

OpenPAWS: An Open Source PAWS and UHF TV White Space Database Implementation for India

Soumik Ghosh, Gaurang Naik, Animesh Kumar, and Abhay Karandikar
Department of Electrical Engineering
Indian Institute of Technology Bombay, Mumbai 400076, India
soumikghosh1987@gmail.com, {gaurangnaik, animesh, karandi}@ee.iitb.ac.in

Abstract—TV white space (TVWS) geolocation database is being used for the protection of the terrestrial TV broadcast receivers, and the coexistence of secondary devices. To the best of our knowledge, though TV White Space calculations are available, an active online database does not exist for India. In this paper, the development of the first TVWS database for India is detailed and is released for public access. A standardized protocol to access the TVWS database on a readily available hardware platform is implemented. A hardware prototype, which is capable of querying the TVWS database and operating in the TV band without causing harmful interference to the TV receivers in UHF TV bands, is developed. The source code of our implementation has been released under the GNU general public license version 2.0.

I. INTRODUCTION

The subject of interference avoidance to the primary users in *TV white space* (TVWS) has been well studied in the literature. Two mechanisms are generally studied in this respect — spectrum sensing [1], [2] and geolocation database [3]. In the spectrum sensing approach, a secondary user must be capable of sensing the spectrum, in order to determine the presence or absence of a primary user, every time it wants to transmit. On the other hand, in the geolocation database approach, a central TVWS database maintains a list of all the available channels at every location within the regulatory domain of the TVWS database. A regulatory domain is a region (usually a country) in which certain set of rules apply regarding the operation of secondary users and TVWS database in the designated secondary band.

In the rest of this paper, we consider the geolocation database driven approach for interference avoidance to the primary by a TVWS network. In such a network, a central entity maintains a TVWS database, or simply a *database*, which consist of mappings from location coordinates (latitude, longitude) and height to a list of channels and permissible transmit powers levels in each channel. The database is created in accordance with the rules set by the regulator; thus, control over secondary devices operating in TVWS channels is maintained by the regulator. Before transmitting in any TV channel, a secondary user in the network must query the database and obtain a list of available channels with the permissible power levels. The secondary user is allowed to transmit in only one of the available channels with the prescribed limit on the transmit power. In order to standardize this process of database query by the secondary devices, the *Internet Engineering Task Force* (IETF) has released the *Protocol to Access White Space* (PAWS) database [4].

In India, Doordarshan is the sole terrestrial TV service provider operating in the *Very High Frequency* (VHF) and *Ultra High Frequency* (UHF) bands. Information about TV transmitter operation is not available in the public domain and consequently, there is no (TVWS) database for India. In this work, we detail the creation of a database for TVWS operation in India. This database has been created by using the computational model used for TVWS availability estimation in the Indian context [5]. OpenPAWS is presented in this work, which is our open-source implementation of PAWS on an OpenWrt based hardware platform [6] and which will use our database for TVWS operations. A database for primary-secondary and secondary-secondary coexistence with a PAWS interface for communication between the TVWS device and the database is present in the literature [7]. However, to the best of our knowledge, this is the *first* open source implementation of the PAWS protocol and is released under the GNU *General Public License* (GPL) v2.0. Our implementation uses readily-available open-source UNIX command line utilities. The entire implementation is done on the wireless device side using bash scripts and PHP on the database side.

Organization: Architectural details of database assisted operations in TVWS are provided in Sec. II. The details of the PAWS protocol are discussed in Sec. III. The implementation details of OpenPAWS are discussed in Sec. IV. Results of OpenPAWS implementation on the hardware platform and a proposed topology for a pilot test-bed using the database are discussed in Sec. V. Finally, concluding remarks are presented in Sec. VI.

II. ARCHITECTURE FOR DATABASE ASSISTED TVWS OPERATIONS IN INDIA

To explain the database assisted TVWS operation, an upcoming experimental test-bed will be used. This test-bed's background is explained next. Broadband access to the rural population has been a challenge long faced by the Government of India. The Government of India has initiated the *National Optical Fibre Network* (NOFN) in order to provide optical optical connectivity at the Gram Panchayat level. The broadband access enabled at the Gram Panchayat can be extended to the villages, where fiber connectivity is difficult or not possible, through a TVWS based mesh-network. The architecture for such a TVWS network supplementing the NOFN plan using a database is shown in Fig. 1.

