Sense*Less*: A Database-Driven White Spaces Network

Rohan Murty
Harvard University
rohan@eecs.harvard.edu

Ranveer Chandra
Microsoft Research
ranveer@microsoft.com

Thomas Moscibroda
Microsoft Research
moscitho@microsoft.com

Paramvir Bahl
Microsoft Research
bahl@microsoft.com

Abstract—The most recent FCC ruling proposes relying on a database of incumbents as the primary means of determining white space availability at any white spaces device (WSD). While the ruling provides broad guidelines for the database, the specifics of its design, features, implementation, and use are yet to be determined. Furthermore, architecting a network where all WSDs rely on the database raises several systems and networking challenges that have remained unexplored. Also, the ruling treats the database only as a storehouse for incumbents. We believe that the mandated use of the database has an additional opportunity: a means to dynamically manage the RF spectrum. Motivated by this opportunity, in this paper we present *SenseLess*, a database driven white spaces network. As suggested by its very name, in SenseLess, WSDs obviate the need to sense the spectrum by relying entirely on a database service to determine white spaces availability. The service, using a combination of an up-to-date database of incumbents, sophisticated signal propagation modeling, and an efficient content dissemination mechanism ensures efficient, scalable, and safe white space network operation. We build, deploy, and evaluate SenseLess and compare our results to ground truth spectrum measurements. We present the unique system design considerations that arise due to operating over the white spaces. We also evaluate its efficiency and scalability. To the best of our knowledge, this is the first paper that identifies and examines the systems and networking challenges that arise from operating a white space network, which is solely dependent on a channel occupancy database.

I. Introduction

The recent FCC ruling [3] has obviated mandatory spectrum sensing in white spaces networks. Instead, the ruling requires white space devices (WSDs) to learn of spectrum availability at their respective locations from a central database of incumbents. Broadly, the database is required to house an up-to-date repository of incumbents including television stations and in certain cases, wireless microphones, and use this information to determine white spaces availability at a WSD's location.

However, the systems and networking challenges that arise when all WSDs have to rely on the database have yet remained unexplored. In this paper we present the design and implementation of SenseLess. As suggested by the very name, in SenseLess, all WSDs obviate the need for sensing by reling on a geo-location service to determine white spaces availability.

In this paper we consider SenseLess in the context of an infrastructure network that consists of two kinds of WSDs: base stations (BSs) and client devices that associate with the BSs. We also discuss possible extensions to other scenarios.

SenseLess is a *service* that (1) contains an up-to-date database of incumbents *and* WSDs in active operation, (2) predicts the availability of white spaces at any WSD's location using sophisticated propagation modeling (including high-resolution terrain-data as well as TV-tower-specific parameters, such as antenna-height, etc.), and (3) provides a framework to dynamically manage all WSD devices connected to the network. Hence, SenseLess encompasses the database functionality as proposed by the FCC ruling and extends it further.

Although the basic approach appears to be simple, developing a comprehensive solution in which devices rely on the geo-location service to determine white spaces availability is non-trivial. This is because our design introduces some unique challenges that must be overcome.

A key issue in a network such as SenseLess is that it must continue to afford the *same* protection to incumbents as spectrum sensing would. This challenge coupled with all WSDs having to rely on a database to discover white spaces is a significant departure from conventional network designs. Hence, architecting such a network raises the following challenges:

TV detection: Relying solely on a geo-location service to detect TVs may result in false positives causing a loss in white spaces, and false negatives resulting in interference with TVs. The extent of these errors depends on the accuracy of RF propagation modeling, which, to the best of our knowledge, has not been quantified in the context of white space availability.

Client Bootstrapping: This is a chicken-and-egg problem. Due to spatial variation [12] in the white spaces availability, when a new client appears, the base station (BS) needs to know the client's location to infer the available channels at the client. However, to avoid interference with primaries, the client cannot transmit its location to the BS *unless* it knows the available white spaces at its own location.

Location Resolution: Although the FCC requires devices to know their location before accessing the geo-location service, constantly determining location is a power intensive error-prone operation [19]. Hence, we ask, is knowing the location of the client crucial for correct operation? If so, at what resolution is this information needed?

We address the above challenges as follows. (1) To detect incumbents, we use a geo-location service that uses propagation modeling with terrain elevation data. We validate the white spaces predicted by this service with spectrum measurements collected by a 1500 mile drive in two states. (2) To solve

the bootstrapping problem, we intelligently code the white space availability in a BS's beacon packets, and finally (3) through detailed measurements, we evaluate the impact of client location information on our network.

From a system's perspective, the design of the Sense-Less system is to a large extent governed by constraints that are unique to white spaces. We show that any white space system in which, instead of sensing, devices rely on a third-party entity to learn about white space availability must follow certain design principles. Specifically, we show that in order to achieve sufficiently accurate (i.e., not overly conservative) prediction of white space availability, sophisticated and computationally-intensive models must be used. This result is derived by comparing our model results with actual measurements of UHF spectrum across Washington state. Also, to avoid wasting significant amounts of available white spaces, accurate client locations must be taken into account when determining the free channels at a base station. Finally, mobility can only be handled efficiently if the service disseminates spectrum availability information to each client at very low-latency. We design and implement SenseLess following these design principles.

This paper makes the following contributions:

- We demonstrate the practicality of using propagation models to accurately predict spectrum availability by comparing our model results to measured ground truth data.

- We derive the characteristic systems challenges that arise from removing sensing from client devices, and quantify their impact on white space utilization.

- We design and implement a scalable system that can efficiently enable networking over white spaces without relying on sensing.

- We provide extensive evaluation of our system in terms of scalability, efficiency, and safety properties.

