

# Design and Implementation of an End-to-End Architecture for 3.5 GHz Shared Spectrum

Chang Wook Kim  
University of Washington  
hyperkcw@uw.edu

Jihoon Ryoo  
Stony Brook University  
jiryoo@cs.stonybrook.edu

Milind M. Buddhikot  
Alcatel-Lucent Bell Labs  
milind.buddhikot@alcatel-lucent.com

**Abstract**—The paradigm of *shared spectrum* that allows spectrum bands that are underutilized by primary owners to be exploited opportunistically by secondary devices has been a subject of intense research in recent years. The 2008 FCC rule-making allowing unlicensed use of underutilized DTV white space represented the first step in that direction and elicited similar efforts in other countries. The 2012 PCAST report [5] in the U.S. pushed this new trend further by advocating shared use of 1000 MHz of federal government spectrum with commercial systems and set the goal to release 500 MHz of spectrum by 2022. As a first step towards this goal, the FCC targeted 3550-3700 MHz band used primarily by naval radars and fixed satellite station for release. As of writing of this paper, the FCC has announced the first set of rules for this band to enable deployment of low-powered network technologies like small cells.

The proposed rules require a Spectrum Access System (SAS), akin to a TV Database (TVDB) to implement an innovative three-tiered spectrum management system – dynamic incumbents in the top tier, Priority Access License (PAL) users in the second tier and Generalized Authorized Access (GAA) users in the third tier. Realizing this vision requires a new end-to-end architecture, component protocols and novel radio systems. Our paper represents a step in this direction. Specifically, we present our candidate architecture and its end-to-end implementation that includes a scalable SAS to activate dynamic exclusion zones for incumbent protection, manage primary/secondary devices and dynamically assign spectrum. We present a new protocol – *Protocol for Tiered Access to Shared Spectrum (PTASS)* for transactions between network entities in the architecture. We also describe in detail our prototype end-to-end 3.5 GHz LTE/Wi-Fi testbed. To the best of our knowledge, this is the first paper about the 3.5 GHz shared spectrum access system and the first end-to-end trial of a live 3.5 GHz wireless communications in practice.

## I. INTRODUCTION

Regulatory bodies world-wide are facing increased demand for sub-6 GHz spectrum needed to build future ultra-broadband wireless networks that must keep up with anticipated exponential growth in wireless data traffic. To that end, they are increasingly pursuing policy innovations based on the paradigm of *shared spectrum* that allows spectrum bands that are underutilized by primary owners to be exploited opportunistically by secondary devices. The 2008 FCC rule-making [1]–[3] allowing unlicensed use of underutilized Digital Television (DTV) white space represented the first step in that direction and elicited similar efforts in other countries. It introduced the concept of a geo-location based database called TV Database (TVDB) that performed two key functions: (1)

implement exclusion zones for protection of primary devices such as TV transmitters and wireless microphones and compute available unused TV channels. (2) provide information on unused DTV channels available for unlicensed use to any secondary device that presents its geo-location. Despite extensive research, a few small-scale network trials and FCC approval of six TVDB operators, DTV whitespace has not experienced significant use in commercial wireless networks.

However, the 2012 PCAST report [5] in the U.S. pushed the shared spectrum debate further along by announcing government’s intentions to explore novel technologies and spectrum policies to share 1000 MHz of federal government spectrum with commercial systems. As the first step towards this goal, the FCC targeted release of 3550-3700 MHz band used primarily by high-powered Department of Defense (DoD) radar systems for ground-based, shipboard, and airborne platforms and non-federal Fixed Satellite Service (FSS) earth stations.

As of writing of this paper, FCC has announced the first set of rules for this band, termed Citizens Broadband Radio Service (CBRS), to enable deployment of relatively low-powered network technologies like small cells [4], [6], [7]. Some of the key aspects of this innovative framework are as follows: (1) The rules require a SAS system that is similar to a TVDB to implement an innovative three-tiered spectrum management system (Figure 1) – dynamic incumbents in the top tier, Priority Access License (PAL) users in the second tier and Generalized Authorized Access (GAA) users in the third tier. The incumbents which have the highest priority spectrum access, can require all secondary systems in a spatial region to relinquish all or portions of spectrum at any time. The secondary devices from the PAL tier are allocated exclusive channels that provide guaranteed interference protection, much like traditional long term licenses. The GAA tier provides no interference protection and therefore, allows multiple independent networks to use a given channel not used by incumbent or PAL user. (2) Rules define PAL licenses in terms of 10 MHz spectrum blocks allocated for three years duration over spatial regions as small as a census block. A PAL owner can acquire multiple PAL licenses and stack them together in space and time to create up to 40 MHz of spectrum chunks, valid over a very large spatial region. (3) Out of 150 MHz spectrum in 3550-3700 MHz, 70 MHz is reserved for PAL and remaining 80 MHz is available for GAA devices. FCC also advocates a “use-it-or-share-it” rule that allows GAA

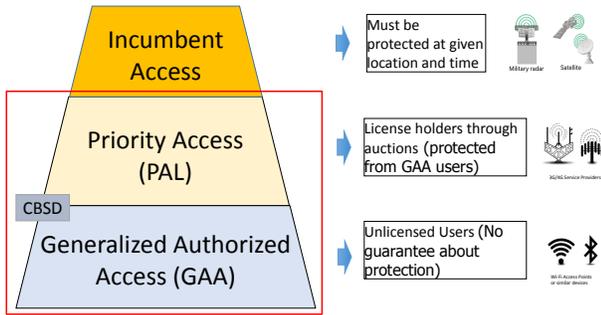


Fig. 1: Tiered spectrum access in 3.5 GHz band.

devices to access entire 150 MHz when PAL devices do not use their allocation. (4) The rules define two types of devices: Category A similar to indoor/outdoor small cells and Category B for point-to-point use. In order to successfully accomplish protection from harmful interference, the FCC ruling currently considers geographical Exclusive Zones for Incumbent Access tier. In case of naval radars, these zones which were initially proposed to be prohibitively large – up to 150+ Km along coast line due to calculations based on macro-cellular deployment [8] have now been shrunk 77% as per more recent the National Telecommunications and Information Administrations (NTIA) calculations [21]. Likewise, non-federal FSS earth stations should be protected with exclusive zones up to 107.4 km against long-term interference and 487 km against short-term interference. These stations operate in receive-only mode from space to earth in 3600-3650 MHz and 37 such stations are deployed in the U.S. On the other hand, to protect PAL users from harmful interference from GAA users, the SAS should determine and assign channels to Citizens Broadband Service Devices (CBSDs) using the information collected from CBSDs such as their geo-location, engineering parameters, and spectrum sensing report. The rules thus require that the SAS should be able to (1) actively assign, reassign, and deactivate channels to secondary users to implement the tiered interruptible access, (2) minimize interference to PAL users from unlicensed GAA users, and (3) minimize aggregate interference from secondary transmissions to incumbent operations.

The FCC rules merely provide a framework for 3.5 GHz shared spectrum ecosystem. Successful implementation of this framework in large scale commercial networks requires significant new research on the design and implementation of an end-to-end architecture, its component protocols, and subsystems such as 3.5 GHz PAL/GAA access points and client devices. In this paper, we report our work in that direction. Our main contributions are as follows:

- We propose a system architecture that consists of two separate logical SASs - a commercial SAS (C-SAS) and a federal SAS (F-SAS) to effectively scale management of incumbent protection and secondary spectrum use.
- We design and implement a new protocol named Protocol for Tired Access to Shared Spectrum (PTASS) for the communication among C-SAS, F-SAS, and CBSDs.

- We design and implement a C-SAS prototype to demonstrate tiered spectrum management. Furthermore, we implement F-SAS and CBSD emulators to evaluate our C-SAS prototype.
- We design and build 3.5 GHz LTE and Wi-Fi Radio Access Network (RAN) components – namely LTE eNB small cells, Wi-Fi Access Points (AP) and corresponding client devices that support 2x2 MIMO. We also integrate the LTE eNB and Wi-Fi APs with spectrum sensors and implement CBSD controller software.