A secondary user having internet connectivity to the database can query the database and obtain 'free channel' for operating at a certain maximum power level. In the Indian

context, a village Gram Panchayat connected via the NOFN plan can qualify as this node. In the PAWS terminology (as will be explained in a later section), this device is called a *Master* device. Villages without internet connection cannot access the

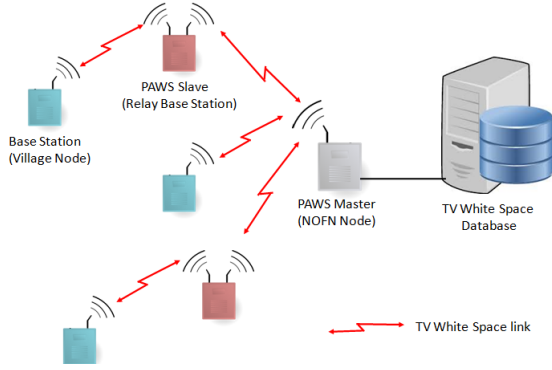


Fig. 1. Architecture for database assisted operations

database directly. To obtain a ‘free channel’, a TVWS node within a village without internet connection, must reach the database through the NOFN node. This TVWS node is termed as a *Slave* device in the PAWS terminology. The Slave device has two wireless interfaces. One of these interfaces is used for communication with the Master device, while the other interface is used to serve client device or Customer Premise Equipment (CPE) and can operate at 2.4GHz so that the end-clients can remain Wi-Fi devices such as smartphones, laptops, and tablets.

The architecture described above allows for spectrum sharing with TVWS nodes acting as secondary service providers. In this approach, the TVWS nodes are susceptible to interference from other secondary devices in their vicinity. This is not particularly favorable to services like fixed backhaul or middle-mile networks described above. An alternate spectrum sharing mechanism termed as *Licensed Shared Access (LSA)* [8] can effectively be used to facilitate rural broadband services using TVWS nodes. The LSA mechanism is a complementary approach to the traditional exclusive licensing and the license-exempt approaches. In the LSA approach, additional users (termed as co-primary users or LSA licensee) are granted license to operate in a frequency band which is assigned (or will be assigned in future) to an incumbent user. The incumbent users agree to the terms of the LSA license, and the license is granted by the regulator.

A certain *Quality of Service (QoS)* is guaranteed to the incumbent users as well as the LSA licensees. The elements of the LSA framework are shown in Fig. 2. The PAWS compliant database is similar to the LSA repository, which interacts with the regulator and the incumbent user (which is the TV broadcaster in this case). The PAWS Master device is analogous to the LSA Controller, which queries the LSA repository in order to prevent any harmful interference to the incumbent user.

Our proposed architecture for a TVWS network

Presently, there are no regulations in India that allow secondary operations in the TVWS. If such operations are allowed

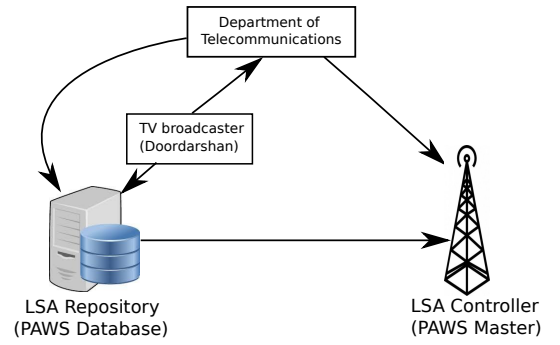


Fig. 2. Elements of the LSA framework, adopted from [8], for Indian scenario

in India, the licensing mechanism used could be a primary-secondary approach or an LSA approach where the UHF TV band devices can be considered as co-primary services operating along with the incumbent users. In either of the two licensing regimes, affordable rural broadband services can be provisioned using TVWS middle-mile links. We have proposed a multi-hop mesh topology for such a network. Fig. 3 shows the topology of a pilot test-bed planned in six villages of the Palghar district of the Maharashtra state. The Khamloli village

