol. The messages of WS are encoded in XML, packaged in HTTP requests and responses, encrypted by TLS, and transported by TCP. The element types used in the XML encoding of messages are defined in Section 6.

## 5. Protocol Framework and Interface of PAWS

The use of white space spectrum is enabled after a white space device queries a database and obtains information about the availability of spectrum for use at a given location. The databases are reachable via the Internet and the devices querying these databases are expected to have some form of Internet connectivity. There could be multiple databases serving white space devices and a master device can select one of them for use. The architecture is shown in Figure 4.

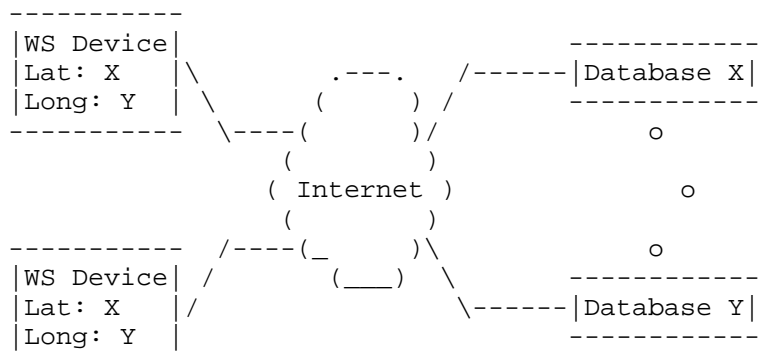


Figure 4: High Level View of the White Space Database Architecture

A messaging interface between the white space devices and the database is required for operating a network using the white space spectrum. The following sections discuss various aspects of such an interface. Other aspects of a solution including provisioning the database and calculating protected contours are considered out of scope of this document.

In order to query white space channel(s) from a database, a master device must provide its geo-location information to the database. There are several different methods for a master device to get its geo-location information; for example, using GPS technology, using street number or building location information etc.

The protocol message interface defines the message contents and message format of WS. The message interface should satisfy the following requirements:

1. The message sent in the message interface should be independent of the specific radio interface technologies (e.g. 802.11af, IEEE 802.16, IEEE 802.22, LTE);

2. The message interface should be spectrum agnostic. The message interface should not only be used for TV white space but also be used for other spectrum;
3. The message interface should satisfy different scenarios for using white space. In different scenarios the white space device's coverage area and the bandwidth may be different;
4. The message should address different regulations by different regulatory bodies;
5. Security requirements, such as ciphering and integrity protection must be met.

#### 5.1. Database Discovery

Before the white space device can transmit in white space spectrum, it MUST contact a trusted database where the device can learn if any channels are available for it to use. In order to connect to the trusted database the master device MUST get the IP address of the database.

The master device MAY be pre-programmed with the Internet IP address of trusted database manually. The master device can establish contact with a trusted database using one of the pre-provisioned IP addresses. We call this method "static database discovery".

Alternatively, the master device may discover the IP address of the database dynamically through the use of a DNS query. It may be configured with the DNS name of a database that is valid for its current location or may discover the name of an appropriate database through means outside the scope of this specification.

#### 5.2. Device Registration with Trusted Database

A registration procedure is used to register the master device's information in the database. Some databases may refuse to respond to queries from unauthorized or uncertified devices. The registration procedure is optional; master devices may not be required to register, depending on the regulator's requirements. When registration is used, before the database will provide information on available radio channels, the master device must register with the trusted database. In the registration procedure, the information may include but is not limited to, device ID, device owner's name, device owner's email address, device owner's phone number etc.

Specific events will initiate registration; these events are determined by regulator policy, for examples:

1. The master device will operate in white space for the first time after power up.
2. The location of master device changes by a predetermined distance.
3. After a certain regular time interval.
4. When the registered information changed, and the master device need update its registration information.

The device registration procedure consists of two messages:

1. Registration request message. This message is from master device to database. The master device shall provide to the database during registration all information required according to local regulatory requirements.
2. Registration response message. This message is from database to master device. The database responds to the registration request with an acknowledgement code to indicate the success or failure of the registration request. Additional information may be provided according to local regulator requirements.

One of two possible results shall be returned by the database:

1. Successful Registration.
2. Failed Registration. The master device is not recognized or authorized by the database.

A successful registration will overwrite any previous registration information for the same master device, as identified by device ID and manufacturer's serial number.

The device registration procedure is depicted in Figure 5.  
REGISTRATION\_REQ is the registration request message;  
REGISTRATION\_RESP is the registration response message.

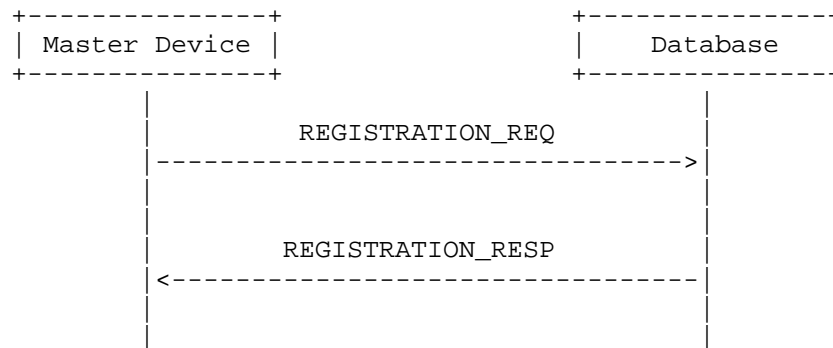


Figure 5: Device Registration with Database Procedure

The registration procedure consists the following steps:

1. The master device sends registration request message to the trusted database. In this message the parameters to be registered are included.
2. The database sends the registration response message to the master device indicating whether the registration is successful or not, Additional information may be provided according to local regulator requirements.

The parameters included in REGISTRATION\_REQ are as follows:

Parameter	Description
device id	Device id of master device.
device type	Device type defined by Regional Regulators, including fixed, mobile, portable, etc.
manufacturer's serial number	The manufacturer's serial number of device.
device owner's information	Includes: name of the individual or business that owns the device, name of a contact person responsible for the device's operation, address for the contact person, and email address for the contact person and phone number of the contact person.

Table 1: Parameters of the REGISTRATION\_REQ Message



The parameters included in the REGISTRATION\_RESP are as follows:

Parameter	Description
result code	Consists of a code number with related description in text which indicates whether the registration request is successful or failed; if it failed the result code will indicate the reason of failure.

Table 2: Parameters of the REGISTRATION\_RESP Message

### 5.3. White Space Channel Query

When master device is to create a radio network using white space, it queries for available white space channels from the database. The master device sends a white space channel query message to the database and fetches white space channel(s) from the database.