#### A. Outline

The rest of this paper is organized as follows. Section II presents in detail our end-to-end system architecture and details of the PTASS protocol. Section III discusses the design and implementation of software components such as the C-SAS prototype and F-SAS/CBSD emulators. Section IV describes implementation of our hardware prototype and the salient aspects of the eNB, AP and client software. Finally, we discuss future work in Section V and conclude this study in Section VI.

## II. END-TO-END ARCHITECTURE FOR 3.5 GHz SHARED SPECTRUM

Figure 2 illustrates our end-to-end architecture for use of 3.5 GHz spectrum in infrastructure based wireless networks. It consists of three main elements: (1) a Commercial SAS (C-SAS), (2) a Federal SAS (F-SAS) and (3) various Radio Access Networks (RANs) containing CBSDs that can use 3.5 GHz spectrum.

The F-SAS entity handles all the information about the incumbent systems. It collects and provides information about the space, time and frequency over which the incumbent systems are in operation. In case of systems for which location and activity parameters are not classified, the incumbent can explicitly provide to the F-SAS such information in the form of a spatial and temporal *exclusion zone*. For classified, mission-critical systems such as naval radars, for which such information cannot be revealed, alternate means are necessary. Recently, Google, Federated Wireless et al. [24] proposed the concept of a network of Dedicated Listening Devices (DLDs) or spectrum sensors aimed at observing and estimating primary activity. The current FCC rules [4] support a general concept similar to DLD called Environment Sensing Capability (ESC). Such a ESC (DLD) network can supply observations to the F-SAS which can then estimate the extent of spatio-temporal exclusion zone and propagate it to the C-SAS.

The C-SAS implements two primary functions to support tiered access:

- (1) *CBSD channel management*: assign mutually exclusive channels to PAL and GAA users at a given location and time when incumbent is not using the spectrum outside of an exclusion zone. Inefficient sub-optimal channel allocation to CBSDs can result in poor secondary network capacity and ultimately, impair business case for use of the 3.5 GHz band.

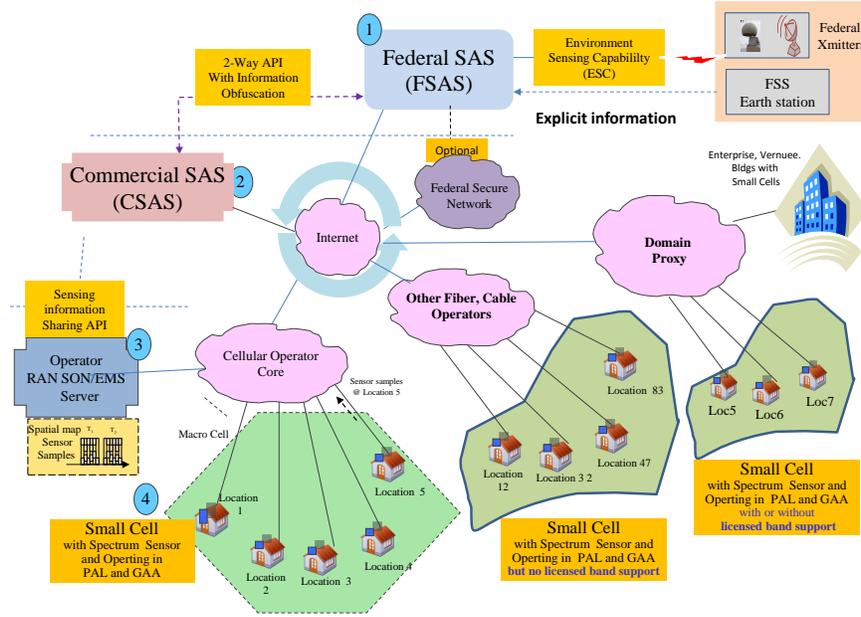


Fig. 2: Overall system architecture and component entities.

To that end, such channel assignment can meet two sets of goals:

*Maximum GAA capacity:* As per the PCAST report [5] and FCC 3.5 GHz NPRM [6], GAA devices by definition cannot expect interference protection and must accept interference from other co-channel transmitters. This means, in a given region, multiple GAA devices belonging to different or same entity (owner) can be assigned the same channel. The C-SAS can actively manage interference environment for a given GAA channel by admission control to maximize wireless capacity for all GAA devices that use it.

*Maximum PAL protection and capacity:* The C-SAS can actively assign channels to PAL owners to offer best interference protection. A PAL device using channel  $M$  may experience adjacent channel interference ( $M - 1$ ,  $M + 1$  or for that matter  $M - k$ ,  $M + k$  for  $k > 0$ ) caused by Incumbent Tier-1 systems and Tier-3 GAA devices. Therefore, C-SAS can reassign GAA devices to an empty or in-use GAA channel to guarantee exclusive access to PAL devices if a new PAL device requests a channel that is being opportunistically exploited by GAA devices.

(2) *Incumbent protection:* dynamically monitor activation of an incumbent exclusion zone and inform, interrupt and reassign all CBSDs inside exclusion zone that are impacted and need to vacate the channel. Optionally, collect measurements and employ spatial data processing techniques to improve dynamic exclusion zone estimates.

Since, the C-SAS manages and monitors all PAL/GAA CBSDs that need access to 3.5 GHz spectrum, it requires that such devices register their geo-location, device type (PAL, GAA or Both), and RF-front end specification such as maximum transmit power, antenna height and gain, receive filter

characteristics, etc. In addition, the need of incumbent and PAL protection requires the C-SAS to actively track device population in PAL and GAA tiers and channel(s) that each device is operating. Therefore, the CBSD devices must report the list of channels it will operate in and the C-SAS must maintain a state record such as *CBSD Device ID*, *IP address*, *location*, *next time epoch of expected contact and channels used* in its database to track at all times. This *back reporting* is an essential aspect of 3.5 GHz due to need for near instant interruption of secondary devices in the event the primary system turns on. Such back reporting is not required in TV databases as the model of primary reclaiming spectrum is in the form of sparse activity of microphones that make a priori reservations in the database.

#### A. CBSD Channel management in C-SAS

Given  $N$  10 MHz channels and  $k$  PAL licenses sold in a region  $R$ , the C-SAS tackles two distinct channel management problems:

- *PAL Channel Management:* (1) If channels are statically allocated to PAL owners, up to  $k$  channels are used for meeting demand of  $k$  licenses. The owners of the licenses configure these channels in their network using their preferred technology (e.g.: LTE). (2) If channels are dynamically allocated to PAL owners, the C-SAS solves the problem of how to allocate  $k$  out of  $N$  channels.
- *GAA channel Assignment:* After PAL channel assignment of  $k$  channels, the C-SAS assigns remaining  $(N - k)$  channels to the GAA devices. We envision two models for how C-SAS can manage these channels to variable degree: (1) Open Loop and (2) Closed Loop

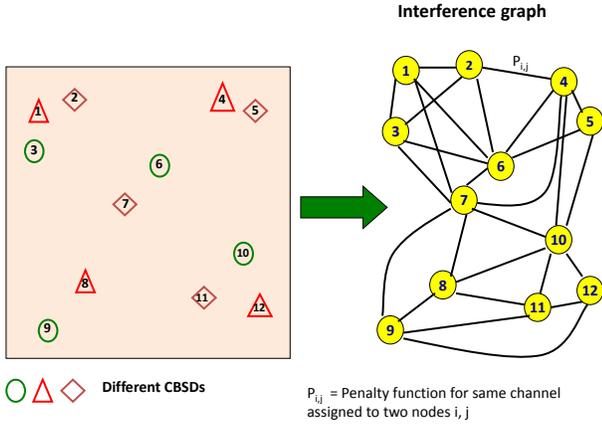


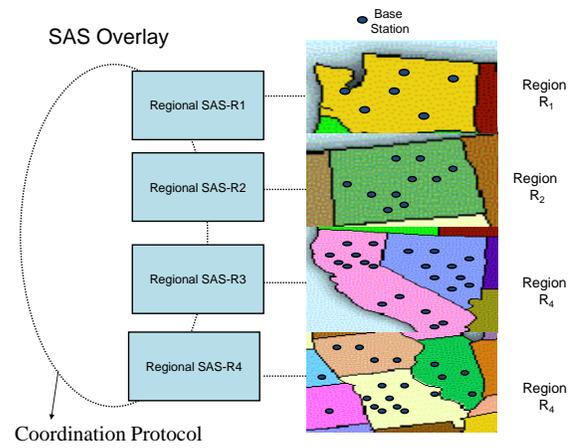
Fig. 3: Interference graph for channel allocation.