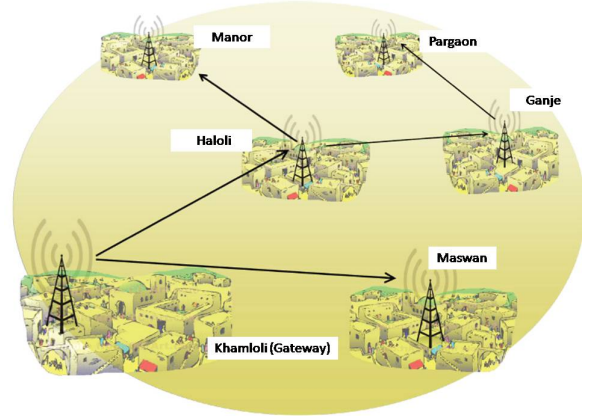


Fig. 3. Topology for a multi-hop mesh network using TV White Space middle-mile links at the pilot test-bed

has access to the Internet through optical fiber. Comparing with Fig. 1, the node stationed at Khamloli village acts as the Master device, while the nodes at Pargaon, Ganje, Haloli, Maswan and Manor villages act as Slave devices. This topology is most relevant in the Indian context, where broadband penetration is low and middle-mile data-transport problems are highly prevalent as many villages are sparsely populated and situated deep inside difficult terrains.

III. OPENPAWS: PROTOCOL DETAILS

PAWS categorizes a White Space device as either a Master or a Slave device. The PAWS protocol defines a Master device as a “device that queries the database, on its own behalf and/or on behalf of a slave device, to obtain available spectrum information.” The Slave device, on the other hand, cannot connect to the database directly. As per the PAWS protocol, a slave device is a “device that queries the database through a master device.” One important assumption made

while developing this protocol is that the Master device and the database, both have connectivity to the Internet.

A. Overview of OpenPAWS protocol

The protocol specifies a sequence of procedures to be followed by the PAWS Master device, PAWS Slave device and the database, which are as outlined below:

- 1) During bootstrapping, the Master device locates or discovers the regulatory domain for its location and the *Uniform Resource Locator* (URL) for the database to send subsequent PAWS messages.
- 2) The Master device then establishes an *Hypertext Transfer Protocol* (HTTP) session with the database.
- 3) An optional initialization message (INIT_REQ) is sent to the database by the Master device.
- 4) The database responds to the initialization message with an initialization response (INIT_RESP).
- 5) Based on the regulatory domain, the Master may have to register itself with the database using the registration request (REGISTRATION_REQ) message.

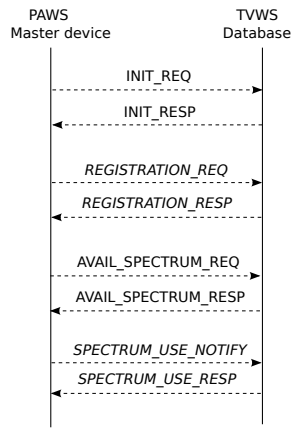


Fig. 4. Messages exchanged between PAWS Master device and the database

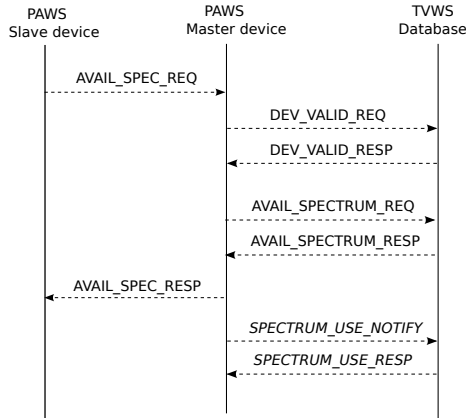


Fig. 5. Messages exchanged between PAWS Slave device, PAWS Master device, and the database

- 6) If the database receives a registration request, it registers the Master device and sends a response (REGISTRATION_RESP) to the Master device.

- 7) Once the initialization and registration procedures are completed, the Master device can query the database for a list of available channels using an available-spectrum request (AVAIL_SPECTRUM_REQ) message to the database.
- 8) The database sends the list of available channels at the Master device's location using the available-spectrum response (AVAIL_SPECTRUM_RESP) message.
- 9) The Master can send the AVAIL_SPECTRUM_REQ message by itself, or for a Slave. If the Master queries the database for a Slave, the Slave device may be validated by the database using the DEV_VALID_REQ and DEV_VALID_RESP messages.
- 10) The Master device can use spectrum use notify (SPECTRUM_USE_NOTIFY) message to inform the database about its decision to operate on a particular channel. The database notes this and sends a spectrum use response (SPECTRUM_USE_RESP). This message is optional as PAWS does not concern the interference between secondary users

The sequence of messages transmitted by the PAWS Master device and PAWS Slave device are shown in Fig. 4 and Fig. 5, respectively. Messages shown in *italic* are optional messages specified by the PAWS protocol. These messages may be required by the regulatory domain of the database and are facilitated in our implementation of the protocol.