To the best of our knowledge, this is the first paper that identifies and examines the systems and networking challenges that arise from operating a white space network, which is solely dependent on a channel occupancy database.

The rest of the paper is organized as follows. We make the case for general design principles of a SenseLess-like approach in Section II, present the SenseLess design in Section III, implementation details in Section V, evaluation in Section VI, and a discussion of related issues in Section VII.

II. DESIGN PRINCIPLES

A white spaces network in which WSDs rely on a database to learn of white space availability, still needs to ensure that (i) they do not interfere with primary users (*safety*) and (ii) it makes full use of the available white spaces (*efficiency*). While we describe our specific design and implementation in Sections III and V, respectively, we now outline three design principles that are key to building such a system.

Our principles are guided by answers to the following questions: How does the database determine white spaces availability, and how closely do the predictions match up with the ground truth? Should white space devices know their respective locations, and if so, to what resolution? And, what is the impact of mobility? The following three subsections give answers to these questions.

A. Determining White Spaces Availability

To learn the white space availability at a given location, the database could use one of two schemes: use spectrum measurements for that location, or compute spectrum availability using RF propagation models. The former, a data-driven approach, requires extensive wardriving measurements at low sensitivity thresholds and may take a long time to be complete. Furthermore, the measurements will have to be repeated whenever the primary user's transmission characteristics, such as transmit power, antenna height, license terms, etc., change. In our experience, these changes are not uncommon. The latter, a *model-driven approach* does not suffer from these drawbacks, and our SenseLess system takes this approach. However, the key question of any such model-driven design is the validity of the RF propagation models.

There exist a plethora of propagation models for UHF signals, each of varying complexity and accuracy. We use real-world measurements to compare the accuracy of different models for computing the available white spaces. Specifically, we compare the following four well-known propagation models: Free Space, Egli, and Longley-Rice (L-R) with terrain, and L-R (without terrain). We restrict ourselves to these models because as per the FCC's prior rulings [11], L-R is the default model to be used for predicting signal attenuation. We exclude detailed descriptions of these models due to lack of space. Briefly, Free Space is the simplest model and has been used to study white space availability in a number of research papers [20], [15]. Egli is often used to compute RF propagation in frequencies between 50 and 1900 MHz. L-R is a much more complex propagation model, that takes into account climactic effects, soil conductivity, permittivity, Earth's curvature, and surface refractivity, and has also been used by the FCC to compute TV contours [11]. L-R with terrain also takes elevation data as input and is very computationally intensive.

The question we seek to answer is, *how closely does the white spaces availability predicted by these models resemble the ground truth*? And, is relying solely on any of these models really a viable alternative to relying entirely on spectrum sensing? We are not aware of prior work that answers all these questions.

1) Measurement Methodology: We measured the UHF spectrum across the state of Washington. As shown in Figure 1, spectrum measurements were taken during the months of July-August 2009, across a driving path of 1500 miles, at a set of 57 diverse locations including large cities, downtowns, suburbs, between large buildings, mountain ranges, forests, valleys, at the edge of water bodies, and also across areas of different population densities.

Using a spectrum analyzer fitted with a UHF antenna, we measured the signal strength for all 30 UHF channels (21 through 51 except channel 37, which is reserved for

Fig. 1. A 1500 mile path over which we measured UHF spectrum at 57 different locations. The push pins indicate points of measurement.
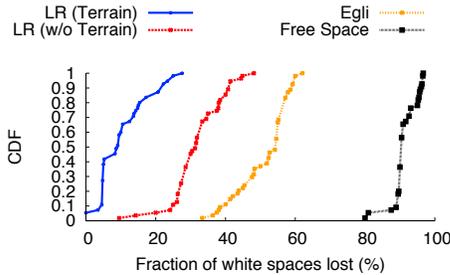

Fig. 2. Comparison with ground truth for different propagation models. We lose fewer white spaces by using terrain based Longley Rice propagation.

astronomy) by restricting the channel on the spectrum analyzer at the center frequency of every TV channel and a bandwidth of 6 MHz. We then measured the channel power of the spectrum across this 6 MHz band. We mark a channel as being occupied if the corresponding channel power is less than or equal to -81 dBm[1], and available otherwise. Using the same threshold, we compare these findings with those predicted by different models and high resolution terrain data (measured at 100 m intervals across the planet's surface). If a channel is occupied in the ground truth data but available in the models, we flag it as a *false positive*; if it is available in the ground truth data but not by the model, it is a *false negative*.

2) Comparison of Different Models: In our entire data set of 57 locations and 30 channels each, none of the models gave any false positives.[2] So, they all met the safety requirement. To quantify efficiency for each location, we express the number of false negative channels as a fraction of the total white spaces available at that particular location. The results are shown in Figure 2 which is a CDF of the fraction of channels lost because of false negatives. It can be seen that the median loss rate for L-R with terrain is only around 8% of the available channels. To us this was a surprising result since it suggests that with careful modeling it is possible to 1) *not* lose too many white spaces and at the same time 2) have very few or

[1]This is a limitation imposed by our spectrum analyzer hardware when measuring channel power values.
[2]In the free-space model, we used a very conservative path-loss exponent of 2. Using an exponent of 3, we measured several false-positives.

no false positives. In contrast, all other models result in many white spaces being wasted.

Hence, we can conclude that a purely model-based approach based on the L-R model with terrain data can predict white space availability at a low false negative rate. In other words, the *"loss of white spaces by not sensing"* is low. However, one observation with important systems implications is that complex and computationally-intensive models based on terrain data can achieve a satisfying accuracy; *using simplistic propagation models results in a waste of white spaces*.