1) *Open Loop GAA Channel Management*: In this unmanaged mode of channel management, the SAS does not dictate which of the available channels a GAA device can use. Two variants are possible:

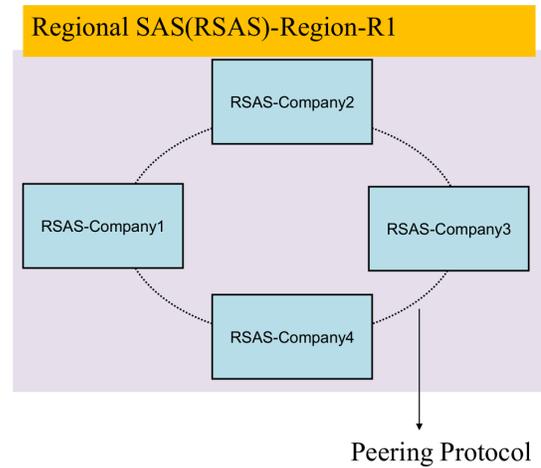
*Basic Open Loop* Here upon receiving a GAA device channel request at location  $L$ , the SAS looks up available  $(N - k)$  channels at the location and provides the entire list. The device selects one or more channels from this list based on its local measurements and other criteria (e.g: traffic needs, time-of-day, energy constraints etc.) and notifies those channels to SAS.

*Enhanced Open Loop* Here C-SAS plays "an advisory role" to improve channel management and interference by providing hints on what channels a GAA device can use to experience better interference environment and better performance. Specifically, for each channel in the available  $(N - k)$  channel, the C-SAS provides an estimate of observed activity level – based on how many CBSD devices (APs) are currently using the channel, the amount of traffic load, observed received signal strength indicator (RSSI), etc. at the requesting location  $L$ . Using these estimates, the C-SAS creates a ranked list in the descending order of quality. Here, the highest ranked channel is the best channel. The GAA device picks one or more of these best-ranked channels and performs mandatory back report of the list of in use channels. Note that it is entirely optional for GAA devices to exploit the hints. Therefore, the operation is considered open loop mode as the C-SAS has no control on how CBSD picks a GAA channel despite its help. It works best when the GAA devices are cooperative devices. Also, in this mode, both data collection and processing complexity are increased to support the "dynamic channel hints".

2) *Closed Loop GAA Channel Management*: In this mode, the C-SAS plays an active role of a channel manager (and even an enforcer) and assigns channels to the GAA tier CBSDs. The channels assigned by the C-SAS are mandatory in that the CBSDs must use only the assigned channel and nothing else. Much the same as in the enhanced open loop option, the C-SAS creates a ranked order list in the descending order



(a) Spatial organization of SAS



(b) Business model driven peered SAS

Fig. 4: Multiple SAS server organization.

of quality and uses it to solve a channel allocation problem. Specifically, the C-SAS creates a model of interference relations, such as an interference graph (Figure 3), using physical interference model (based on the RF propagation model, transmitter power, physical distance between transmitters, terrain, etc.) [10]–[12] between all GAA CBSDs in the region and solves the channel allocation problem. Note that the same channel can be allocated to neighboring or nearby GAA devices and their interference relationship can be captured using a penalty function  $P_{i,j}$  in Figure 3. A non-zero value of  $P_{i,j}$  indicates amount of co-channel interference between two interfering nodes when they are assigned the same channel. If the two nodes are assigned different channels, then  $P_{i,j}$  captures adjacent channel interference.

Note that closed loop channel model can also be applied to dynamically assign channels to base stations of multiple PAL license holders in a region. In this case, exclusive nature of PALs requires that the base stations of different operators in the region must be assigned different channels. Such inter-

provider conflict is modeled in the interference graph by inserting an interference edge between any two graph nodes that correspond to base station of different operators. The penalty function for such edges is set to  $P_{i,j} = \infty$  to signify that the two nodes cannot ever be given the same channel.

Our architecture envisions three types of CBSD networks (Figure 2): (1) A traditional cellular operator deploying small cells that combine traditional licensed bands with 3.5 GHz PAL channels and optionally 3.5 GHz GAA channels. Here the small cell CBSDs report to the operator Self Organizing Network (SON) server or Element Management Server (EMS) which serves as a CBSD proxy and manages all interaction with the C-SAS. (2) A cable Wi-Fi operator or emerging fiber-to-home operators such as Google, which can combine 3.5 GHz PAL/GAA channels with traditional Wi-Fi access points. Here as well, the operator may deploy a domain proxy (DP) server to aggregate the signaling to the C-SAS. (3) A new class of indoor operator networks deployed in large enterprises, venues such as high rise buildings, train stations, airports, etc. which can combine 3.5 GHz PAL/GAA channels to offer capacity to in-building users. Here a domain proxy server is able to aggregate the signaling traffic to the C-SAS.

### B. Organizing Multiple SAS servers

The F-SAS and C-SAS are abstract and logical entities and their practical implementation must account for requirements of scalability and competitive business models. To that end, we envision that the 3.5 GHz spectrum over entire USA can be managed by a large collection of Regional SAS server domains – each designated to manage spectrum in a large subset of the region such as multiple cities or states. Figure 4(a) illustrates an example of four regional SAS server domains independently managing spectrum in four sub-regions  $R_1, R_2, R_3, R_4$ . These server domains need to coordinate spectrum assignments in the areas where two or more regions border [12].

Further, the FCC FNPRM [4] policy mandate envisions SAS be operated as a competitive business and as such, each regional SAS domain may consist of SAS servers from multiple companies. In the example in Figure 4(b), four different SAS servers operated by four different companies – RSAS-Company 1 to RSAS-Company 4 collectively manage 3.5 GHz spectrum in Region R1 and CBSDs from this region can connect to any of the four servers to access basic spectrum access service. Each SAS may provide additional value-added service for a price. However, the SASes must peer with each other using a peering protocol and guarantee consistent and correct channel allocation.

Note that each regional SAS company may implement its SAS server using cloud and network function virtualization (NFV) paradigms. This means each regional SAS implementation may span multiple machines in one or more data centers.

With this prelude on the architecture and organization of multi-SAS ecosystem, in the following, we present our new signaling protocol called PTASS.

### C. PTASS Protocol

PTASS, which stands for *Protocol for Tiered Access to Shared Spectrum*, provides the necessary messaging substrate between the F-SAS, C-SAS, and CBSD. The protocol is inspired from the IETF standard Protocol to Access White Space (PAWS) designed in the context of DTV whitespace [13]. It supports simple, intuitive and efficient HTTP-based REQUEST/RESPONSE message exchange mechanism using JavaScript Object Notation (JSON) as data object [25].

PTASS uses HTTPS transaction to guarantee secure message exchange, specifically HTTP REQ/RESP with HTTP post. We also support two communication modes for PTASS message exchange: asynchronous and synchronous transactions. In the asynchronous mode, the recipient, either C-SAS or CBSD, immediately sends an acknowledgment on successful reception of a PTASS\_REQ message through a HTTP response and ends the HTTP connection. After processing the received request, new HTTP connection is initiated in the opposite direction of the request in order to deliver a PTASS\_RESP message. The synchronous transaction is used only when the recipient needs to handle error(s) in the request message. This allows the implementation to minimize the number of active, alive TCP connections – a critical resource especially on the C-SAS side. On the other hand, in the synchronous mode, the PTASS\_RESP message is sent over the HTTP response after holding the HTTP connection open during the request process.

The full version of protocol is under review in a multi-stakeholder standardization group on spectrum sharing<sup>1</sup> and here we present only key features.