The channel query procedure consists of four messages:

1. Channel query request message. This message is from the master device to the database. The channel query request message takes parameters as required by local regulatory requirements to the database; these parameters will be used by the database to decide the available white space channel(s) for the master device;
2. Channel query response message. This message is from the database to the master device. The channel query response message takes parameters as required by local regulatory requirements to the master device; the white space query result code of success or fail will be included in this message. If there are available white space channel(s) for the master device, the result code of success will be returned to the master device and the availability white space channel(s) with related information will be returned to the master device; otherwise, if there is no available white space channel for the master device, the result code of failure with the failure reason will be returned to the master device;
3. Channel usage report message. This message is from the master device to the database. When the master device receives the white space channel(s) returned from the database, it uses this message to inform the database of the anticipated channel usage. Because not all of the regulatory rules require the reporting back of usage, some databases may not support this message, so it is optional.

4. Channel usage acknowledge message. This message is from the database to the master device. This message is an acknowledgement of the channel usage report message. This message will be sent only when the channel usage report message is used.

The white space channel query procedure is depicted in Figure 6. AVAIL\_WS\_REQ is the available white space query request message; AVAIL\_WS\_RESP is the available white space query response message; CHANNEL\_USAGE\_REPORT is the channel usage report message; CHANNEL\_USAGE\_ACK is the channel usage acknowledge message.

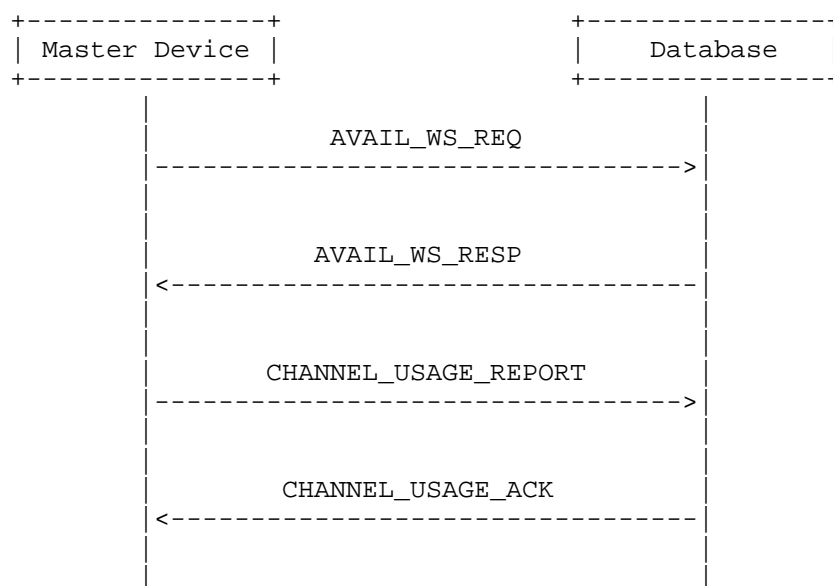


Figure 6: The Available Channel Query Procedure

The query procedure using the following steps:

1. The master device sends the white space query request message to database and waits a limited period of time for white space query response message from the database. When the time expires and no query response message is returned from the database, the query procedure will be failed.
2. On receiving the white space query request message, database will find out the available white space channels and send the available white space channel list to the white space device. The result code in the query response message (AVAIL\_WS\_RESP message) will indicate whether the channel usage report message

is needed to be sent.

3. If the channel usage message is needed, the master device will send the channel usage message to the database after receiving the query response message. If there is no available channel or no acceptable response is received within the limited time, the master device concludes that no channel is available.
4. When the database receives channel usage report message, it will acknowledge the master device of receiving the message by channel usage acknowledge message.

The parameters included in AVAIL\_WS\_REQ are as follows:

Parameter	Description
device id	Device id of master device that sends the query message.
device type	Device type defined by Regional Regulators, including fixed, mobile, portable, etc.
List of coverage area(s) information	A list of coverage area(s) where white space access service will be provided. This field includes: geo-location (latitude, longitude) of the master device, uncertainty of geo-location (in meters), and confidence (in percentage) for the location determination, coverage range.
antenna characteristics	Antenna characteristics of the master device that will use the white space. This field includes: antenna height above the ground, antenna direction, antenna radiation pattern, antenna gain, maximum output power, spectrum mask.
RAT type	Specifies information about the type of RAT of the master device.
bandwidth	Bandwidth that the master device needs to form the wirelss network.

Table 3: Parameters of the AVAIL\_WS\_REQ Message

The parameters included in AVAIL\_WS\_RESP are as follows:

Parameter	Description
device id	Device id of master device, the value of this field is the same as the device id in AVAIL_WS_REQ message.
device type	Device type defined by Regional Regulators, including fixed, mobile, portable, etc. The value of this field is the same as the device type in AVAIL_WS_REQ message.
result code	Consists of a code number with related description in text which indicates whether the available white space request is successful or failed; if it has failed the result code will indicate the reason of failure.
white space channel list	This field includes: frequency information, available bandwidth, available time duration, coverage area, maximum transmission power.

Table 4: Parameters of the AVAIL\_WS\_RESP Message

The parameters included in CHANNEL\_USAGE\_REPORT are as follows:

Parameter	Description
device id	Device id of master device that sends the query message.
white space channel list	This field includes: frequency information, available bandwidth, available time duration, coverage area, maximum transmission power.

Table 5: Parameters of the CHANNEL\_USAGE\_REPORT Message

The parameters included in CHANNEL\_USAGE\_ACK are as follows:

Parameter	Description
result code	Consists of a code number with related description in text which indicates whether the CHANNEL_USAGE_REPORT message is received by the database.

Table 6: Parameters of the CHANNEL\_USAGE\_ACK Message

#### 5.4. Validation Procedure

The validation procedure is used for the database to validate the slave device. When the slave device connects to the master device, the master device MAY start the validation procedure to validate the slave device.

The validation procedure consists of two messages:

1. Validation request message. This message is from master device to database. After the slave device connects to the master device, the master device can send the slave's validation information such as slave device ID, slave device's manufacturer serial number etc to the database in validation request message.
2. Validation response message. This message is from the database to the master device. This message is used to indicate if the slave device is validated by the database.

The validation procedure is depicted in Figure 7. VALIDATION\_REQ is the validation request message; VALIDATION\_RESP is the validation response message.

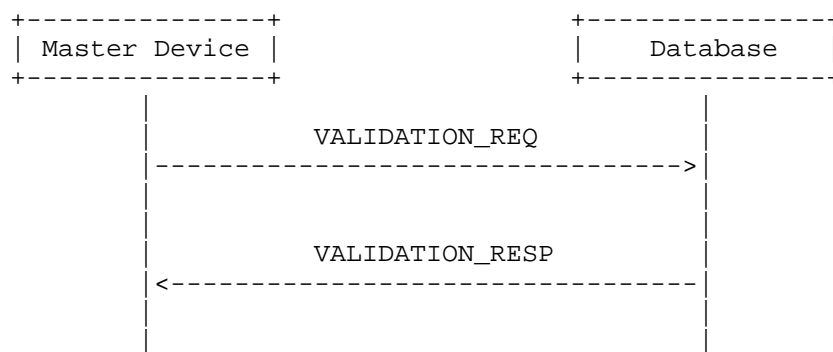


Figure 7: Slave Device Validation with Database Procedure

The validation procedure using the following steps:

1. After the slave device has connected to the master device and sent its slave device id and slave device type to the master device, the master device will send slave device id included in VALIDATION\_REQ message to the database.
2. The database validates the slave device and returns the result code to the master device. The result code indicates whether the slave device is validated or not, and if the slave device is not validated the result code will indicate the reason of not validated.
3. If the the slave device is validated by the database, the white space device will provide service to the slave device, otherwise the master device will deny the slave device's access.