B. Format of PAWS messages

All messages exchanged between the PAWS Master device, PAWS Slave device and the database are exchanged using *JavaScript Object Notation* (JSON) encoded strings. JSON strings have multiple parameters encoded in them. E.g., the AVAIL_SPECTRUM_REQ message has the latitude-longitude of the device, the height above average terrain, and the device ID specified. Response from the database contains parameters such as the operating frequencies, and transmit powers encoded. All the message formats are specified by PAWS [4].

The JSON encoded string for the INIT_REQ message used in OpenPAWS implementation is shown below:

```

1 {
2   "jsonrpc": "2.0",
3   "method": "spectrum.paws.init",
4   "params": {
5     "type": "INIT_REQ",
6     "version": "1.0",
7     "deviceDesc": {
8       "serialNumber": "D4:CA:6D:FF:06:15",
9       "rulesetIds": ["IITB-TvWhiteSpace"]},
10    "geo-location": {
11      "point": {
12        "center": {"latitude": 72.90, "
13                  longitude": 18.97}
14      }
15    }
16  }
17 }
  
```

IV. IMPLEMENTATION ARCHITECTURE

In this section, the implementation details of the PAWS Server, and the PAWS Master and Slave devices is discussed.

An embedded Linux based operating system, OpenWrt, was used for the implementation of OpenPAWS. OpenWrt is a configurable open-source operating system that provides extensive functionality. It supports drivers for RF cards from most vendors. OpenWrt was also selected for implementation since a large number of available baseband processor platforms support this operating system. Thus, the source code for OpenPAWS can be readily ported on any hardware platform supporting OpenWrt. The hardware platform used for our implementation is the RouterBoard 433AH baseband processor [10], with RF cards from Doodle Labs (DL535) [11] operating in the frequency band 540-590MHz.

Our implementation of PAWS has two components — one running on the OpenWrt platform and another running on the database. The OpenWrt component of OpenPAWS runs as a user space application, while the database side implementation is done on a desktop server using Apache2 Web Server running PHP scripts. We have released the OpenWrt-side as well as database-side source code under the GNU GPL version 2.0 [9]. Details of OpenPAWS implementation at the database and devices are discussed next.

A. TVWS database for India

In our previous work [5], the quantitative assessment of TV White Space was carried out in four zones of India. Subsequently, Doordarshan has provided data of North Zone and we have extended the analysis of [5] to include North Zone information. We briefly summarize the results of our analysis carried out for all the TV towers operating in India.

Considering the pollution viewpoint (viewpoint of the secondary user), with an allowable interference level (γ) of 15dB, 88.82% of the area in India has all the 15 channels available as white space, while in 99.99% of the area, 12 or more channels are available. On the other hand, if we consider the pollution viewpoint (viewpoint of the primary user) with (Ψ) a fading margin (an additional margin over the Signal to Interference plus Noise ratio that the regulator provides in case of deep fading events) of 1dB, in 78.82% of the area in India, all the 15 channels are available for TVWS secondary operations, and in 99.61% of the area, 10 or more channels are available. Even with the FCC regulations, which is a conservative point of view, 94.97% of the area in India has all 15 channels available, while in 100% of the area 12 or more channels are available. The detailed noise effect is documented in [5].

The data obtained from Doordarshan contains operational parameters (except antenna patterns) of all towers operating the 470 – 590 MHz band in India. This data, along with the computational model described in [5], have been used to create a TV White Space database for India. We discuss the implementation of the PAWS compliant TVWS database next.

PAWS server implementation: The PAWS server is a web-server located at a remote location. This server hosts the White-Space database. This is a MySQL database listing out all possible channels available for transmission corresponding to a particular geolocation. The PAWS master device must have an external connectivity to the PAWS server over a wired or wireless channel. All communications between the PAWS Master, Slave and the database is carried on over the HTTP protocol.