A typical sequence of PTASS operations, illustrated in Figure 5(a), is as follows:

- *Designated SAS Discovery (INIT REQ/RESP)* A CBSD sends an initialization message including device credentials (e.g.: operator identity), geo-location and other information to a well-known *Portal C-SAS*. The portal C-SAS maintains the latest database for the serving C-SAS. Once the C-SAS receives an initialization message, it responds with information for a corresponding C-SAS serving the CBSD geo-location.
- *Device Registration (REGISTRATION REQ/RESP)* The CBSD sends a registration message to the designated serving C-SAS in order to prove its rights to access the spectrum. The successful registration-response message to a verified CBSD contains a *Ticket* – issued per network interface and valid only for certain time period during which it can be used for further queries to obtain and activate available spectrum. The default value in our implementation is set it to 24 hours but it can be changed to any amount of time to meet the C-SAS operation policy. The *Ticket* mechanism provides enhanced security and anonymity.

<sup>1</sup>A full technical report [9] with detailed message and packet formats of the latest version of protocol is available upon request.

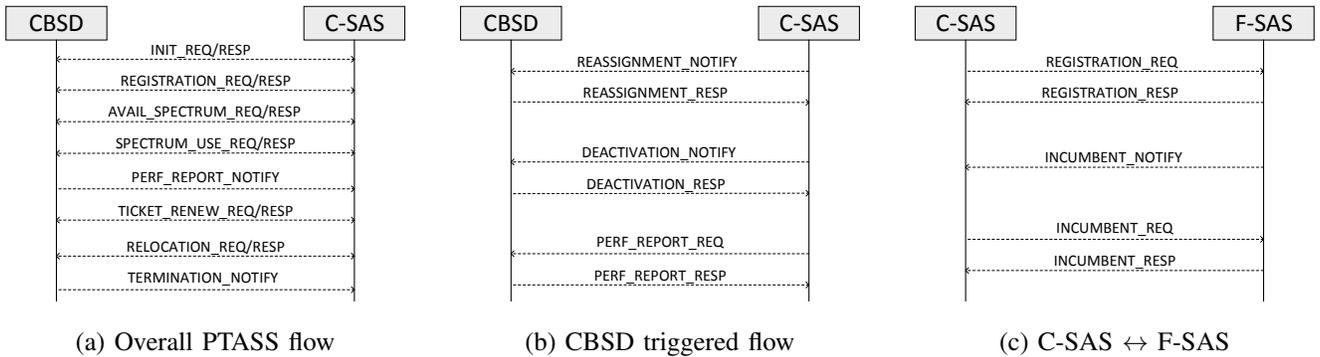


Fig. 5: PTASS message flow.

- *Available Spectrum Request* (*AVAIL SPECTRUM REQ/RESP*) The CBSD sends an available spectrum request message to the C-SAS, the response to which contains one or more available channels depending on the channel assignment model (described in Section II-A).
- *Spectrum Use Request* (*Spectrum USE REQ/RESP*) The CBSD sends a spectrum-usage request message to the C-SAS in order to inform its intention to operate within the assigned spectrum. The response message confirms its operation.
- *Performance Report* (*PERF REPORT NOTIFY*) The CBSD periodically sends a performance report to the C-SAS which can utilize it for performance improvement by optimizing channel assignment. The report can contain metrics for (1) observed performance in the in-use channels such as number of retransmissions, average throughput, end-to-end delay, throughput and delay variance and (2) spectral energy scan observed in adjacent channels and optionally in the entire 150 MHz of CBRS band.
- *Ticket Renew Request* (*TICKET RENEW REQ/RESP*) The CBSD sends a ticket renewal request message to extend the validity of previously assigned ticket. The renewal depends on if the C-SAS allows the extension or terminates the ticket due to various reasons (such as scheduled incumbent activity, or channel re-assignment).
- *Relocation Request* (*RELOCATION REQ/RESP*) The CBSD sends a relocation request message to inform that its location has changed more than the required geographical threshold - specifically if location change exceeds 50 meters horizontal and 3 meters elevation from the last reported location [4], and checks if the current C-SAS can still serve at the new location. The C-SAS checks if the updated location is still in its spatial and business jurisdiction and the previously assigned channel is valid at the new location.
- *Termination Notify* (*TERMINATION NOTIFY*) The CBSD sends a termination notification to the C-SAS when it stops the operation in the assigned spectrum. The C-SAS updates its database to free the spectrum resource for assignment to other CBSDs.

In the asynchronous mode, our proposed communication

mechanism assumes that both CBSD and C-SAS support the `http server-service`, which enables the C-SAS to send a notification message to the CBSD in order to reassign spectrum or deactivate its operation for performance optimization and higher-tiered access protection. Due to the active request from the C-SAS, it is able to immediately assign new spectrum to the CBSD, protect higher-tiered users from the harmful interference caused by lower/same tiered users. Moreover, it helps performance optimization by requesting additional performance report by the C-SAS. Figure 5 (b) shows message exchange triggered by the C-SAS.

Figure 5 (c) shows message exchange between C-SAS and F-SAS. Every C-SAS must be registered to the F-SAS (via `REGISTRATION_REQ/RESP`) in order to query incumbent information by providing its authority. In the procedure of incumbent request, the C-SAS can configure the schedule of incumbent access activity by using the information in the `INCUMBENT_RESP` message. Therefore, the C-SAS must reassign or deactivate their CBSDs to protect the incumbent users. When the F-SAS needs to inform new incumbent schedule or update the existing activity in the 3.5 GHz spectrum, the F-SAS shall notify the C-SAS with the updated schedule via `INCUMBENT_NOTIFY` message.

### III. DESIGN OF C-SAS SERVER

Figure 6 depicts the functional components and their interactions in our C-SAS software. Our implementation relies on multi-threading to allow component modules to perform their roles independently and simultaneously. We also use thread-safe blocking queue for data transfer between modules. The modules that process a large number of tasks internally maintain a thread pool to quickly invoke each task as a single thread. We implement our C-SAS prototype using Java platform and its libraries such as concurrent package and HTTP library and MySQL for the database system.

1) *Communication Manager*: The main function of the communication manager is to establish HTTPS connections and communicate with CBSDs, domain proxies (DPs) and F-SAS in order to exchange information. It consists of HTTPS Server Manager, HTTPS Client Manager, Input Message Queue, and Output Message Queue. The HTTPS Server Manager opens the HTTPS socket and waits for requests

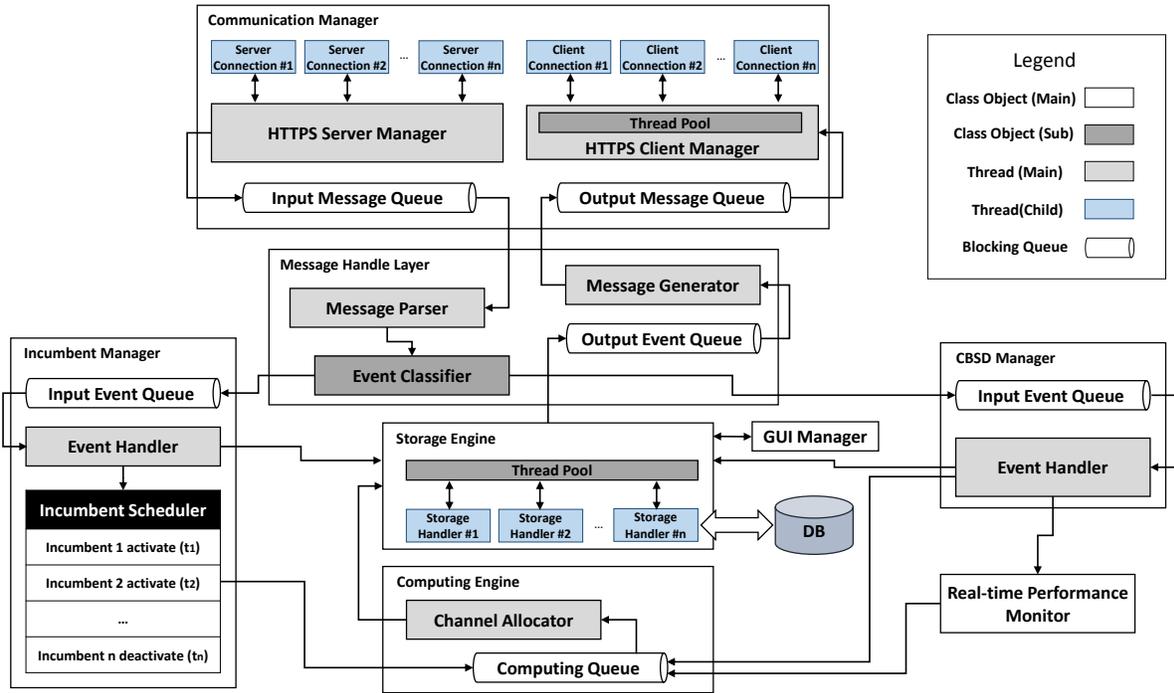


Fig. 6: Our C-SAS architecture: main components and their relationships

from CBSDs. It receives and accepts the HTTP request and enqueues the received PTASS message in the Input Message Queue. The HTTPS Client Manager takes PTASS message from the Output Message Queue and sends it to corresponding client.