3) More White Spaces using Variable Thresholds: We note that using a geo-location service, the BS can do better. A geo-location service allows the BS to determine the source of TV transmissions as well. Since the FCC has put forth different protection contours for different classes of digital or analog TV transmitters [2], the geo-location service can appropriately change the threshold for computing white spaces. For example, high power UHF DTV stations need a minimum signal strength of `51.0 dBu` at a receiver whereas for low power UHF DTV stations it is expected to be `41 dBu` [2]. There are similar such differences for analog TV transmitters as well.[3] The authors in [13] recognize the need for differentiated thresholds depending on the source of TV transmissions. We build on this insight and determine the precise thresholds to be used for each class of TV transmitter. In Section VI-A, we evaluate this approach via ground truth measurements and our geo-location service.

B. Need for Location

At first glance, it is not obvious why clients in white space networks that do not primarily rely on sensing need to be equipped with location information. As suggested by prior proposals [1], it must be possible for BSs to either use their own locations as a substitute for locations of clients associated with them, or to conservatively operate only on channels that are free in the entire coverage area of the BS.

The intuition is as follows. Each BS can merge the white spaces availability at every single location in its coverage area to derive a "comprehensive map" of white spaces availability. A channel that is free everywhere in its coverage area can be used to communicate with a client that is located anywhere in its coverage area. The problem is that this method can result in a loss of available white spaces as the comprehensive map is a result of a union of all occupied channels within the coverage area of BS. In this section, we quantify this loss (and thus also the need for client-location).

Methodology: We use L-R with terrain data (see Section III) to compute RF propagation from TV towers. We emulated the deployment of a BS in 90 points – 30 each of city downtowns, towns and suburbs in the US. We fixed the transmit power to 4W EIRP, as specified in the FCC ruling and computed the coverage area of the BS (again using L-R with terrain data model, see Section III). We divided the coverage area in

[3]The FCC has granted a longer transition period to certain classes of analog TVs to remain on the air.
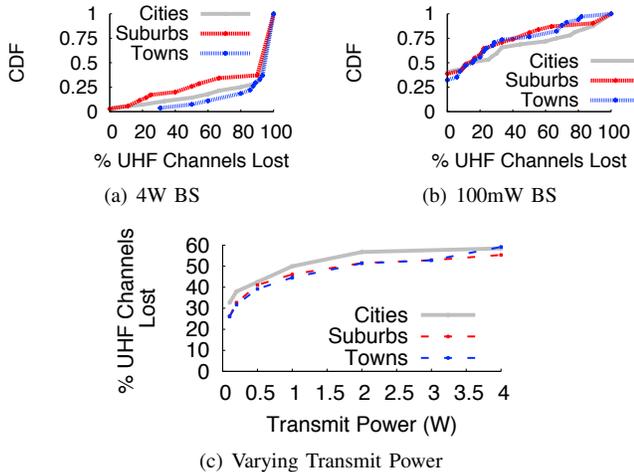
(a) 4W BS

(b) 100mW BS

(c) Varying Transmit Power

Fig. 3.   The impact of using BS location as a proxy for client location on the white spaces lost, as a function of the BS transmit power.
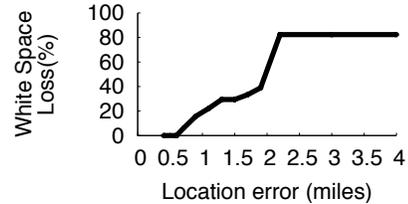

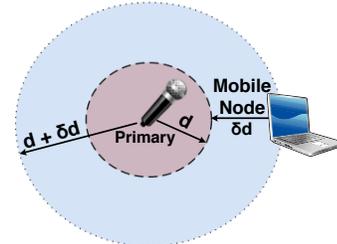Fig. 4.   The impact of location errors on the loss of white spaces.

Fig. 5.   Impact of mobility on the *protection* region around an incumbent. The extra δd is required to account for delay in informing the BS and client about the presence of a new incumbent.

100m×100m miles grid, and for every grid point we computed the available white spaces. We declared a TV channel to be available at a grid point, if the receive signal strength from any TV tower is not greater than -114 dBm, which is the sensing threshold specified in the FCC ruling. For any BS location, the number of white space channels lost is computed as $(C_{Loc} - C_{noLoc})/C_{Loc}$, where C_{Loc} and C_{noLoc} represent the number of available channels with and without location, respectively. C_{noLoc} is the number of channels that are available everywhere in the BS's coverage area: these are the channels a BS can select without having client location information (because the BS must assume that a client could be anywhere within its coverage range). In contrast, C_{Loc} is the number of common channels available between the BS and a client that is located randomly at one of the grid points. These are the number of channels the BS can select if it knows its client's location.

White Spaces Lost Without Location: In Figure 3 we plot the CDF for the number of channels lost across the 90 locations. The median loss of white spaces by using the BS's location as a proxy for client locations results is more than 80%. There are two reasons for this high loss. First, without knowing client location, the BS is *forced* to be very conservative and use only those channels that are available even in the most distant area of its coverage area. Second, due to changes in terrain profile, there are often points in the coverage area of the BS that receive better signal reception from TV transmitters than other points. Hence, the white spaces availability in such locations is lower than other surrounding points.[4]

We conclude that clients *must* know their locations for SenseLess to utilize the available white spaces efficiently.