In our implementation, we have used the Apache2 server as the HTTP web-server to host the database. A PHP script runs at the database and keeps listening for incoming PAWS request messages. On receiving the initialization request from Master, PHP script at the server detects parses the request to obtain the coordinates provided by the Master. If the coordinates are not found in the database, the Master device is not within the regulatory domain of the database, and server sends an Error message to the Master and stops communication. If the coordinates are found in the database, the server decides that the point is within its regulatory domain, i.e. in India. The database sends a response message with a ruleset information and records a unique device identifier (MAC address in our implementation) in a repository within the database. Though not the main aim of this paper, the repository keeps record of all the channels being used in a particular location. In case, multiple secondary users start operating in that location, this repository resolves the issue of interference between them. The ruleset information is a guideline with parameterized-rule values which has to be followed by the radio devices while connected to the database. The Master device then queries the channels available at the geolocation. This query may be on behalf of itself or any Slave device connected to the Master. If a match for the coordinates is found, the server looks up for the coordinates in the database and gets the list of available channels with permissible power levels. The server encodes this list in the JSON format and sends it to the Master device through the response message. It must be noted that the PAWS protocol does not specify any particular implementation of the TV White Space database. It only defines the *Application Program Interfaces* (APIs) required at the TV White Space database and the messages exchanged between the PAWS Master device and the database.

B. OpenWrt implementation

1) *OpenWrt Master device:* The PAWS Master is configured to operate as an Access Point (AP) with a single wireless interface. A wired connectivity to the database is provided using an Ethernet cable connected to the Internet backbone. The Ethernet interface of the OpenWrt hardware is bridged to the Wireless interface, which can be configured using the *Command Line Interface* (CLI) of OpenWrt. Detailed configuration procedure for OpenWrt Attitude Adjustment is available in OpenWrt forum. Once the OpenWrt hardware device is configured to operate as an AP in the bridged-local area network mode, the device is ready to be configured as a Master device, which is done using a configuration script *install.sh*. This configuration script checks for all pre-requisites, and installs the required executables for OpenPAWS. All PAWS Master device scripts are set up in the PAWS root directory and a PAWS *User Interface* (UI) is set up. At this stage, the device is configured to operate as a PAWS Master device.

The operational parameters required in order to connect to the database and obtain a list of channels are listed below.

- IP Address of the PAWS Server,
- Coordinates of Master device (latitude, longitude),
- Height of the Master device antenna,
- MAC Address of the Master device.

These parameters can be set up using the PAWS UI as well as the OpenWrt CLI. When all the above parameters are provided and a request is sent to the database, the user space application runs at the Master device. This application initializes and registers the Master device with the database and sends a query to the database. The HTTP connection with the database is established using the bash command ‘cURL’. The JSON encoded string consisting of the above mentioned parameters is sent to the database using an HTTP POST request. This procedure is followed for all messages sent by the Master device to the database. To each of these messages, the database responds with the corresponding response message which decides the action to be taken by PAWS Master device.

The TVWS availability at any location may change over time. The PAWS protocol specifies a parameter ‘max-PollingSecs’ which denotes the maximum interval in seconds after which a secondary device must query the database. In India, TV tower operations do not vary largely over time. Hence, we set this parameter to 86400 seconds, i.e. a secondary device running OpenPAWS must send the AVAIL_SPECTRUM_REQ message to the database once every 86400 seconds. This is done using a PAWS daemon which runs at the Master device and sends the channel request periodically.

The state diagram of the Master device is shown in Fig. 6. Once the Master device obtains the list of available channels

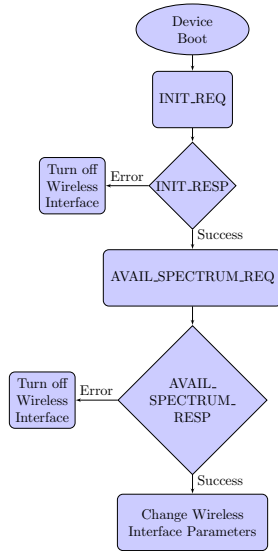


Fig. 6. PAWS Master device state diagram

and corresponding transmission powers, it selects a channel from the list and sets its wireless interface to operate at the channel and transmit at a power lower than the permissible level. In any of the responses from the database, if an Error message is received, the secondary device turns off its wireless interface. The fixed wireless node (Master or a Slave), as shown in Fig. 1 is expected to be set up at a remote location in a rural area. At such remote locations, there is a possibility of power failure. In order to avoid user intervention in the event of a power failure, all PAWS request and response parameters are stored in a local configuration file *config.cfg*. Using this file, the Master device can be set to operate at parameters provided by the database in the event of power failure. All instances of

spectrum query requests are stored in a PAWS update file and can be viewed remotely to check the operational parameters of the Master device as well as Slave devices.