2) *Message Handle Layer*: The Message Handle Layer including Message Parser and Message Generator is responsible for encoding/decoding of PTASS messages. First, The Message Parser decodes a PTASS message and converts it into an event based on its message type by taking a PTASS message from the Input Message Queue. Then, the generated event is passed to the Event Classifier which correctly puts it into Input Event Queue of Incumbent or CBSD Manager. Also, the Message Generator takes an event from the Output Event Queue in order to encode it to a PTASS message and adds the message to the Output Message Queue in the Communication Manager.

3) *Incumbent Manager*: The Incumbent Manager is dedicated to processing incumbent events from F-SAS. It analyzes and process the events from the Input Event Queue and passes the incumbent information to Storage Engine which saves or updates in the database and creates tasks for handling incumbent activation/deactivation. The tasks are added into the Incumbent Scheduler that can schedule commands in temporal order to execute periodically. The scheduler puts incumbent information into Computing Queue in order to reassign channel to CBSDs inside exclusive zone of incumbent.

4) *CBSD Manager*: The CBSD Manager handles the events enqueued in its event queue by CBSDs or DPs. The event manager sequentially processes the events and demultiplexes it

to appropriate function block. For example, it passes the device registration event to the Storage Engine for the database transaction. For the the spectrum query event, it enqueues it into Computing Queue to assign channel to the CBSD. It sends the performance report event to Real-time Performance Monitor block to account for observed metrics. The CBSD manager can also perform relevant checks before demultiplexing the event – e.g. it can verify CBSD’s authorization to use 3.5 GHz spectrum or validity of its PAL license before sending the event to Channel Allocator.

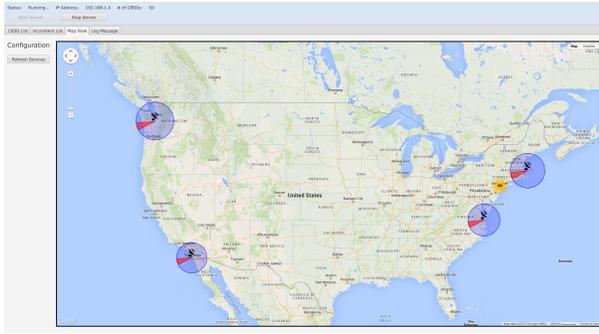
5) *Storage Engine*: The Storage Engine deals with all database transactions such as query, insert, and update. It provides query-related APIs to other modules such as a query for CBSD information and engineering parameters. In case of data manipulation events such as device registration and channel assignment, the Storage Engine allocates a thread to a single event, binding an instance of Storage Handler class. This allows concurrent processing of multiple time-consuming events using event-based asynchronous pattern. The Storage Engine takes the result of completed task and generates output event after the Storage Handler finishes its database transaction. Finally, the generated event is added to the Output Event Queue in the Message Handler Layer.

6) *Computing Engine*: This module implements algorithms that compute assignment of channels in 3550-3700 MHz to PAL and GAA CBSDs as outlined in Section II-A.

7) *Real-time Performance Monitor*: The main purpose of this block is to extract statistical trends in data on performance observed by CBSDs and detect performance degradation. It sends a request for channel reassignment into the Computing

CBSD ID	Interface ID	Operator	Auth. Status	Latitude	Longitude	Tx Power (dBm)	Max. Height (m)	Assigned Spectrum (MHz)	Band Plan	Assigned Power (dBm)	Service ID	MTCPS Info.	Current Status	
1	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
2	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
3	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
4	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
5	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
6	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
7	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
8	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
9	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
10	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
11	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
12	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
13	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
14	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
15	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
16	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
17	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
18	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
19	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
20	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
21	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
22	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
23	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
24	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
25	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
26	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
27	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
28	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
29	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE
30	1	At&T	PLN	40.712000	-73.972000	30.0	10.0	3000.0	3000.0	10.0	00	0000000000	100.000.0.0	ACTIVE

(a) CBSD list view shows the list of CBSDs and their parameters



(b) Map view shows geographical location of exclusive zones of incumbents and CBSDs

Fig. 7: GUI screenshot of the C-SAS prototype.

Queue when poor performance is detected. Currently the performance report includes parameters such as throughput, the number of retransmissions, packet delay and Packet Error Rate (PER).

8) *GUI Manager*: Screenshots of our C-SAS prototype are shown in Figure 7. The GUI Manager provides a graphical frontend for the real-time information on current status and parameters of CBSDs and Incumbent users. It offers *list* views of CBSD/incumbent and a map view to help visualize current location of CBSDs and exclusive zone of incumbent users on the geographical map.

A. F-SAS & CBSD Emulators

We designed and implemented the F-SAS and CBSD emulators using Java and MySQL platform to help evaluate the C-SAS prototype.

1) *F-SAS Emulator*: The important function of the F-SAS emulator is to create a virtual incumbent event by defining its exclusive zone as a (1) set of geographical coordinates  $(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$ , (2) operating frequency range and (3) total time of operation. The emulator also generates the `INCUMBENT_NOTIFY` message and sends it to the authorized C-SAS in order to inform of the newly scheduled activity or update to an ongoing activity of the incumbent. It also answers with relevant incumbent information through `INCUMBENT_RESP` when it receives `INCUMBENT_REQ` from the C-SAS.

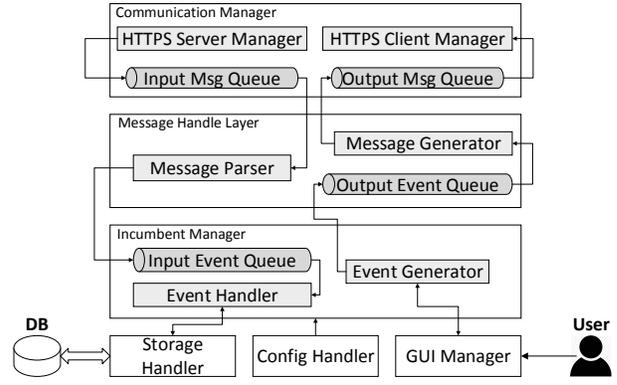


Fig. 8: F-SAS emulator: main components

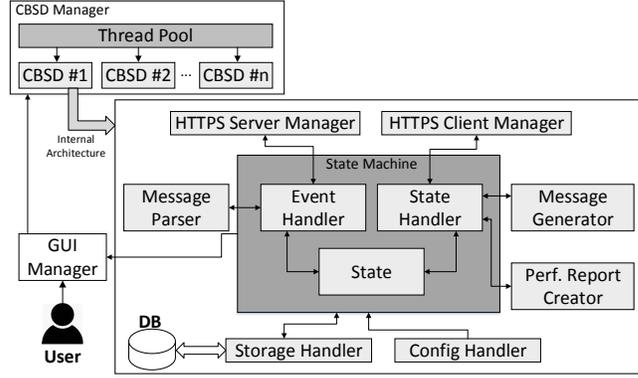


Fig. 9: CBSD emulator: main components

The FSAS emulator implementation (Figure 8) leverages components of the C-SAS prototype. The Communication Manager includes HTTPS Server Manager, HTTPS Client Manager, and two Message Queues. The HTTPS Server Manager receives a request from C-SAS and puts into the Input Message Queue. Also, the HTTPS Client Manager connects to the C-SAS and sends `INCUMBENT_NOTIFY` or `INCUMBENT_RESP` by taking messages from the Output Message Queue in order. Message Handle Layer performs the same role with the one in the C-SAS prototype with Message parser/Generator and Output Event Queue. Event Handler in Incumbent Manager takes an incumbent query event from its Input Event Queue and processes it by querying to Storage Handler which associates with the incumbent database. This results in an incumbent response event which is added to the Output Event Queue.

