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.

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 thisdocument 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".

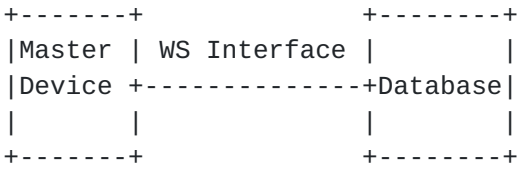

Figure 1: System Model of PAWS

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.

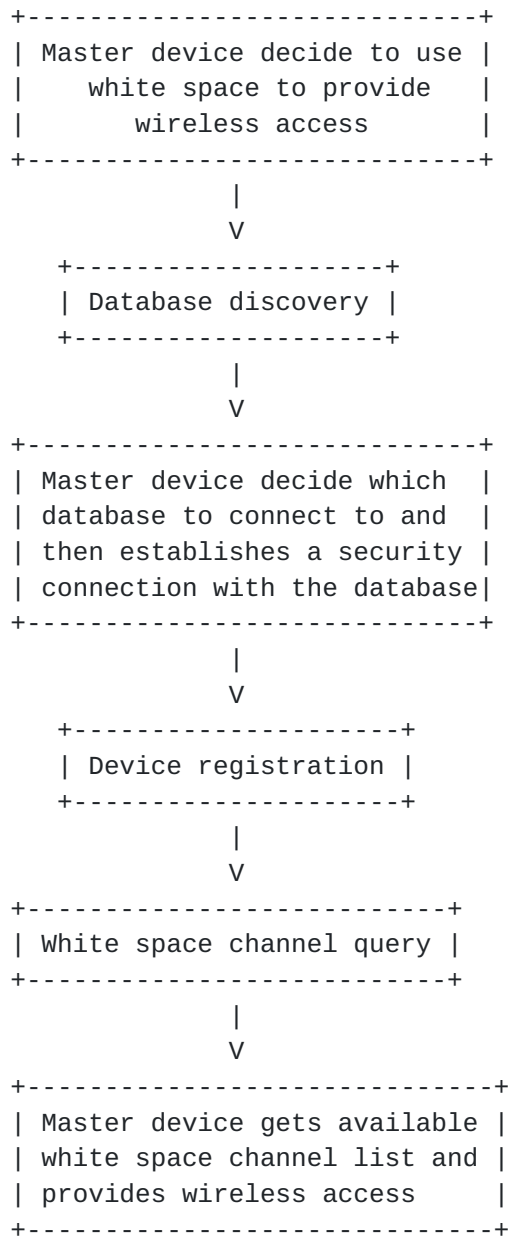

Figure 2: The procedure for getting white space channel for master device

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.

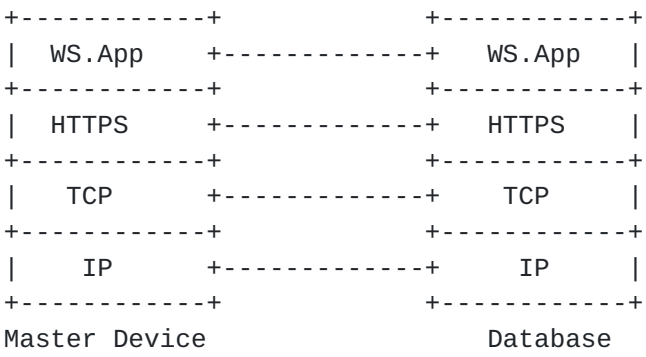

Figure 3: Protocol Stack of PAWS

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.

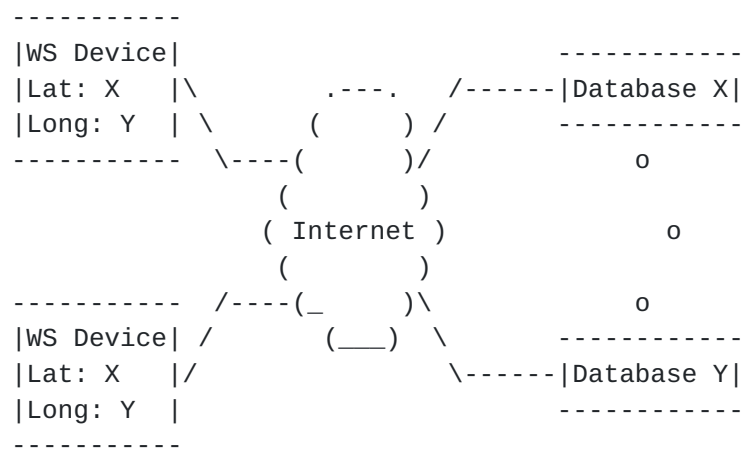

Figure 4: High Level View of the White Space Database Architecture

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.