Location Granularity: A remaining question is to what

[4]For this reason, a similar experiment using inaccurate propagation model *without* terrain data would result in a much smaller loss of white-spaces due to not knowing location.

resolution clients must know their location. GPS is accurate outdoors, but does not work in indoors. Cellular technologies can yield location errors of more than a mile [25]. Location errors are problematic as they can manifest themselves in the form of both false negatives and (even worse) false positives. In order to avoid false positives, an additional safety protection range would have to be used by the system, which would result in an additional loss of white spaces. To quantify the impact of location errors, we measure this loss in available white spaces. We show results for the worst of the 90 locations. As Figure 4 demonstrates, higher location errors causes significant reduction in the overall white spaces. For example, an error of 2.5 miles can result in 80% loss of white spaces. We conclude that *for a system without sensing at white space devices to function effectively, clients must have location information within 0.5 miles.*

C. Handling Mobility

In SenseLess every WSD primarily relies on the database to determine the white space availability. Hence, this results in a delay in the WSD learning about changes in spectrum availability. Either the WSD will have to poll the database, or the database will have to push updates to the WSD. This problem becomes worse when the devices are mobile. If mobile, the WSD could have travelled some distance between the time it receives two subsequent spectrum updates (Figure 5). To address this problem, mobile devices will have to add a *protection range* of δd to determine the white spaces that it can use. That is, even though a channel might be available at a device's location, it must not use it, if this channel is blocked off at any location within distance δd of the current location. For example, for a polling interval of 1 minute, a mobile client that accurately knows its location and can travel at 60 mph will have to add a protection range of 1 mile. As we see in Figure 4, this would result in 20% loss of available white space spectrum.
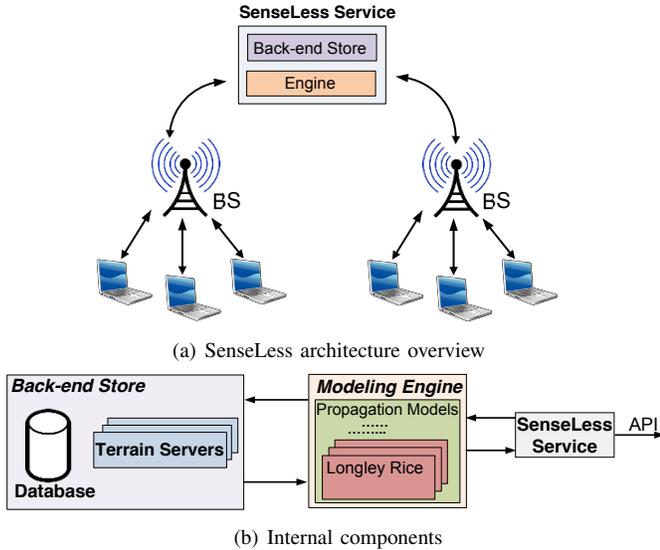
(a) SenseLess architecture overview

(b) Internal components

Fig. 6. The SenseLess System Architecture

API	Description
GetWhiteSpaces(L)	Returns white spaces at location L
GetPrimaries(L)	Returns list of primaries and RSSI at L
UpdatePrimary(p)	Update information on a primary p
Register(n, L)	Register BS/client n with grid location L
CoverageArea(b)	Computes coverage area for a BS b
SetCover(b)	Computes set cover channels for coverage area of BS b

TABLE I
SAMPLE API EXPOSED BY THE SENSELESS SERVICE.

Hence, we conclude that *in order to efficiently handle mobility, a white space network that relies on the database to determine white space availability must disseminate spectrum changes to the clients at very low latencies.*

D. System Design Considerations

In this section, we have derived characteristic system design constraints and challenges for a white spaces network driven by a database:

- The system must be driven by highly-sophisticated signal propagation models that are based on terrain data. Such models are computationally expensive.
- Using conservative estimates for clients with unknown location results in a significant waste of white spaces, i.e., client location must be incorporated into the design.
- In order to handle mobility, information dissemination latency must be very low because otherwise, a significant amount of white space is wasted.

Besides these design constraints, there is one more challenge that needs to be solved. Since clients lack sensing capabilities, there must be a mechanism for the system to learn about the existence of wireless mics. In the next section, we show how our system, SenseLess, addresses all the above constraints and challenges.

III. SENSELESS: SYSTEM DESIGN

In this section we describe the design of SenseLess, details of its internal components, and how SenseLess addresses the challenges derived in the previous section.

A. System Overview

Figure 6 illustrates the overall SenseLess architecture and its internal components. The aim of the system is to enable infrastructure-based wireless networks operating in white

spaces that primarily rely on the database as a means to determine white space availability.

At the heart of the system lies the *SenseLess Service*, a *logically centralized* entity to which client devices and base stations (BS) are connected to. The SenseLess service determines, for any given location, which parts of the spectrum are white spaces. Any change in spectrum occupancy (e.g. due to say a TV transmitter being switched off at night or a wireless microphone being switched on) is reflected in the SenseLess service. All WSDs, be they clients or BSs, rely solely on the SenseLess service to learn about the availability of white spaces at their respective locations.

B. SenseLess Service

The primary task of the SenseLess service is to respond with an accurate white spaces availability for a given location. The white spaces availability is a bitmap of channel availability at a given location, where location is represented by $L = $ Latitude, Longitude. The subset of the APIs provided by the service is shown in Table I.

The SenseLess service can operate in two modes. First, it can be queried giving a specific location as input. Alteratively, using a publish-subscribe model, the SenseLess service can track changes in the white spaces availability of a BS or its associated clients. On detecting changes in the white spaces availability at any of these locations, the service fires an event that informs the BS of the changes in white spaces availability either at the BS itself or at one its clients. Hence, to support both modes, an active connection is maintained between the service and each BS. The BS then disseminates this spectrum availability information to its clients.

Each WSD has associated with it, a set of BSs. In-turn each BS services a set of clients. Each time a client associates with a BS, this information is relayed to the database, which in turn maintains an active record of all the client devices it services across all BSs. Similarly when a client either leaves the network or times out (for example, when switched off), the BS relays this information to the database. Hence, the SenseLess service keeps track of all active WSDs in the network. This information, as we will discuss later may be used to enhance the operation of the WSD.