The F-SAS emulator can be operated through its GUI interface. The GUI Manager module updates the GUI and passes the new incumbent information to Event Generator in the Incumbent Manager when a virtual incumbent is created or updated by the user. After that, the Event Generator creates an incumbent notification event that generates `INCUMBENT_NOTIFY` message delivered to the C-SAS prototype and puts it in the Output Event Queue.

2) *CBSD Emulator*: The CBSD emulator module (shown in Figure 9) implements concurrent operation of numerous virtual CBSDs that independently connect to C-SAS, exchange

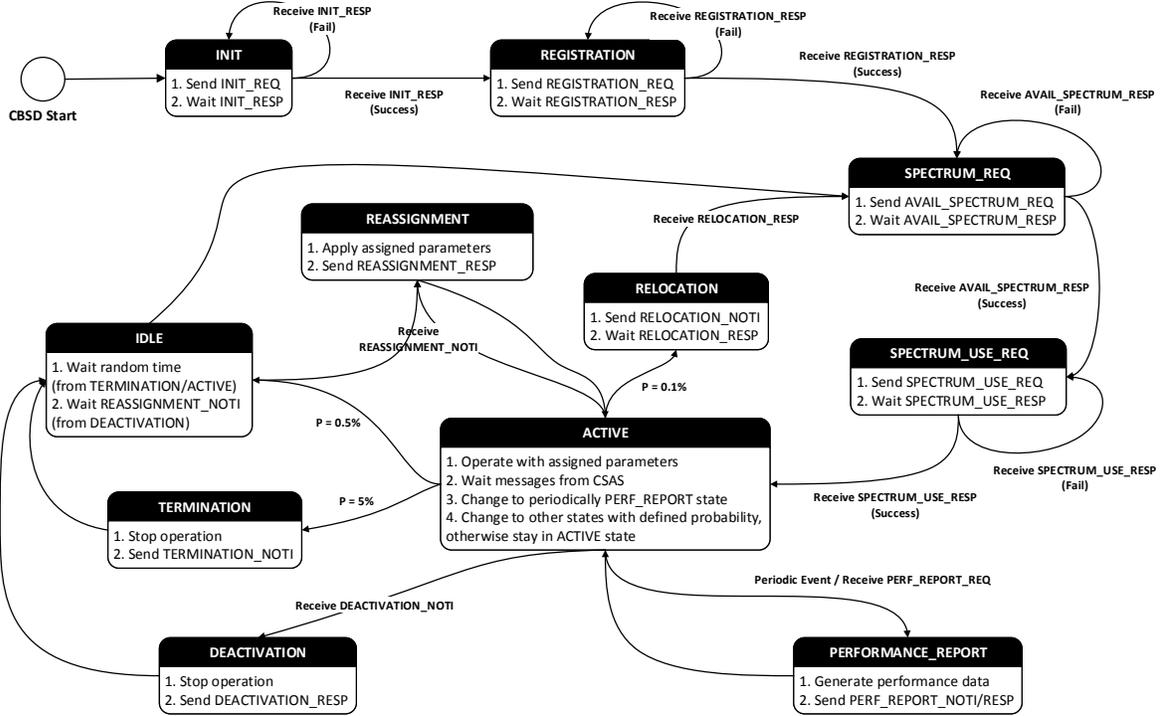


Fig. 10: The state transition diagram in our CBSD emulator.

PTASS messages and perform actions like a real CBSD. A user can create virtual CBSDs and configure their parameters such as operator-id, location, authorization type, transmit power, antenna height through GUI Manager. The CBSD Manager runs a number of threads corresponding to PAL or GAA class instances inherited from the CBSD class in a thread pool when the emulator starts. Therefore, each thread acts like a single CBSD node with *State Machine* in the class instance by automatically generating and sending PTASS messages to the C-SAS prototype, and updating its state. Furthermore, virtual CBSDs are able to receive PTASS messages from the C-SAS through their HTTPS server and update current parameters to the local CBSD database.

The State Machine internally runs an infinite loop and executes a corresponding method in State Handler for the current state. The State Handler encodes an appropriate PTASS message through Message Generator and passes the message to HTTPS Client Manager when its method is called. Also, a PTASS message is passed to Event handler when HTTPS Server Manager receives it from C-SAS prototype. The Event Handler decodes the message through Message Parser and updates to a new state and also, notifies updates in the parsed information to the database in case of receiving available spectrum response or reassignment/deactivation query message. The State module in the State Machine maintains current state information and provides it to Event Handler and State Handler. All database transactions such as query and update are handled by Storage Handler. Finally, in ACTIVE state,

it periodically changes to PERFORMANCE\_REPORT state in order to create a performance report including performance parameters and spectrum sensing data and randomly with user-defined probability to other states such as TERMINATION, IDLE, and RELOCATION states in order to emulate a realistic CBSD. Figure 10 describes state transition in the CBSD Emulator.

In PERFORMANCE\_REPORT state, the State Handler creates a performance report notification in Performance Report Creator and passes it to the HTTPS Client Manager like other PTASS messages to be delivered to the C-SAS. To generate practical spectrum sensing data, we implement several propagation models which can possibly be utilized to the signal strength calculation in the 3.5 GHz spectrum. The models presented in our work are empirical models and estimate mean path loss with various parameters such as distance between base station and receiver, antenna height, frequency, terrain type and etc. For higher frequency band and cellular networks, various extensions of Hata model [14] are mostly used. Therefore, we implement four propagation models to give choices for the emulation: Stanford University Interim (SUI) [15], COST-231 Hata [16], ECC-33 [17], and Ericsson model [18].

In our emulation, we assume all virtual CBSDs are capable of performing spectrum sensing. The Performance Report Creator in the virtual CBSD node calculates RSSI values  $P_{R_i}(f)$  from neighboring CBSD nodes  $i = 1, 2, \dots, n$  which operate in the same frequency  $f$ . First, we calculate the  $P_{R_i}(f)$

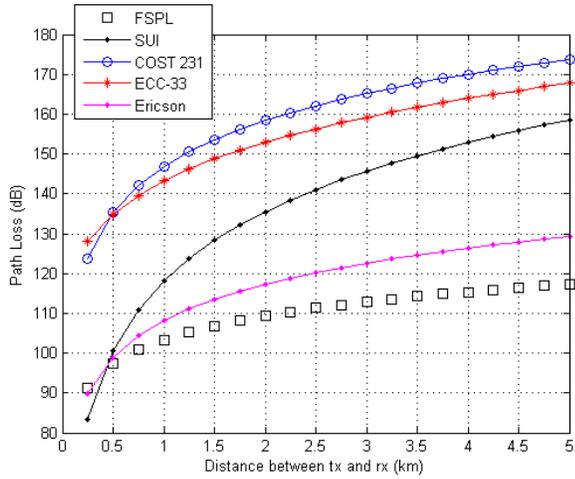


Fig. 11: Path loss in urban/large city environment at 10 m transmitter and receiver antenna height and 3.5 GHz operating frequency

as

$$P_{R_i}(f) = P_{T_i} + G_{T_i} + G_R - PL(f) \quad (1)$$

where  $P_{T_i}$  and  $G_{T_i}$  are transmit power (in dBm) and antenna gain (in dB) for any neighboring CBSD  $i$  respectively,  $G_R$  (in dB) is the receiver antenna gain, and  $PL(f)$  (in dB) is the path loss predicted from propagation models.