The parameters included in VALIDATION\_REQ are as follows:

Parameter	Description
device id	Slave device id.
device type	Slave's device type defined by Regional Regulators, including fixed, mobile, portable, etc.
device owner's information	Identification of the individual or business that owns the device.

Table 7: Parameters of the VALIDATION\_REQ Message

The parameters included in VALIDATION\_RESP are as follows:

Parameter	Description
result code	Consists of a code number with related description in text which indicates whether the slave device is validated or not; if it's failed the result code will indicate the reason for failure.

Table 8: Parameters of the VALIDATION\_RESP Message

### 5.5. White Space Channel Update

The availability of a white space channel may be changed, because a primary user may obtain the channel.

In order to avoid interfering with the primary user or other secondary user, the white space updating mechanism is provided in this document.

The white space channel update procedure is used for master device to update the white space channel from the database. The update procedure SHOULD be implemented when one of the followings occurs:

1. Periodically implemented as required by the regulation to verify that the operating channels continue to remain available.
2. When master device changes its location more than a threshold distance.

The white space device MUST access the database to obtain and update the list of available channels that could be utilized by the device to verify that the operating channels continue to remain available. According to some regulatory rules the white space device SHOULD update the white space channel periodically, and the period may be different due to different regulatory rules.

The white space channel update mechanism is based on the white space channel query procedure. After the master device gets a white space channel from the database, a white space channel update timer is set to a certain value, which is determined by local regulatory body, when the timer expires the master device will start white space channel query procedure to query white space channels from the database.

When the master device changes its location more than a threshold distance it SHOULD query the database for available operating channels, the value of threshold is specified by local regulatory policy.

### 5.6. Result Codes

The following result codes are provided by the database on responses to the master device to communicate the status of requests made by the master device; all of the result codes used in this document is defined here.

Code	Description	Returned Text
0	successful	"successful"
1	successful with no channel available	"no channel available"
2	Successful and CHANNEL_USAGE_REPORT message needs to be sent.	"CHANNEL_USAGE_REPORT message needs to be sent"
3	Successful and CHANNEL_USAGE_REPORT message needs not to be sent.	"CHANNEL_USAGE_REPORT message needs not to be sent"
4	device id is invalid	"device id is invalid"
5	device type is invalid	"device type is invalid"
6	device type is not supported	"device type is not supported"
7	Device has not registered	"Device has not registered"

Table 9: Result Codes



## 6. Message Encoding

### 6.1. XML Schema Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://tools.ietf.org/wg/paws/"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <!-- Definition of element Types-->

  <!--definition of the element type of protocol version-->
  <xs:simpleType name="versionType">
    <xs:restriction base="xs:byte"/>
  </xs:simpleType>

  <!--definition of the element type of device's ID-->
  <xs:simpleType name="deviceIDType">
    <xs:restriction base="xs:string">
      <xs:length value="20"/>
    </xs:restriction>
  </xs:simpleType>

  <!--definition of the element type of device -->
  <xs:simpleType name="deviceType">
    <xs:restriction base="xs:integer"/>
  </xs:simpleType>

  <!--definition of the element type of manufacture series number-->
  <xs:simpleType name="manufactureSeqNumType">
    <xs:restriction base="xs:string">
      <xs:length value="32"/>
    </xs:restriction>
  </xs:simpleType>

  <!--definition of the element type of WS device information-->
  <xs:complexType name="deviceOwnerInfoType">
    <xs:sequence>
      <xs:element name="nameOfOwner" type="xs:string"/>
      <xs:element name="nameOfOperator" type="xs:string"/>
      <xs:element name="addressOfOperator" type="xs:string"/>
      <xs:element name="phoneNumberOfOperator"
        type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <!--definition of element type of geo-location-->
  <xs:complexType name="geoLocationType">
```

```

        <xs:sequence>
          <xs:element name="latitude" type="xs:string"/>
          <xs:element name="longitude" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>

      <!--definition of element type of coverage range-->
      <xs:simpleType name="coverageRangeType">
        <xs:restriction base="xs:float">
          <xs:minInclusive value="0"/>
        </xs:restriction>
      </xs:simpleType>

      <!--definition of element type of coverage area-->
      <xs:complexType name="coverageAreaType">
        <xs:sequence>
          <xs:element name="geoLocation" type="geolocationType"/>
          <xs:element name="uncertaintyOfGeoLocation" type="xs:float"/>
          <xs:element name="confidence" type="xs:float"/>
          <xs:element name="coverageRange" type="coverageRange"/>
        </xs:sequence>
      </xs:complexType>
      <!--definition of element type of list of coverage area-->
      <xs:complexType name="coverageAreaListType">
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
          <xs:element name="coverageArea"
            type="coverageAreaType"/>
        </xs:sequence>
      </xs:complexType>

      <!--definition of element type of result code-->
      <xs:complexType name="resultType">
        <xs:sequence>
          <xs:element name="code" type="xs:byte"/>
          <xs:element name="discription" type="xs:string"/>
        </xs:sequence>
      </xs:simpleType>

      <!--definition of element type of antenna characteristics

Unit:
antenna gain          db
antenna height        m
antenna direction     rad
maximum output power  dbm
-->

      <xs:complexType name="antennaCharacterType">

```

```
<xs:sequence>
  <xs:element name="antennaHeight" type="xs:float"/>
  <xs:element name="antennaGain" type="xs:float"/>
  <xs:element name="antennaDirection" type="xs:float"/>
  <xs:element name="maxOutputPower" type="xs:float"/>
</xs:sequence>
</xs:complexType>

<!--definition of element type of bandwidth
unit:
bandwidth                kHz
-->
<xs:simpleType name="bandwidthType">
  <xs:restriction base="xs:float"/>
</xs:simpleType>

<!--definition of element type of white space channel
unit:
frequency                kHz
-->
<xs:complexType name="channelType">
  <xs:sequence>
    <xs:element name="frequency" type="xs:float"/>
    <xs:element name="bandwidth" type="bandwidthType"/>
  </xs:sequence>
</xs:complexType>

<!--definition of element type of RAT-->
<xs:simpleType name="RATType">
  <element name="rat" type="xs:string"/>
</xs:simpleType>

<!--definition of element type of available duration time
of channel-->
<xs:complexType name="timeDurationType">
  <xs:restriction base="xs:dateTime">
    <xs:element name="beginTime"
      type="timeDurationType"/>
    <xs:element name="endTime"
      type="timeDurationType"/>
  </xs:restriction>
</xs:simpleType>

<!--definition of element type of maximum transmit power
unit:
```

```
maximum transmit power          dbm
-->
<xs:simpleType name="maxTransmitPowerType">
  <xs:restriction base="xs:float"/>
</xs:simpleType>