As demonstrated in the previous section, the accuracy of the spectrum map depends on the quality of the propagation model and fine granularity of the terrain data. Therefore, the SenseLess service is comprised of two components: the *back-end store* that maintains the terrain and primary user data (such as TVs and wireless microphones) and the *SenseLess*

engine that accurately computes the white spaces availability at a location. These components are illustrated in Figure 6(b).

C. Back-end Store

The back-end store consists of the database and the terrain server.

Database: The db stores the following information:

- Information on all television transmitters – including the TV tower location, channel, height, transmit power, antenna directionality, mechanical beam tilt, etc., all of which are updated on a daily basis. This is based on publicly available data from the FCC's Consolidated Database System (CDBS) [18]
- Wireless microphone registration data – including location of wireless microphone, channel, and time of last report.
- Information on all WSDs that are serviced by the database – including the location, ID, channel, height, transmit power, antenna information, time of the last report from the WSD, etc.
- The database also serves as a cache for the white spaces availability computed previously for various locations. We explain this further in Section III-D1.

Terrain Servers: The primary responsibility of the terrain server is to provide terrain elevation data at any arbitrary point on the planet's surface. It stores high resolution world-wide terrain elevation data. We obtain this data from various publicly available government efforts for mapping the planet's terrain. We use various elevation data sets, such as GLOBE [9], measured at 1 kilometer intervals across the planet's surface, or the high resolution SRTM data [7] from NASA measured at 30 meter intervals. The terrain server provides terrain elevation data between the two points at a specified interval. To this effect, the terrain server receives requests specifying (L_{TX}, L_{RX}, Interval) where L_{TX} and L_{RX} are the coordinates for the transmitter and receiver, respectively, and interval is the resolution at which terrain data must be sampled between these two points. Given this input, the server computes a direct path between the L_{TX} and R_{TX} along a great circle, and returns elevation samples between the two points at the specified interval.

D. SenseLess Engine

To accurately determine the white spaces availability at a location L, the SenseLess engine computes the attenuation of UHF signals via propagation modeling by performing the following steps.

1. List all primary user's transmitters within a large search radius from L. Conservatively, these are all the primary users whose signals could potentially be heard at L at an RSSI greater than the specified threshold.[5]

---

[5]For TV towers with 1MW Tx power, an antenna height of 200ft, and at a frequency of 515Mhz (lowest channel), that range can be computed to be roughly 300 miles.
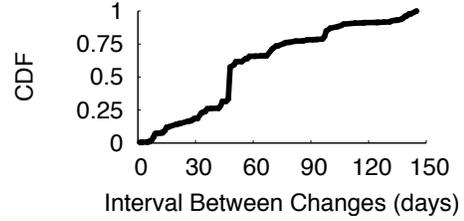

Fig. 7. Changes to entries in the FCC TV transmitter database over a period of five months (monitored on a day-to-day basis).

2. If the primary is a TV transmitter, the engine fetches the elevation data between the TV tower and L from the terrain server. If the primary is a mic, we assume a conservative fixed protection radius around it ([2]).

3. The elevation data in conjunction with the transmitter's parameters (such as height, power, antenna directionality, etc.) are used to determine the signal attenuation using the Longley-Rice (L-R) propagation model. The attenuation in turn is used to compute the RSSI of the transmitter at L.

4. The white spaces availability at location L is created. As part of this, a channel C is deemed to be blocked if there is at least one primary who is *active*, i.e., an incumbent whose RSSI at L is greater than -114dBm.

$$Map_L(C) = \begin{cases} 1 & \text{if an incumbent is active on C at L} \\ 0 & \text{otherwise} \end{cases}$$

The critical step in this process is determining the attenuation of a UHF signal over a certain distance and how this affects the RSSI. This is achieved via propagation modeling of UHF signals using the Longley-Rice propagation model, as discussed in Section II. We use the same parameters for L-R propagation as those specified by the FCC [11].

1) Improving Performance: The SenseLess engine plays a crucial role since it computes L-R propagation using terrain elevation data. This particular task is computationally intensive and can impact the latency of the results returned. To improve this latency, we make two key optimizations to the engine.

Precomputations: We monitored the FCC's TV tower database (updated daily) over a period of five months and found additions/deletions/changes to licenses were made on average, once in 2 days (Figure 7), which is relatively infrequent when compared to wireless microphones being switched on/off several time a each. Also, as seen in Figure 7, of the 10,000+ entries in the FCC database on average, fewer than 1% of the entries are changed, and we can quickly recompute those grids only these few entries cause. Hence, we note that once we compute the white spaces availability for a location, in most cases, we can cache and reuse this information for future queries. We enable this optimization and cache the white spaces availability for each grid point in our back-end database. We divide the planet's surface into grids of size 100m×100m. For each grid we store the list of incumbents whose signals are received at $\geq$ -114 dbm at the center of the grid. Note, events such as changes in a TV transmitter's configuration can impact some grid points. To

efficiently handle such events, we invalidate the channel in the white spaces availability of the affected grid points and recompute only the affected points. In response to queries for a given location, the SenseLess Engine returns the precomputed spectrum map of the nearest grid point. We evaluate the efficacy of this approach in Section VI.

Per-Region Terrain Server: The cost of fetching terrain data from the back-end terrain servers is high since such requests go over the network and read from disk. Consequently, we maintain a per-region terrain servers that store an in-memory copy of the region's terrain. As we show in Section VI, this optimization significantly reduces computation latency. We also maintain a terrain cache on the SenseLess Engine that corresponds to the latest tile fetched from the terrain server.