Therefore, the sensed RSSI  $P(f)$  (in mW) is defined as:

$$P(f) = \sum_{i=1}^n 10^{P_{R_i}(f)/10} + P_N(W) \quad (2)$$

where  $P_N(W)$  is the noise power which is the sum of thermal noise over bandwidth  $W$  and Additive White Gaussian Noise (AWGN) with normal distribution. The frequency resolution for spectrum sensing in the Performance Report Creator can be defined as a configuration parameter of the emulator. We exclude own transmit signal strength of the sensing CBSD because the spectrum sensing is usually performed during the *quiet period* in which CBSD mutes its transmissions.

Figure 11 shows path loss predictions using four implemented models and additional Free Space Path Loss (FSPL) model. To produce these, we set the frequency to 3.5 GHz, both transmitter and receiver antenna heights to 10 m, and the terrain type to urban/large city. Also, the path loss formulas and other parameters for propagation models can be found in [19], [20]. The significant difference among empirical models suggest needs for an accurate model using real measurement data. Such a model can better predict coverage of secondary devices and resulting interference and will help enhance the performance of the channel assignment in the C-SAS.

#### IV. IMPLEMENTATION OF 3.5 GHz SHARED SPECTRUM TESTBED

Figure 12 illustrates our end-to-end 3.5 GHz LTE/Wi-Fi testbed. It consists of three main components: (1) scalable

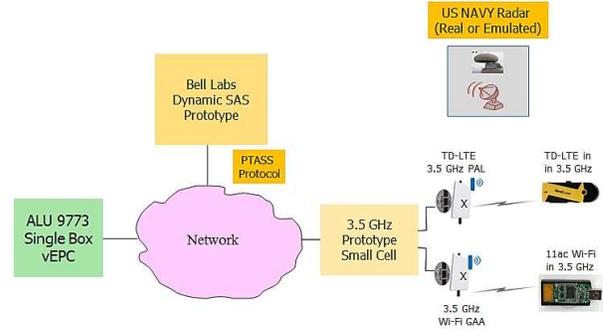


Fig. 12: 3.5 GHz End-to-end LTE/Wi-Fi testbed

SAS infrastructure consisting of C-SAS, F-SAS emulator and CBSD emulator. (2) Hardware prototypes of 3.5 GHz LTE eNB and 3.5 GHz GAA Wi-Fi Access Point (AP), and (3) 3.5 GHz LTE and Wi-Fi user equipment (UE) devices. We implemented two applications – (a) CBSD Controller to manage eNB, AP and (b) Client Controller to manage UE components of our prototype. The AP and eNB operate as CBSD by associating and registering with the C-SAS. The hardware platform of the CBSD consists of a host computer, a spectrum sensor, a IEEE 802.11n compatible dual band 2x2 MIMO Wi-Fi card with Qualcomm Atheros AR95xx chipset and a 2x2 MIMO frequency transponder that translates 5.2/5.4/5.8 GHz frequencies to 3.5 GHz. We use standard `hostapd` – a 802.11n Wi-Fi router functionality and `dnsmasq` – with DHCP server functionality available in Ubuntu 14.04. Our configuration supports the HT40 mode using 40 MHz wide bandwidth, 400 ns short guard interval, space-time block code (STBC) and MIMO with two dedicated antennas. The 5.8 GHz to 3.5 GHz MIMO transponders are connected to the host computer via USB to provide the low latency control capability that can reliably tune the operating frequency to any 3.5 GHz channel. The antenna port of each transponder is connected to directional Wi-Fi antennas supporting the 3.5 GHz band.

The 3.5 GHz LTE eNB consists of a 2.5 GHz LTE small cell, a LTE-in-Box complete EPC solution and a 2x2 MIMO transponder to convert 2.5 GHz to 3.5 GHz frequencies and vice versa.

Lastly, we implement the spectrum sensing function, required for the performance report, using a ThinkRF WSA4000 spectrum sensor [22] capable of operating in the 100 KHz -10 GHz wide frequency range and offering an instantaneous scan bandwidth of 125 MHz. This device is compatible with the PyRF Python library [23] which provides classes and APIs for device configuration such as configuration of scanning frequency range, reading I/Q samples, Fast Fourier transform (FFT) computation, and calibration. This capability helps our

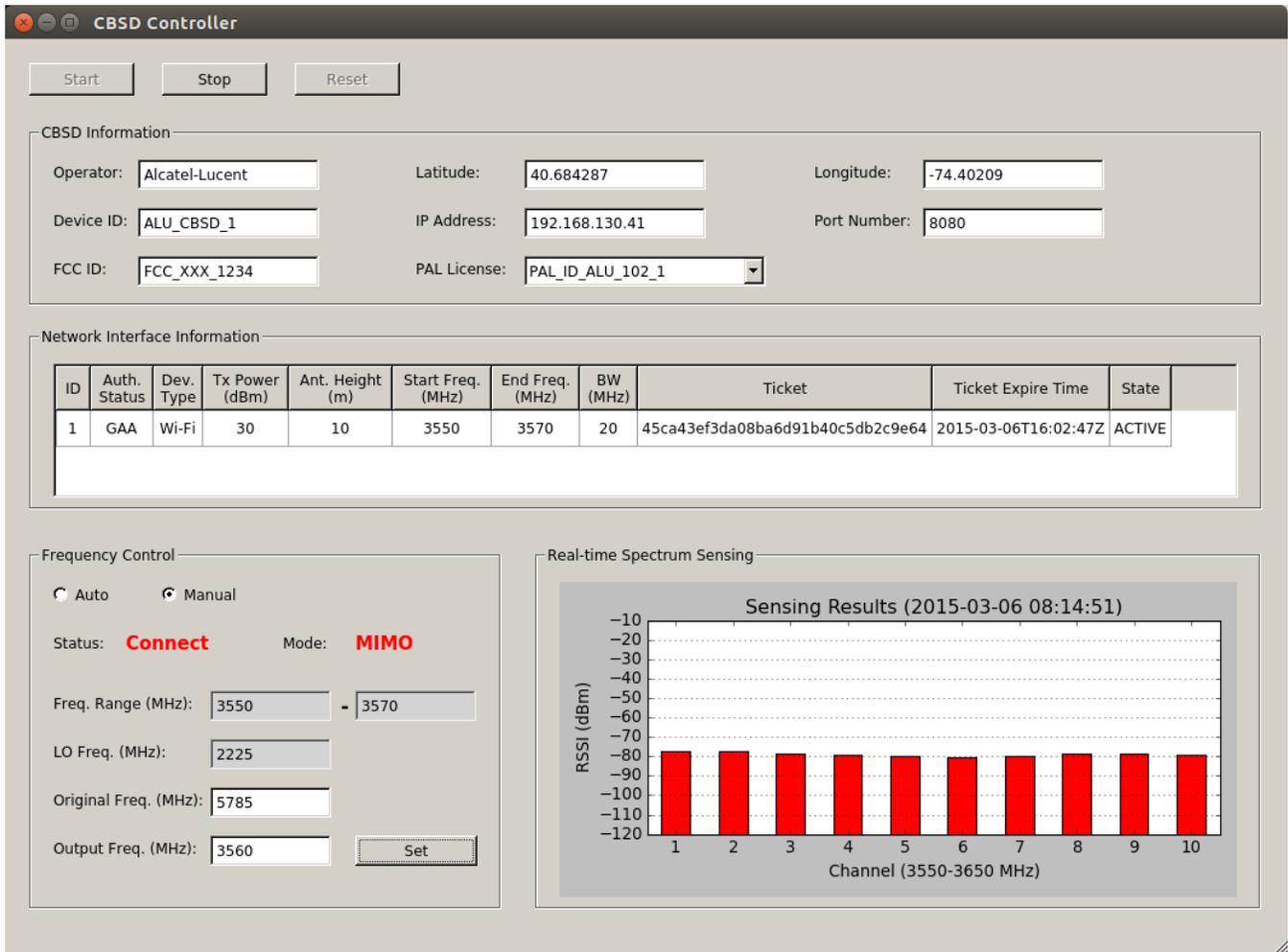


Fig. 13: The GUI screenshot of CBSD Controller.

prototype to obtain sophisticated sensing data across entire 3.5 GHz band.