<!--definition of element type of list of channel-->
<xs:complexType name="channelListType">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="channel" type="channelType"/>
    <xs:element name="timeDuration" type="timeDurationType"/>
    <xs:element name="maxTransmitPowerType"/>
    <xs:element name="coverageArea" type="coverageAreaType"/>
  </xs:sequence>
</xs:complexType>

<!-- Definition Of Messages-->

<!--definition of registration request message-->
<xs:element name="REGISTRATION_REQ_MSG">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="version" type="versionType"/>
      <xs:element name="deviceID" type="deviceIDType"/>
      <xs:element name="device" type="deviceType"/>
      <xs:element name="manufactureSeqNum" type="manufactureSeqNumType"/>
      <xs:element name="deviceOwnerInfo" type="deviceOwnerInfoType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--definition of registration response message-->
<xs:element name="REGISTRATION_RESP_MSG">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="version" type="versionType"/>
      <xs:element name="result" type="resultType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--definition of availabel ws query request message-->
<xs:element name="AVAIL_WS_REQ_MSG">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="version" type="versionType"/>
```

```
<xs:element name="device" type="deviceType"/>
  <xs:element name="deviceID" type="deviceIDType"/>
  <xs:element name="coverageAreaList" type="coverageAreaListType"/>
  <xs:element name="antennaCharacter" type="antennaCharacterType"/>
<xs:element name="rat" type="RATType"/>
<xs:element name="bandwidth" type="bandwidthType"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<!--definition of availabel ws query response message-->
<xs:element name="AVAIL_WS_RESP_MSG">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="version" type="versionType"/>
    <xs:element name="deviceID" type="deviceIDType"/>
    <xs:element name="device" type="deviceType"/>
      <xs:element name="result" type="resultType"/>
    <xs:element name="channelList" type="channleListType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

<!--definition of channel usage report message-->
<xs:element name="CHANNEL_USAGE_REPORT">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="version" type="versionType"/>
    <xs:element name="deviceID" type="deviceIDType"/>
    <xs:element name="channelList" type="channleListType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

<!--definition of channel usage report acknowledge message-->
<xs:element name="CHANNEL_USAGE_ACK">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="version" type="versionType"/>
    <xs:element name="result" type="resultType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

<!--definition of validation request message-->
<xs:element name="VALIDATION_REQ">
```

```

    <xs:complexType>
      <xs:sequence>
        <xs:element name="version" type="versionType"/>
        <xs:element name="deviceID" type="deviceIDType"/>
        <xs:element name="device" type="deviceType"/>
        <xs:element name="deviceOwnerInfo" type="deviceOwnerInfoType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!--definition of validation response message-->
  <xs:element name=" VALIDATION_RESP ">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="version" type="versionType"/>
        <xs:element name="result" type="resultType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

## 6.2. HTTP Encoding

This section will describe how to encode the PAWS protocol message of XML format in HTTP protocol. The PAWS protocol messages of XML format are carried as an entity of HTTP message.

Here are some examples of how to encode PAWS messages of XML format in HTTP protocol.

### 1. Registration Request message.

PUT {URL} HTTP/1.1

Accept: \*/\*

Proxy-Connection: Keep-Alive

Host: {host name}

Content-Type: text/xml

```

<?xml version="1.0" encoding="UTF-8"?>
< REGISTRATION_REQ_MSG xmlns="http://tools.ietf.org/wg/paws/">
  < version>{_version}</version>
  < deviceID >{_deviceID}</deviceID>
  < device >{_ device }</ device >
  < manufactureSeqNum >{_ manufactureSeqNum }</ manufactureSeqNum >
  < deviceOwnerInfo >{_ deviceOwnerInfo }</ deviceOwnerInfo >
</ REGISTRATION_REQ_MSG >

```

where

{URL} is the URL of the database.

{host name} is the host name of the database.

{\_version}, {\_deviceID}, {\_device}, {\_manufactureSeqNum },  
{\_antennaCharacter }, {\_deviceOwnerInfo } are the elements  
defined in the XML schema.

## 2. Registration Response message.

HTTP/1.1 200 OK

Cache-Control: private

Content-Length: {LENGTH}

Content-Type: application/xml; charset=utf-8\r\n

```
<?xml version="1.0" encoding="UTF-8"?>
< REGISTRATION_RESP_MSG xmlns="http://tools.ietf.org/wg/paws/">
  <version>{_version}</version>
  < result >{_result}</ result >
</ REGISTRATION_RESP_MSG >
REGISTRATION_RESP_MSG
```

where

{LENGTH} is the length of the XML body.

{\_version}, {\_result} are the elements defined in the XML  
schema.

## 3. Available white space channel query request message.

GET {URL} HTTP/1.1

Accept: \*/\*

Proxy-Connection: Keep-Alive

Host: {host name}

Content-Type: text/xml

```
<?xml version="1.0" encoding="UTF-8"?>
< AVAIL_WS_REQ_MSG xmlns="http://tools.ietf.org/wg/paws/">
<version>{_version}</version>
< deviceID >{_deviceID}</deviceID>
< device >{_device }</ device >
< coverageAreaList >{_ coverageAreaList }</ coverageAreaList >
< antennaCharacter >{_ antennaCharacter }</ antennaCharacter >
<rat>{_rat}</rat>
< bandwidth >{_ bandwidth }</ bandwidth >

</ AVAIL_WS_REQ_MSG >
```

where

{URL} is the URL of the database.

{host name} is the host name of the database.

{\_version}, {\_deviceID}, {\_device}, {\_coverageAreaList },  
{\_antennaCharacter }, {\_rat}, {\_bandwidth } are the elements  
defined in the XML schema.

#### 4. Available white space channel query response message.

HTTP/1.1 200 OK

Cache-Control: private

Content-Length: {LENGTH}

Content-Type: application/xml; charset=utf-8\r\n

```
<?xml version="1.0" encoding="UTF-8"?>
< AVAIL_WS_RESP_MSG xmlns="http://tools.ietf.org/wg/paws/">
<version>{_version}</version>
< deviceID >{_deviceID}</deviceID>
< device >{_device }</ device >
< result >{_result }</ result >
< channelList>{_channelList}</ channelList>
</ AVAIL_WS_RESP_MSG >
```

where



{LENGTH} is the length of the XML body.

{\_version}, {\_deviceID}, {\_device }, {\_result} are the elements defined in the XML schema.

5. Channel usage report message.

PUT {URL} HTTP/1.1

Accept: \*/\*

Proxy-Connection: Keep-Alive

Host: {host name}

Content-Type: text/xml

```
<?xml version="1.0" encoding="UTF-8"?>
< CHANNEL_USAGE_REPORT xmlns="http://tools.ietf.org/wg/paws/">
<version>{_version}</version>
< deviceID >{_deviceID}</deviceID>
< channelList>{_ channelList}</ channelList>
</ CHANNEL_USAGE_REPORT >
```

where

{URL} is the URL of the database.

{host name} is the host name of the database.