E. Accomodating Microphones

The FCC's ruling makes room for microphones by allocating two channels to mics across all loactions. In those locations where these two channels are unavailable, the ruling specifies alternate ways of allocating these channels. SenseLess is designed to accomplish these goals.

To account for wireless microphones at stadiums, matches, and other such one-off events, the ruling requires manual insertions of mics into the database. Each such insertion into the databse also contains a timestamp of the time at which the mic was inserted. SenseLess also supports *bounded* timeouts whereby, periodically, those mics that have expired, are culled from the database.

F. Administrative Control

A database of incumbents offers an additional opportunity to improve the performance of the white space network. WSDs periodically query the database to learn of the available white spaces. Hence, the SenseLess database also stores information on each WSD actively querying it. These are devices that are actively using the white spaces network. This includes the WSD's location, transmit power, frequency, etc. Hence, the database stores information on all incumbents and the WSDs actively using the white spaces. This approach permits the service to expose information on the entire operation of the network to all WSDs on the network, which in turn may use it to optimize their decisions such as frequency selectio, power control, and association. But the primary reason we embed this feature into SenseLess is for the network administrator to be able to shut down those WSDs that are deemed to be misbehaving.[6] Hence, we expose an API $Blacklist(C)$ for a system administrator to dissociate an offending client C and prevent it from associating with the network.

G. Authentication

For purposes of accountability and verifiability, SenseLess must authenticate new client WSDs joining the network. There are several approaches to address this problem. For example, using techniques similar to [14], the database can also maintain

a list of radiometric signatures of all WSD clients and use them to authenticate the client based on the first few packets transmitted (also known as a contact signal in the second FCC ruling [3]). However, we can also adopt other approaches based on cryptographic key exchanges or RADIUS-like [] techniques. In all these approaches, the SenseLess database can be leveraged to authenticate client devices.

IV. SYSTEM IN OPERATION

During the operation of the system, each BS must learn about which channels are available at itself and its clients in order to select an feasible white space channel. We now show how this information is conveyed to a BS.

Every client periodically sends its location and the technology used to determine the location to its associated Base Station. The BS uses this information to determine the commonly available white spaces at the client and itself. From among all these usable white spaces, the BS selects an appropriate chunk, and clients associate to the BS on this channel.[7] The BS also subscibes to push updates from the SenseLess Service at the client's location as well as grid points around it depending on the location error of the technology used to determine the location. We use conservative values for the location error, e.g. 50 meters for GPS, 150 meters for Wi-Fi and 1 mile for GSM based localization.

Thus, SenseLess requires all clients and BSs know their location. This design choice is motivated by our findings in Section II. In SenseLess, location can be determined by any means, including existing technologies based, e.g. GPS, Wi-Fi [8] or GSM [25]. In the future, we plan to use TV technology for localization [5].

The above approach works once all clients are associated to the BS and as long as white space availability does not change. We will now explain how SenseLess handles bootstrapping and spectrum updates.

1) Bootstrapping: Bootstrapping is a critical problem in any SenseLess-like system in which clients rely on the database to determine white space availability. When a new client arrives, it cannot transmit any packets since it does not know the white spaces available at its location. Hence, it is unable to even transmit its location to the BS since the corresponding packet could interfere with an incumbent nearby.

This bootstrapping problem would not occur if the BS always transmitted on a channel that is available at *every* location in its coverage area. In that case, a client could simply discover the BS and associate to it. While this would greatly simply our design, our results in Section II-B show that 1) a commonly available channel across the entire coverage area does not exist in most places, and 2) even in places where such a channel exists, using only this channel would result in a significant loss of white spaces. Hence, the channel selection at

---

[6]The problem of determining how a device misbehaves is orthogonal to this paper and requires further work.

[7]In this paper, we only consider the problem of determining the set of available channels. The problem of deciding *which* spectrum chunk among the available channels should be used is orthogonal and has been studied in prior work [12], [16].

the BS in SenseLess takes into account actual client locations, which results in bootstrapping becoming a challenge.

Channel Availability Beacons: To resolve the dead lock during the bootstrapping phase, every BS periodically (once every second) broadcasts a beacon containing the channel availability in *all* regions of its coverage area. Specifically, a beacon contains for each grid-point in its coverage area one available channel that a client located at that grid-point could use to contact the BS.[8] At a lower frequency (once every minute), the BS switches to each of these channels and listens for clients that want to join. The switching overhead can be reduced by using techniques presented in [12]. When a client joins the system, it listens for beacons from the BS and moves to the channel that is marked in the beacon as being available in the grid pertaining to the client's current location. The client then broadcasts its location on this channel, which is eventually picked up by the BS. Hence, the beacon is used as a lookup table by client when bootstrapping.

Algorithm for Reducing Beacon Size: A challenge when designing the above system is to efficiently communicate the channel availability for all regions in the BS's coverage area. Transmitting one channel for every grid-point in the coverage area of a BS will result in prohibitively large beacons. For instance, at a coverage range of 10 miles, and a grid-granularity of 100m, the beacon size would be more than 100 KB, if 5 bits are used to encode a channel number.