2) *OpenWrt Slave device*: Configuration and setup of the PAWS Slave is done using the configuration script (*install.sh*). In case of the Slave device, the most important requirement is the presence of a PAWS Master device in its radio coverage area. PAWS does not specify the protocol for communication between the Master device and the Slave device. In our implementation, we assume that there are two wireless interfaces available at the PAWS Slave device. One of these interfaces is used for communication with the Master device (Master-side interface), while the second interface is used to serve the CPEs (CPE-side interface). The architecture of the PAWS Slave device described above is designed taking into consideration the Indian conditions as described in Section II. If there is no Master device present in the vicinity of the Slave device, the Slave device turns off its CPE-side wireless interface. The following parameters are required for Slave device operation and can be set using the PAWS UI or the OpenWrt CLI.

- IP Address of the PAWS Master,
- Coordinates of Slave device (latitude, longitude),
- Height of the Slave device antenna,
- MAC Address of the Slave device.

Once the Slave device is connected to the Master, rest of the PAWS messages are exchanged in a manner similar to the Master-database communication. The state diagram of the Slave device is shown in Fig. 7.

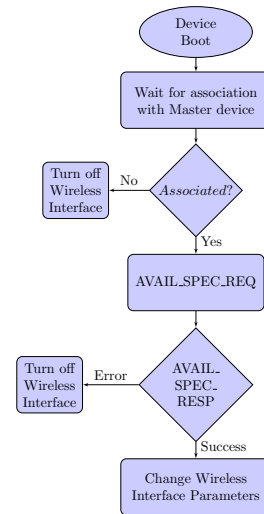


Fig. 7. PAWS Slave device state diagram

V. EXPERIMENTAL RESULTS

To test the working of OpenPAWS, in a laboratory setting, one device was configured to operate as a PAWS Master device, while another device was configured to operate as a PAWS Slave device. The Database was configured on a desktop server running an Apache2 server within the lab. Fig. 8 shows a sample response to an Error message. Such an error message can occur when (i) the hardware device is outside the regulatory domain of the TVWS database, (ii) the

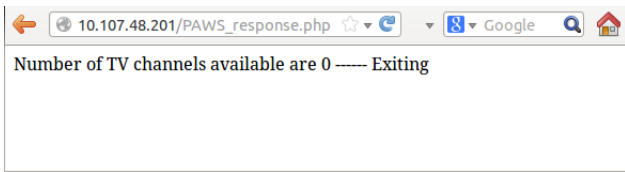


Fig. 8. PAWS Failure Response

hardware device sends an AVAIL_SPECTRUM_REQ before initialization and registration or (iii) there are no channels available at the location of hardware device. In case of Fig. 8, the Error message occurs because the location sent by the hardware device is outside the regulatory domain of the TVWS database. Fig. 9 shows a Success message response. The coordinates provided to the TVWS database are (19.6614, 72.8641) corresponding to the Khamloli village in our pilot test-bed. At this location, all the TV Channels are available.

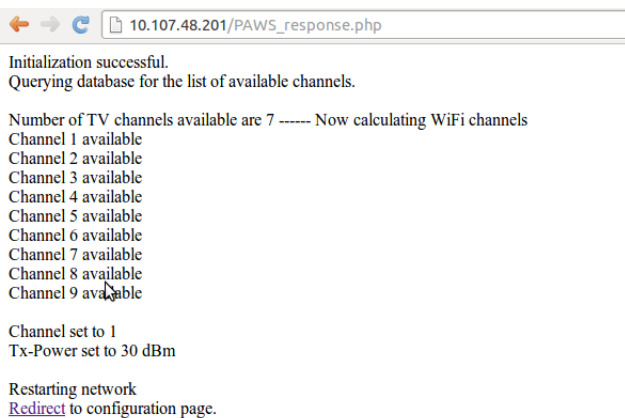


Fig. 9. PAWS Success Response

Table I shows the channels available at the six test-bed sites in the Palghar district. At each of the locations, all the TV Channels are available for secondary user transmission. Consequently, all channels are available for the RF card DL535. OpenPAWS, thus, selects Channel 1 for the operation the devices at each of these locations.