{\_version}, {\_deviceID}, {\_channelList} are the elements defined in the XML schema.

6. Channel usage report acknowledge message.

HTTP/1.1 200 OK

Cache-Control: private

Content-Length: {LENGTH}

Content-Type: application/xml; charset=utf-8\r\n

```
<?xml version="1.0" encoding="UTF-8"?>
< CHANNEL_USAGE_ACK xmlns="http://tools.ietf.org/wg/paws/">
<version>{_version}</version>
< result >{_ result }</ result >
</ CHANNEL_USAGE_ACK >
```

where

{LENGTH} is the length of the XML body.

{\_version}, {\_ result} are the elements defined in the XML schema.

#### 7. Validation request message.

PUT {URL} HTTP/1.1

Accept: \*/\*  
Proxy-Connection: Keep-Alive  
Host: {host name}  
Content-Type: text/xml

```
<?xml version="1.0" encoding="UTF-8"?>
< VALIDATION_REQ xmlns="http://tools.ietf.org/wg/paws/">
  <version>{_version}</version>
  < deviceID >{_deviceID}</deviceID>
  <device>{_ device }</ device >
  < deviceOwnerInfo >{_ deviceOwnerInfo }</ deviceOwnerInfo >
</ VALIDATION_REQ >
```

where

{URL} is the URL of the database.

{host name} is the host name of the database.

{\_version}, {\_deviceID}, {\_ device }, {\_ deviceOwnerInfo } are the elements defined in the XML schema.

#### 8. Validation response message.

HTTP/1.1 200 OK

Cache-Control: private  
Content-Length: {LENGTH}  
Content-Type: application/xml; charset=utf-8\r\n

```
<?xml version="1.0" encoding="UTF-8"?>
< VALIDATION_RESP xmlns="http://tools.ietf.org/wg/paws/">
  <version>{_version}</version>
  < result >{_ result }</ result >
</ VALIDATION_RESP >
```

where

{LENGTH} is the length of the XML body.

{\_version}, {\_ result} are the elements defined in the XML schema.

## 7. Security Considerations

There is much sensitive information, such as location and identity, which may be transmitted between the master device and the database when PAWS is used. According to the security requirements given in the problem statement draft [2] attackers may have full access to the network medium between the master device and the white space database and many types of attack may be carried out by the attackers if there is a lack of security in PAWS. Therefore, to guarantee the security considerations of the communication between the master device and the white space database, the following security features should be considered:

1. The identity of the master device and the white space database must be authenticated, namely the mutual authentication must be implemented and an authorization check shall be carried out by both of them.
2. The connection between the master device and white space database must be private; that means the messages transmitted on the connection between the master device and the white space database are confidentiality protected.
3. The connection between the master device and white space database is reliable; that means that the message transport must support including a message integrity check.
4. The negotiation of a shared key is secure: the negotiated secrets which are used for confidentiality protection and Integrity protection are unrevealed to eavesdroppers.
5. The negotiation is confidential: attacker must be not able to modify the content of negotiation process without being detected by the endpoints during the communication.

The security threats, security features and security countermeasures associated with the use of white space spectrum by secondary usages via PAWS are not discussed in details in this document.

## 8. IANA Considerations

There have been no IANA considerations so far in this document.

## 9. Acknowledgements

Thanks to my colleagues for their sincerely help and comments when drafting this document.

## 10. References

### 10.1. Normative References

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

### 10.2. Informative References

- [2] Probasco, S. and B. Patil, "Protocol to Access White Space database: PS, use cases and rqmts", draft-ietf-paws-problem-stmt-usecases-rqmts-06 (work in progress), July 2012.

Authors' Addresses

Xinpeng Wei  
Huawei

Phone: +86 13436822355  
Email: weixinpeng@huawei.com

Zhu Lei  
Huawei

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

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

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





PAWS  
Internet-Draft  
Intended status: Informational  
Expires: January 10, 2013

Y. Wu  
Y. Cui  
Huawei  
July 9, 2012

Protocol to Access White Space database: security considerations  
draft-wu-paws-secutity-00.txt

## Abstract

PAWS is an access protocol between the Master device and the White Space (WS) Database. Master Devices connect to the white space database directly using WS interface, but only authorized devices can get the service from database. If an attacker can have full access to the network medium between the master device and the database, the attacker may deploy varieties of attacks on the network if there is lack of security mechanism.

The present document describes the security threats to the current framework of PAWS, and meanwhile proposes the corresponding countermeasures.

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 10, 2013.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

1. Introduction . . . . .	4
2. Conventions and terminology . . . . .	4
2.1. Conventions used in this Document . . . . .	4
2.2. Terminology . . . . .	4
3. Overview of Security threats and Requirements . . . . .	5
3.1. Fundamental system architecture of PAWS . . . . .	5
3.2. Security threats . . . . .	5
3.2.1. Impersonation of a master device . . . . .	6
3.2.2. Impersonation of database . . . . .	6
3.2.3. MitM on the interface between master device and database . . . . .	6
3.2.4. Attacks on the link of interface between master device and database . . . . .	6
3.2.5. Attacks on the master device itself . . . . .	7
3.2.6. Other potential attacks(To be added) . . . . .	7
3.3. Security countermeasures . . . . .	7
4. Security schemes . . . . .	8
4.1. Overview . . . . .	8
4.2. Analysis of security schemes . . . . .	8
4.2.1. Databases deployed by third-party . . . . .	8
4.2.2. Databases deployed by regulatory body of white space . . . . .	12
4.3. TLS protocol . . . . .	13
4.3.1. Brief introduction of TLS protocol . . . . .	13
4.3.2. Security establishment procedure between master device and database . . . . .	13
4.3.3. Drawbacks of TLS protocol . . . . .	15
5. Security Considerations . . . . .	15
6. IANA Considerations . . . . .	15
7. Acknowledgments . . . . .	15
8. Normative Reference . . . . .	16
Authors' Addresses . . . . .	16

## 1. Introduction

Portions of the radio spectrums that are allocated to a licensed, primary user but are unused or unoccupied at specific locations and times are defined as "white space". The concept of allowing secondary transmissions (licensed or unlicensed) in white space is a technique to "unlock" existing spectrum. Currently, the widely accepted scheme of utilizing white space is by querying the database and the related protocol "Protocol to Access White Space database (PAWS)" is proposed.

Entities of master device and Database, the interface between the two entities would have been defined in PAWS. There are much sensitive information, such as location and identity of master devices, which MAY be transmitted between the interface of the master device and the white space database when the PAWS is used. Attackers are able to make various types of attack by using the sensitive information if there is lack of security mechanism. Therefore, the messaging interface between the master device and the database needs to be secured. Meanwhile, the two entities SHALL be mutually authenticated and they both MUST be authorized by authority of white space management institution.

In this document, the security threats, the security features and the security mechanism(TLS) are discussed in details.

## 2. Conventions and terminology

### 2.1. Conventions used in this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMEND", "MAY", and "OPTIONAL" that appear in this document are to be interpreted as described in [RFC2119].