The GUI screenshot of the CBSD Controller implemented in Python is shown in Fig 13. This software consists of four modules, each independently running as a thread: *CBSD Manager*, *Frequency Controller*, *Channel Notifier*, and *Spectrum Sensing*. The CBSD Manager connects with the C-SAS via HTTPS in order to send/receive PTASS messages. It is registered with the C-SAS as a CBSD and assigned a 3.5 GHz channel to operate its wireless networks. The assigned parameters such as frequency range and the SAS issued ticket are shown in network interface information section of the GUI. We implemented the Frequency Controller which associates with the transponders in order to read/write current operating frequency via USB command. It adjusts the transponders to the assigned frequency after the CBSD Manager receives the SPECTRUM\_USE\_RESP message from the C-SAS. Before this operation, the Channel Notifier sends the assigned channel information to the currently connected clients. To achieve this, the Channel Notifier opens a TCP socket and waits for client connection. Then the socket accepts the client and maintains the connection to notify whenever the channel is

newly assigned. Finally, the *Spectrum Sensing* module controls the spectrum sensor and periodically performs spectrum sensing within the 3.5 GHz shared spectrum. The generated signal strength data is averaged for each 10 MHz channel and displayed on the GUI. Also, it saves the sensing data into the local noSQL database and sends a performance report by summarizing the stored data every user-defined interval or explicit request from the C-SAS.

We built LTE/Wi-Fi UE hardware platform using the same components and technologies used for LTE eNB and Wi-Fi AP. Also, the Client Controller has the Frequency Controller to set frequency of transponders via USB. In addition to this, it shows parameters of the connected AP such as SSID, frequency, bit rates, and signal level and receives newly assigned 3.5 GHz channel from the AP through TCP connection between Frequency Receiver and Frequency Notifier in the CBSD Controller.

We have successfully operated our end-to-end testbed to demonstrate 105 Mbps 3.5 GHz 11n MIMO Wi-Fi throughput and 35 Mbps SISO LTE throughput under the control of the C-SAS prototype.

## V. FUTURE WORK

For the C-SAS, we plan to study (a) channel allocation algorithms for optimizing performance of CBSDs and channel assignment time and (b) fast algorithms for detecting CBSD performance degradation. We plan to investigate a cloud-based system to further scale the C-SAS and support platform independence. We also plan to investigate how our localized sensing measurements can be used in near-real time to create Radio Environment Map (REM) that accounts for incumbent as well as secondary user activity. We will also leverage CBSD measurements to develop better propagation model for 3.5 GHz wireless networks.

Last but not the least, we intend work with newly launched US Government effort on spectrum sharing test-bed, specifically the National Advanced Spectrum and Communications Test Network (NASCTN) to seek opportunities to use our test-bed for realistic LTE/Wi-Fi co-existence testing in presence of live incumbent radars.

## VI. CONCLUSIONS

In this paper, we presented a novel system architecture for 3.5 GHz shared spectrum ecosystem. We discussed in detail the component elements, namely the C-SAS, the F-SAS, and the Wi-Fi and LTE radio access network infrastructure. We presented the design of a new signaling protocol called PTASS envisioned to provide messaging substrate in our architecture. We have implemented the C-SAS prototype and the F-SAS/CBSD emulators to demonstrate the PTASS transactions. We also presented details of our live 3.5 GHz Wi-Fi/LTE test-bed and its component applications. In summary, our paper provides much needed proof of feasibility of incorporating shared spectrum under the control of a spectrum management server in the future wireless networks.

## REFERENCES

- [1] FCC, *In the Matter of Unlicensed Operation in the TV Broadcast Bands: Second Report and Order and Memorandum Opinion and Order*, FCC Document 08-260, Nov. 2008.
- [2] FCC, *In the Matter of Unlicensed Operation in the TV Broadcast Bands: Second memorandum opinion and order*, FCC Document 10-174, Sep. 2010.
- [3] FCC, *In the Matter of Unlicensed Operation in the TV Broadcast Bands: Third memorandum opinion and order*, FCC Document 12-36, April 2012.
- [4] Amendment of the Commissions Rules with Regard to Commercial Operations in the 3550-3650 MHz Band, Report and Order and Second Further Notice of Proposed Rulemaking, Federal Communications Commission (FCC) Released: April 21, 2015.
- [5] President's Council of Advisors on Science and Technology, *Realizing the Full Potential of Govt-held Spectrum to Spur Economic Growth*, July 2012.
- [6] FCC, *In the Matter of Amendment of the Commission's Rules with Regard to Commercial Operations in the 3550-3650 MHz Band*, FCC Document 12-148, Dec. 2012.
- [7] FCC, *In the Matter of Amendment of the Commission's Rules with Regard to Commercial Operations in the 3550-3650 MHz Band*, FCC Document 14-49, April 2014.
- [8] U.S. Department of Commerce, *An Assessment of the Near-Term Viability of Accommodating Wireless Broadband Systems in the 1675-1710 MHz, 1755-1780 MHz, 3500-3650 MHz, and 4200-4220 MHz, 4380-4400 MHz Bands*, Oct. 2010.

- [9] J. Ryoo, Kim, C., and Buddhikot, M. "Protocol for Tiered Access to Shared Spectrum (PTASS), Version 1", Bell Labs Technical Report, Oct 2014 and April 2015.
- [10] A. Subramanian, H. Gupta, S. Das and M. Buddhikot, *Fast Spectrum Allocation in Coordinated Dynamic Spectrum Access Based Cellular Networks*, Proceedings of IEEE DySPAN 2007, Dublin, Ireland, April 17-21, 2007.
- [11] A. P. Subramanian, M. Al-Ayyoub, H. Gupta, S. Das, M. Buddhikot, *Near Optimal Dynamic Spectrum Allocation in Cellular Networks*, Proceedings IEEE DySPAN 2008, Chicago, USA, October 14-17, 2008.
- [12] M. Buddhikot and K. Ryan, *Spectrum Management in Coordinated Dynamic Spectrum Access Based Cellular Networks*, Proceedings of the IEEE DySPAN 2005, Nov 8-11, 2005 (US PATENT: US8958810 and continuations).
- [13] The IETF Draft in Standard Track, V. Chen, *et al.*, <https://tools.ietf.org/html/draft-ietf-paws-protocol-20>.
- [14] M. Hata, *Empirical Formula for Propagation Loss in Land Mobile Radio Services*, IEEE Transactions on Vehicular Technology, vol. VT-29, pp. 317-325, Sep. 1981.
- [15] V. Erceg, K. V. S. Hari, *et al.*, *Channel Models for Fixed Wireless Applications*, tech. rep., IEEE 802.16 Broadband Wireless Access Working Group, Jan. 2001.
- [16] COST Action 231., *Digital Mobile Radio towards Future Generation Systems, final report*, tech. rep. European Communities, EUR 18957, 1999.
- [17] Electronic Communication Committee (ECC) within the European Conference of Postal and Telecommunications Administration (CEPT), *The Analysis of the Coexistence of FWA Cells in the 3.4 - 3.8 GHz Band*, tech. rep., ECC Report 33, May 2003.
- [18] *Cellular Expert Technical Reference*, pp 32-34.
- [19] V. S. Abhayawardhana, I. J. Wassell, D. Crosby, M. P. Sellers and M. G. Brown, *Comparison of Empirical Propagation Path Loss Models for Fixed Wireless Access Systems*, 61th IEEE Vehicular Technology Conference, pp. 73-77, May 2005.
- [20] J. Milanovic, S. Rimac-Drlje and K. Bejuk, *Comparison of Propagation Model Accuracy for WiMAX on 3.5 GHz*, 14th IEEE International Conference on Electronics, Circuits and Systems, pp. 111-114, Dec. 2007.
- [21] -, *3.5 GHz Exclusion Zone Analyses and Methodology*, NTIA Technical Report TR-15-517, June 18, 2015.
- [22] -, *ThinkRF, Inc.*, <http://www.thinkrf.com>.
- [23] -, *PyRF Library*, <http://www.pyrf.org/en/latest>.
- [24] -, <http://apps.fcc.gov/ecfs/document/view?id=60001039128>, March 2015.
- [25] -, <http://json.org>