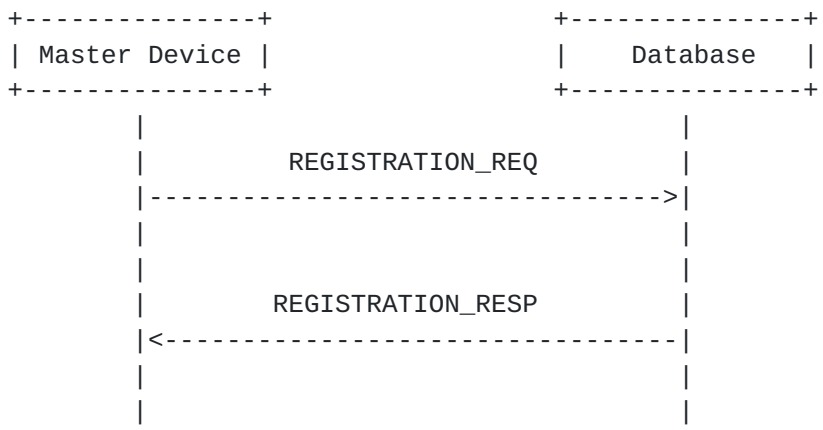

Figure 5: Device Registration with Database Procedure

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 channl 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.

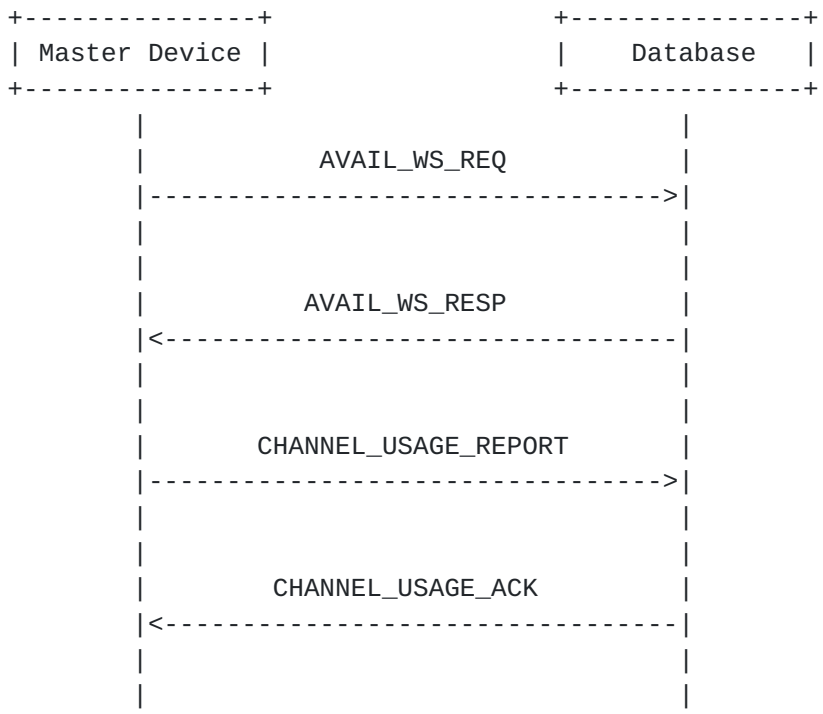

Figure 6: The Available Channel Query Procedure

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.

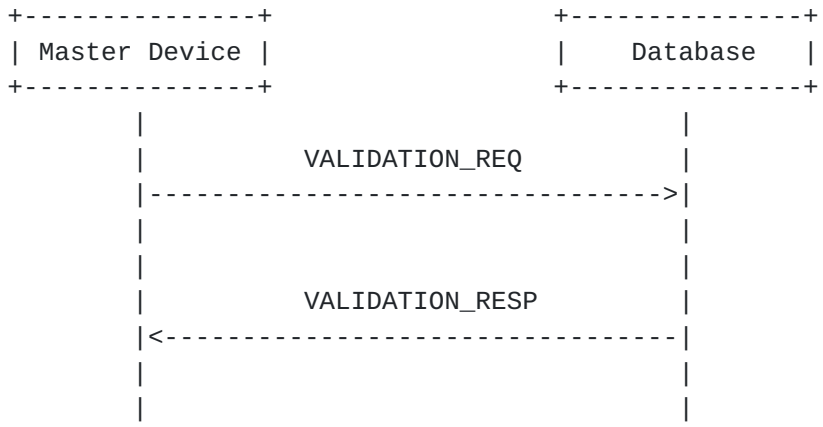

Figure 7: Slave Device Validation with Database Procedure

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 thelave 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 xlmns: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
    uint:
    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:simpelType>

    <!--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