### 2.2. Terminology

The Terminology Section of the latest version of [I-D.ietf-paws-problem-stmt-usecases-rqmts] shall be included by reference.

#### WS interface

The interface between master device and Database specifies data model and process of PAWS in this document.

### 3. Overview of Security threats and Requirements

#### 3.1. Fundamental system architecture of PAWS

Figure 1 shows a common system model of PAWS, the master device is connected to internet by any means other than using the white space radio. More details of PAWS are described using the following steps:

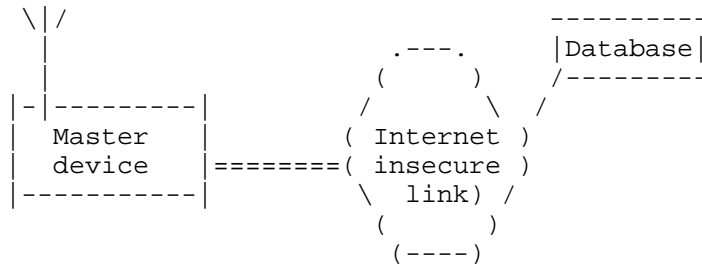


Figure 1: system architecture of PAWS

Description of system architecture:

- 1) The master device needs to discover white space database in the relevant regulatory domain by connecting to the internet by any means other than using the white space radio.
- 2) The master device will connect to the white space database, the link between master device and the white space database can be wired or wireless and provide IP connectivity, which may be insecure.
- 3) The master device may register with the white space database before the white space database provides information on available white space radio channels according to regulatory domain requirements. The information used in registration may include, but is not limited to, the device ID, the device's location, serial number assigned by the manufacturer and so on.
- 4) The master device queries the white space database for available white space channel lists.

#### 3.2. Security threats

If there is lack of security mechanism in above PAWS architecture, various threats detailed in "ietf-paws-problem-stmt-use-cases-rqmts" may exist, and the corresponding attacks can be deployed. It can be summarized as follows from a security point of view:

### 3.2.1. Impersonation of a master device

If there is no authentication of the master device, the white space database cannot detect the rogue master device, and the available white space channel list will be passed to the Rogue master device. This enables a rogue master device to use the available channels. Besides, the rogue master device can connect to the white space database by using the registration exchanges, the DoS type attacks may be initiated. This shows that it is essential to perform some type of authentication of master device.

### 3.2.2. Impersonation of database

If there is no authentication of the database, an attacker may attempt to spoof a white space database and provide responses to a master device which are malicious and result in the master device causing interference to the primary user of the spectrum. At the same time, the attacker also can retrieve an available white space channel list from a legal database using the registration exchanges, which received from the master device.

### 3.2.3. MitM on the interface between master device and database

A man in the middle (MitM) node is inserted in between the master device and database, it can be considered to be a variant of the above attacks. The real master device will connect to the MitM node and the MitM node can connect to the real database. The MitM node can transparently transmit, receive, view, and modify the traffic between the real master device and the database without either of those nodes being aware of it. The important security point illustrated by this attack is that not only is it essential to perform mutual authentication of the master device and the white space database, it is important to ensure that all security tunnels from the master device terminate in the trusted white space database instead of in a MitM node.

### 3.2.4. Attacks on the link of interface between master device and database

The link between the master device and the white space database can be wired or wireless. An attacker may listen to the communication between a valid master device and database. The threats of this are as follows:

- 1) Steal the confidentiality of data transmitted in the packet payload, such as utilize the information about available channels by utilizing those channels. The result of such an attack is unauthorized use of channels by an unauthorized device.

2) The location/identity information can be gleaned by an eavesdropper and be used for tracking purposes.

3) An attacker could modify the communication between the master device and the database. The channel information and some other type parameters could be modified by an attacker which may result in interference to the primary user of that channel. Alternatively the attacker may indicate no channel availability at a location resulting in a denial of service to the master device.

#### 3.2.5. Attacks on the master device itself

The master device may be deployed in vulnerable locations, and the less trusted types of transmission links will be used to interconnect that equipment to the database. Breaking the master device to get sensitive data is theoretically possible. The attacker may dig out the master device-database shared secret or a long term certificate from the master device and tries to add another master device.

#### 3.2.6. Other potential attacks(To be added)

### 3.3. Security countermeasures

To mitigate the above threats, the security countermeasures below should be used. Namely:

1) The master device shall be authenticated by database based on a globally unique and permanent master device identity when it wants to establish connection with the database.

2) The master device shall authenticate the identity of database.

3) The master device and the white space database shall check that both of them are authorized by the regulator body of white space.

4) Sensitive data including authentication credentials, user information, cryptographic keys shall not be transmitted between the master device and the white space database in plaintext in unauthorized access. It means that the transport of data over the interface between the master device and the database shall be integrity, confidentially, and replay protected from unauthorized.

5) The master device should have a secure module to store long term key or certificate. The identity of master device could be stored in a trusted physical module and/or a possible non-removable smartcard.



## 4. Security schemes

### 4.1. Overview

According to the previous analysis, the security mechanism shall be able to provide the following security features:

#### 1) Mutual authentication

Mutual authentication between the master device and the database shall be performed using certificates or pre-shared keys and so on. Besides, the master device and the database shall further be authenticated by the authority of regulatory body of white space to ensure that they are both authorized by regulatory body of white space due to the fact that the database may not be deployed by regulatory body of white space. The credentials and critical security functions for authentication shall be protected inside physically secure environment, such as Trusted Environment(TrE).

#### 2) The protection mechanisms of the data

The data over the interface between the master device and the database shall be protected for integrity, confidentiality and anti-replay from unauthorized parties.

#### 3) Trusted environment

The TrE shall be a logical entity which provides a trustworthy environment for the execution of sensitive functions and the storage of sensitive data. All data produced through execution of functions within the TrE shall be unknowable to unauthorized external entities.

### 4.2. Analysis of security schemes

For business reasons or ease of management, databases may be deployed by different third-party that is authorized by regulatory body of white space. It means that there are two possible deployment cases: one is that the databases deployed by the third-party which are authorized by regulatory body of white space; the other is that the databases are directly deployed by regulatory body of white space.