Site Name	Coordinates	TV Band Available	Channel selected by OpenPAWS	Tx Power
Khamloli	(19.661, 72.864)	470-590 MHz	Channel 1	30 dBm
Haloli	(19.673, 72.908)	470-590 MHz	Channel 1	30 dBm
Manor	(19.714, 72.925)	470-590 MHz	Channel 1	30 dBm
Pargaon	(19.596, 72.880)	470-590 MHz	Channel 1	30 dBm
Ganje	(17.781, 73.807)	470-590 MHz	Channel 1	30 dBm
Maswan	(19.709, 72.854)	470-590 MHz	Channel 1	30 dBm

TABLE I. CHANNEL AVAILABILITY PREDICTED BY OUR DATABASE AT THE SIX TEST-BED SITES

Once the channel and power values are set, the OpenWrt network is restarted and the updated channel and power values take effect. The database created is open for public access [12] and can be used to view the list of all TV towers operating in India in the UHF TV band along with all operational parameters. It also displays TV towers operating on any particular channel along with the tower's coverage area calculated in [5].

PAWS server scalability was tested by concurrent database queries from 5 Master devices a hundred times. All the incoming requests were handled seamlessly, and channel list was

given to all the Master devices successfully. The execution time of the PAWS application at the Master device was calculated. The hardware device operating at 680MHz clock took an average of 0.9sec to complete the process of initialization and spectrum querying. Restart of wireless network took an additional 3 seconds on an average. Thus the entire process of querying the database, obtaining and processing the list of channels and setting up the wireless interface takes an average of 3.6 seconds in our trial. The worst results found in this process are as follows: (i) real: 4.10s, (ii) user: 0.95s, and (iii) sys: 0.75s. Real time measurement is higher as it includes other processes and boot up time of the device. User time is time for executing only the initialization and the query process. System time is time for execution of these processes in the kernel only.

VI. CONCLUSIONS

The implementation of a publicly accessible TVWS database for India was described. Protocol to access TVWS database was implemented on readily available hardware platform. Even though TV transmitter operation in the UHF TV band is sparse presently in India, a TV tower database and the protocol are essential to ensure protection to the TV receivers. With our developed database in place, an open-source implementation OpenPAWS was released under the GNU GPL version 2.0. Thus, a complete TVWS system capable of operating with a database and ensuring no harmful interference to the TV receivers is made available for India.

ACKNOWLEDGEMENTS

This research has been supported by the Ford Foundation.

REFERENCES

- [1] H. Chen and W. Gao, "Spectrum Sensing for TV White Space in North America," IEEE Journal on Selected Areas in Communications, vol.29, no. 2, Feb. 2011, pp. 316-326.
- [2] C. Cordeiro, M. Ghosh, D. Cavalcanti, and K. Challapali, "Spectrum Sensing for Dynamic Spectrum Access of TV Bands," Intl. Conf. on Cognitive Radio Oriented Wireless Networks and Comm., Aug. 2007.
- [3] D. Gurney, G. Buchwald, L. Ecklund, S. Kuffner and J. Grosspietsch, "Geo-location Database Techniques for Incumbent Protection in the TV White Space," IEEE Symp. on New Frontiers in Dynamic Spectrum Access Networks, Oct. 2008.
- [4] V. Chen, S. Das, L. Zhu, J. Malyar, P. McCann, "Protocol to Access White-Space (PAWS) Databases, draft-ietf-paws-protocol-19," Sept. 2014. <https://datatracker.ietf.org/doc/draft-ietf-paws-protocol/>
- [5] G. Naik, S. Singhal, A. Kumar and A. Karandikar, "Quantitative Assessment of TV White Space in India," in Proc. of National Conference on Communications, Mar. 2014.
- [6] <https://www.openwrt.org/>
- [7] K. Ishizu, H. Murakami and H. Harada, "TV White Space Database for Coexistence of Primary-Secondary and Secondary-Secondary Systems in Mesh Networking," Intl. Symp. on Wireless Personal Multimedia Communications, Sept. 2012.
- [8] M. Matinmikko, H. Okkonen, M. Palola, S. Yrjola, P. Ahokangas and M. Mustone, "Spectrum Sharing Using Licensed Shared Access: The Concept and Its Workflow for LTE-Advanced Networks," IEEE Wireless Communications Magazine, vol.21, no.2, Apr. 2014, pp. 72-79.
- [9] https://www.ee.iitb.ac.in/tvws/PAWS_sourcecode
- [10] <http://routerboard.com/RB433AH>
- [11] <http://www.doodlelabs.com/products/radio-transceivers/sub-ghz-range/470-790-mhz-tvws-100/>
- [12] <https://www.ee.iitb.ac.in/iwswd>