To reduce the beacon size, the BS performs the following algorithm. First, in the *initialization phase*, it queries the SenseLess engine to compute its *potential coverage area* (PCA). The PCA is defined as the set of grid points in $100m \times 100m$ steps where the receive signal strength from the base station is greater than -90 dBm, i.e. the receive sensitivity at the lowest rate for many wireless cards. The PCA is computed using the L-R model. For every grid point within its PCA, the BS then retrieves the set of available channels.[9]

During the *operation* of the network, the BS uses this information to compute a *minimum channel cover* (MCC). An MCC for a potential coverage area A is a set of channels of minimum cardinality, such that for every grid point location within A, there is at least one available white space channel in in the channel cover. Formally, let $\mathcal{C}$ and $\mathcal{L}_A$ be the set of channels and location in the potential coverage area A. A channel cover MCC_A is defined as a minimum-sized set $MCC_A \subseteq \mathcal{C}$, such that for every $L \in \mathcal{L}_A$, it holds that $|C \in MCC_A \text{and} Map_L(C) = 0| \geq 1$. In SenseLess, we use a standard greedy set-cover approximation algorithm to compute a good approximation to the MCC.

The BS now knows that at least one of the channels in MCC is available at all grid points in its coverage area. The beacon contains the following information.

- The set of channels included in the computed MCC_A.
- A listing of one available channel from among the channels in MCC_A for every grid point in its coverage area (given in row major form). Each channel is encoded using $\lceil \log(|MCC_A|) \rceil$ bits.

As we show in Table IV of our evaluation in Section VI, the size of MCC_A is at most 4 in the US, and hence, 2 bits per grid-location suffices. Finally, a BS uses RLE compression to compress MCC_A information about adjacent grids where the channel availability is similar. This significantly reduces the size of the beacon. We evaluate this approach in Section VI-C.

2) Spectrum Updates: To handle updates in spectrum availability, such as the appearance of a mic, a BS subscribes for *push updates* from the SenseLess engine for all grid points in its coverage areas. Therefore, when a mic is switched on and the SenseLess engine receives notifiction of this event, it first quickly determines those grids impacted by the mic. We show in Section VI, the time taken to do this is very low (less than 500 ms when supporting up to 1000 mics being concurrently switched on). Based on this information, the SenseLess engine determines if a mic is switched on in the coverage area of a BS. If that is the case, the BS is notified of those grids that are impacted by this change in the white spaces availability. This is done automatically as part of computing the PCA, as described above. When either a new primary user appears, or the parameters of an existing primary user change, the SenseLess engine computes the changes to the affected grid points around the primary user and sends the updates to all the BSs that have subscribed to changes for these grid points. If needed, the BS recomputes the spectrum it is operating over and/or the contents of the beacon.

We note that the above steps affect the *latency* in propagating a change in the available white spaces. We evaluate the latency in Section VI. We note that when a primary user is already active, our push-based system architecture is very nimble in handling client mobility. Since the BS has an up-to-date view of spectrum availability and the client locations, it can precompute and adapt in advance to the spectrum that is available at all its clients, thereby avoiding client disconnections as well as interference with the primary users.

As we have shown in Section II, keeping query latencies low is a critical design constraint in white space networks when supporting mobility. This is the prime reason for our system to adopt a *push-based model* as otherwise, client-devices would have to poll the database at very high frequencies.[10]

V. SenseLess: Implementation

We have implemented a prototype of the SenseLess that includes the service, engine, and the back-end store. The entire system is implemented using C# and consists of approximately 13000 lines of code.

---

[8]Notice that when estimating the coverage area of a BS, we can be conservative and assume a large potential area. Because spectrum availability computation at the BS takes into account, this does not have negative consequences on the amount of white spaces that can be used, it merely increases the beacon size.

[9]Note that this is a one-time step and is performed only when the Base Station is first set up. Subsequently, the SenseLess engine only sends updates.

[10]In fact, sensing-based white space devices have the same drawback; they will have to sense (=pull) at very short intervals in order to sustain client mobility. Thus, the FCC ruling mandates client devices to sense every 30s.

Back-end Store: We use SQL Server to implement the database in the back-end. This database stores primary user data as well as precomputed white spaces availability for various locations (Section III-D1).

We implement per-region terrain servers as multi-threaded processes that store a large in memory cache of the terrain data. We run this process on server class machines with at least 8GB of RAM. Hence, we limit the in-memory cache for a terrain server to 4GB. We have also implemented a terrain cleaner process that periodically removes untouched or old elevation tiles from memory.

SenseLess engine: The SenseLess engine is implemented as a multi-threaded service capable of accepting multiple requests for either white space availability or updates to the back-end databases. We dedicate some of these threads for process requests for inserting the microphone information. As part of the engine, we have also implemented a library of various propagation models, including the point-to-point variant of L-R [10] and a broadcast version that does not factor terrain data L-R (no terrain). We have also implemented other popular and simpler UHF propagation models such as Okumura-Hata, Egli, and Free Space. All propagation models are implemented as dynamic-link libraries that can be loaded/unloaded easily at run time. BSs access the SenseLess service remotely via the SenseLess service. We divide the entire US into grids of area 100 sq. m since this is close to the smallest possible resolution supported by our elevation data. Our investigations reveal white spaces availability due to TV transmitters tend remain more or less uniform within the grid. The local terrain cache in the SenseLess engine is a 10 MB cache that stores the recently fetched terrain data from the terrain server.

SenseLess service: BSs and clients to interface with the SenseLess via the SenseLess service. We have two implementations of the service: (i) as a web service and (ii) as a multi-threaded socket based server. While we use the latter for all our experiments, we consume the web service to build a front-end web based GUI for SenseLess

Terrain Data: We obtain the terrain elevation data from publicly available high resolution SRTM4 and GLOBE data sources, which is maintained by NASA. The SRTM4 dataset contains elevation data measured from satellites at approximately 100 meter intervals on the planet's surface while the GLOBE data is measured in 1 km intervals.