#### 4.2.1. Databases deployed by third-party

In this scene, when the master device wants to query the database for an available white space list, it shall be able to connect to the database and also be authorized by regulatory body of white space to use white space. It means that twice authentications shall be implemented, two suits of credentials are stored in master device and

white space database which are provided by different trusted authorities. There are two possible authentication models which are showed as follow:

Model 1

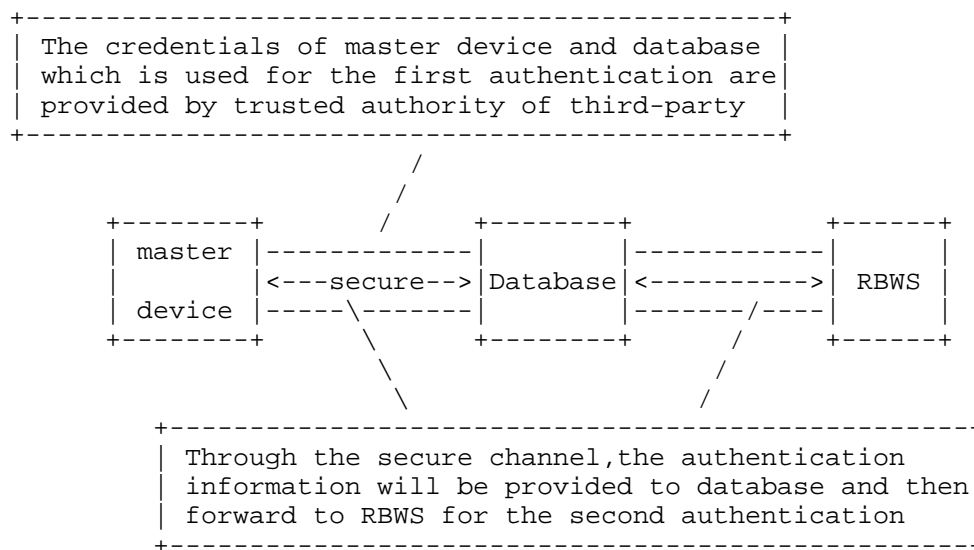


Figure 2: authentication model 1

In this model, the credentials of the master device and the database for second authentication both shall be checked by authority of RBWS after the master device successfully access to the white space database whenever the master device query the database, but what the database needs to do is able to establish connection with authority of regulatory body of white space (RBWS).

Under this circumstance, the whole querying procedure of white space channels can be showed as followed:

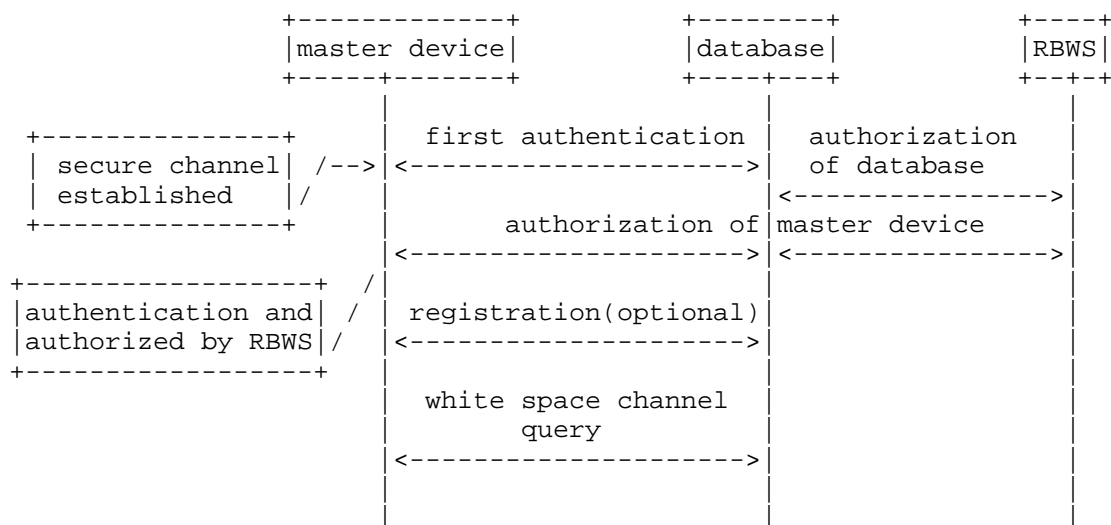


Figure 3: procedure of querying database

Firstly, the mutual authentication between the master device and the database shall be supported based on the credentials which are provided by an authority trusted by the third-party, e.g. the authority of third-party or by another party trusted by the third-party. Only the master device which has credentials with the third-party that deployed the database can connect to the database. And the master device also shall evaluate the connected database if it is a trusted database where the master device is able to register and receive service from the database. Namely the secure connectivity between the master device and the database shall be established first before the master device can query the available white space from the database.

Secondly, following the above successful mutual authentication between the master device and the database, they both shall also be authenticated and authorized by regulatory body of white space. The master device provides the information for authentication to database through the WS interface and then forward to regulatory body of white space by database. The credentials used for secondary authentication shall be provided by authority trusted by regulatory body of white space, e.g. the authority of regulatory body of white space or by another party trusted by the regulatory body of white space.

Finally, only when the twice mutual authentications are both passed,

the master device can query the database for available white space channels.

In addition, the information which is used for second authentication or for registration may be transmitted between master device and database after the first successful authentication, these information may contain sensitive data which shall not be known by others. So the secure channel shall be established after the first authentication which is used to provide security protection. The master device and the database shall not engage in any communication prior to the completion of the establishment of the secure channel other than messages for establishing the secure channel.

#### Model 2

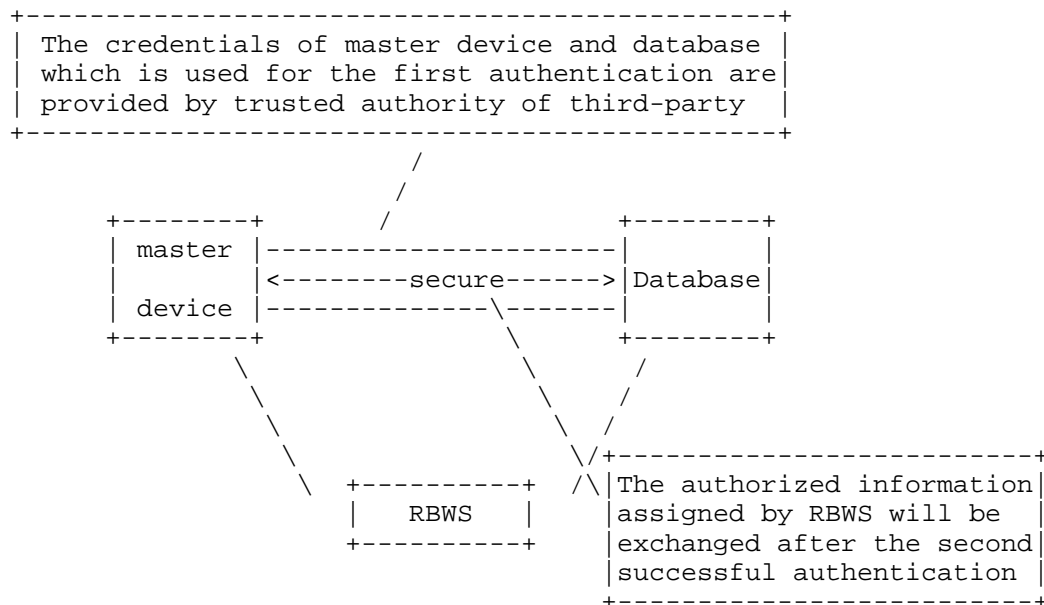


Figure 4: authentication model 2

In this model, the master device and database both can connect to regulatory body of white space.

On this occasion, the credentials provided by trusted authority of third-party are used to mutual authentication between the master device and the database first. The master device and the white space database must establish connection with RSWB respectively and authenticated by the authority of RBWS when the previous mutual

authentication is successful. Then the master device and database shall check whether the authorized information assigned by regulatory body of white space will be exchanged through the WS interface is legal. Only the two authentication procedures are both successful, the master device is able to query the database for available white space channel.

#### 4.2.2. Databases deployed by regulatory body of white space

In this scenario, the master device can directly query the connected database for available white space lists when it is able to connect to the trusted database due to the fact that the databases are deployed by regulatory body of white space and the management of white space is also by regulatory body of white space. Only one credentials are needed, the credentials used to mutual authentication between master device and database can be assigned by trusted authority of regulatory body of white space, e.g. the authority of regulatory body of white space or by another party trusted by the regulatory body of white space. The security model can be showed as followed:

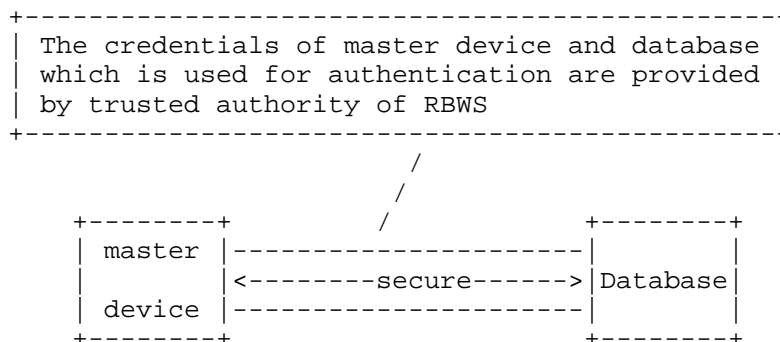


Figure 5: authentication model 3

After the successful mutual authentication, the secure channel shall be established to protect the communication between the master device and the database. The master device and the database shall not engage in any communication prior to the completion of the establishment of the secure channel other than messages for establishing the secure channel.

The TLS protocol depicted in section 4.3 can be considered to establish the secure channel according to the above analysis. The alternative security scheme IPsec can also be used in PAWS. But since TCP and HTTP protocol are recommended to use for PAWS, only the

TLS is introduced in this document.

#### 4.3. TLS protocol

##### 4.3.1. Brief introduction of TLS protocol

TLS protocol provides the communications privacy over the internet which is composed of two layers: the TLS Record Protocol and the TLS Handshake Protocol.

1) The TLS Handshake Protocol is responsible for negotiating a session, which is used to allow peers to agree upon security parameters for the record layer, authenticate themselves, instantiate negotiated security parameters, and report error conditions to each other.

a) The peer's identity can be authenticated using asymmetric, or public key, cryptography (e.g., RSA, DSA, etc). X509 certificate is recommended. When the certificate is used to authenticate the identity of the entity, each party shall verify the other's certificate whether it is valid and has not expired or revoked.

b) According the [RFC5246], RSA or Diffie-Hellman can be used for authentication and key exchange.

c) A `pre_master_secret` is created by key exchange process in TLS handshake protocol, which will be used to generate the `master_secret`. The master secret is required to generate the encryption keys and integrity keys.

2) The TLS Record Protocol takes messages to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, and transmits the result. Received data is decrypted, verified, decompressed, and reassembled, then delivered to higher level clients. Two basic security properties are as follows:

a) Confidentiality: symmetric cryptography is used for data encryption (e.g., AES, etc). The derivation of encryption keys are based on a secret negotiated by the TLS handshake protocol.

b) Integrity: A keyed MAC is used to message integrity check. Secure hash functions (e.g., SHA-1, etc) are used for MAC generated.

##### 4.3.2. Security establishment procedure between master device and database

In related reference of PAWS, TCP and HTTP are recommended to be used

to load the messages in the interface between the master device and the database.

When the TLS protocol is used, all HTTP data between master device and the white space database must be sent as TLS "application data". Normal HTTP behavior should be followed. It means that security association shall be set up by using TLS protocol before establishment of the HTTP connection, and the mutual authentication shall be implemented in TLS protocol.

The following procedures shall be implemented first before the master device contacts to the database using a well-defined access method when it has determined the relevant white space database.

The master device acting as the HTTP client should also act as the TLS client. It should initiate a connection to the white space database on the appropriate port and then sent the TLS ClientHello to begin the TLS handshake.

1) The procedure of TLS handshake protocol can be described as follows which is based on the [RFC5246]GBP[not]it contains four stages:

a) The first stage: security capabilities including protocol version, session ID, cipher suite, compression method, and initial random numbers are established

b) The second stage: certificate of the database, key exchange, and request certificate shall be sent by database

c) The third phase: master device sends certificate if requested. Key exchange and certificate verification may be sent by master device

d) The last phase: change cipher suite and finish handshake protocol

2) Authentication methods

In above establishment procedure of TLS secure channel, Public key certificates or symmetric keys (namely pre-shared keys or PSKs) can be used for the mutual authentication between master device and database. In the PSK case, the shared key needs to be pre-configured in the master device and in the database by manual operation; When the PSK authentication is selected, the certificate and Certificate Request payloads are omitted from the response. The detailed procedure can reference the RFC4279. In the certificate case, the master device and database can obtain an operator certificate through the enrolment procedure.

The use of certificates has advantage that there is a standardized procedure for enrolling the private key corresponding to the certificate while the use of the PSK requires manual operation for establishing the PSK. On the other hand, the use of PSK has advantage that no PKI is required and the procedure after pre-establishment of PSK is simple. When using certificate for mutual authentication, a part of the usual certificate handling is replaced by subscription handling.

### 3) Security protection

For the completion of the TLS handshake protocol, the integrity, confidentiality and replay protected are all activated, all communications between the master device and the database shall be protected by the secure channel. The further authentication of the master device and the white space database should be protected by the secure channel if it needed.

#### 4.3.3. Drawbacks of TLS protocol

Because the TLS runs over TCP, it is susceptible to a number of denial-of-service (DoS) attacks. An attacker who initiates a large number of TCP connections can cause a server to consume large amounts of CPU for doing RSA decryption. Besides, attackers can forge RSTs, thereby terminating connections, or forge partial TLS records, thereby causing the connection to stall. In this situation, implementers or users who are concerned with this class of attack should use IPsec.

## 5. Security Considerations

All contents of this document are dealing with security.

## 6. IANA Considerations

There have been no IANA considerations so far in this document.

## 7. Acknowledgments

Thanks to my colleagues for their sincerely help and comments when drafting this document.



## 8. Normative Reference

- [I-D.ietf-paws-problem-stmt-usecases-rqmts]  
Probasco, S. and B. Patil, "Protocol to Access White Space  
database: PS, use cases and rqmts",  
draft-ietf-paws-problem-stmt-usecases-rqmts-03 (work in  
progress), February 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119, March 1997.

## Authors' Addresses

Yizhuang Wu  
Huawei Technologies  
  
Email: wuyizhuang@huawei.com

Yang Cui  
Huawei Technologies  
  
Email: cuiyang@huawei.com