Adaptive Elevation Selection: Due to errors in measurement and conversion, it is common to find 'holes' in the elevation data sources. These are more prevalent in higher resolution data such as SRTM than in GLOBE, which is a low resolution data source. Hence, governmental agencies responsible for the data take additional steps to 'fill in' these holes via a variety of techniques such as bilinear interpolation or gradation smoothing. However, in spite of such efforts, anomalies in the terrain data continue to exist, which in turn, impacts the results of the terrain based propagation models. For example, our L-R implementation returns an error code signalling errors in which parameters are out of range. We have found, these are typically caused by errors in the elevation data. To counter such anoma-

Additional channels gained	Number of Locations
0	2
1	10
2	3

TABLE II
GROUND TRUTH MEASUREMENTS OF ADDITIONAL WHITE SPACES GAINED BY VARIABLE THRESHOLDS ACROSS 15 LOCATION IN MASSACHUSETTS.
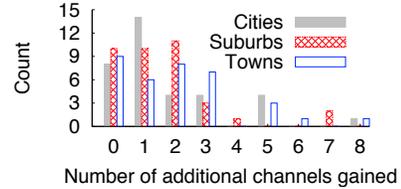

Fig. 8. Predicted gain in white Spaces from variable thresholds.

lies, we implement an adaptive elevation switching strategy whereby for a signal propagation computation between two points, we first try the higher resolution elevation data. If the propagation model returns an error stating some parameters are out of bounds, we then examine the path profile between the two points to determine if there are holes in the path. If that is the case, we then switch to the lower resolution GLOBE data and try the same again. If that fails as well, we finally switch to the L-R (no terrain), an L-R variant that computes signal propagation without taking elevation data into account (as opposed to point-to-point which factors in elevation data between two points). As we will demonstrate in Section VI-B, this strategy significantly improves the accuracy of the results by reducing false positives and false negatives.

VI. EVALUATION

We evaluate three aspects of SenseLess: (i) effectiveness in handling holes in the terrain elevation data, (ii) efficacy of our beacon reduction technique, and (iii) system scalability.

In our experiments the SenseLess Engine and back-end databases ran on a single server class 64-bit quad-core machine with 8GB of RAM. We dedicated a separate machine with 4GB of RAM as a terrain server. To emulate BSs and clients, we used a rack of 4 Blade workstations, each equipped with 4GB of RAM. All nodes were connected to each other using a 100 Mbps subnet.

All measurements and evaluations of SenseLess are performed across 33 cities, 30 suburbs, and 27 towns. We picked these locations based on populations distributions as advertised in various mainstream media such as CNN, Economist, Yahoo Travel, and BusinessWeek as being the most liveable places in the US. For each one of these locations, using SenseLess we deployed a BS for our experiments. Hence, in essence, we had 90 BSs. We set the default EIRP of a BS to 4W, the limit allowed by the FCC ruling.

A. Variable Thresholds

As described earlier in Section II-A3, apriori knowledge of a TV transmitter type can be used to set different detection thresholds and potentially open up more white spaces. We

Elevation Source	# False Positives
Adaptive	0
SRTM only	21
GLOBE only	25

TABLE III
IMPACT OF VARIOUS ELEVATION SOURCES ON THE FALSE POSITIVES IN
SENSELESS WHEN COMPARED TO THE GROUND TRUTH MEASUREMENTS.

Category	Channels Needed
Cities	4
Suburbs	4
Towns	3

TABLE IV
THE *minimum* NUMBER OF CHANNELS NEEDED TO COVER THE ENTIRE
COVERAGE AREA FOR A WHITE SPACES BASE STATION DEPLOYED IN 90
DIFFERENT CITIES IN THE US.

Fig. 9. Size of compressed beacons as computed across various locations in
the US

evaluate the benefits of this approach via (i) outdoor measurements of TV signal reception at 15 different locations in Massachusetts and (ii) predicting TV reception at the 105 locations using the geo-location service. To the best of our knowledge, this is the first such empirical and modeling driven study to show the benefits of variable thresholds.

Using the TV contour and co-channel interference criteria provided in the FCC ruling [2], we calculated the thresholds for the different classes of TV transmitters. Using a hardware platform similar to the one described in [24], we measured TV reception on UHF channels (21-51) across a 400+ mile driving path in the state of Massachusetts[11]. For each channel, we measured TV reception down to a threshold of -114 dBm, the minimum specified sensing threshold for TVs as proposed by the FCC [2]. For each location L, we use the geo-location service to determine the type of the TV transmitter t we expect to detect on a particular channel. From this we determine $Thresh_t$, the appropriate threshold a TV station of type t. If the geo-location service determines two (or more) TV transmitters to be active on the same channel at L, we pick the lower threshold. If $RSSI_L^t$, the measured RSSI at L from a TV transmitter t, meets the condition: $-114dBm \leq RSSI_L^t \leq Thresh_t$, we classify t to be *inactive* at L. If a channel has no *active* TV transmitters left, we declare such a channel as being a white space.

The results from our empirical study using this approach shown in in Table II demonstrate that on average we gained at least 1 channel across most locations, which is on average 50% more white spaces at these locations. We conducted a similar study across the 105 locations using only our geo-location service. The results presented in Figure 8 show that by using variable thresholds, we gain white spaces across all locales. Hence, *there are practical realizable benefits for variable thresholds.*

B. Countering Holes in Elevation Data

As outlined in Section V, holes in terrain elevation data can cause false positives, i.e. claim the channel to be free when there is a primary user. This is in violation to the safety requirement of white space systems. To show the effectiveness of our approach (Section V) in filling the holes and resolving this problem, we compared the number of false positives and false negatives with measured ground truth data (Section II-A) for two cases: (i) when using L-R with only a single elevation source, and (ii) using L-R with our adaptive elevation technique. We found all these approaches yield a similar number of false negatives. However, as seen in Table III, only the adaptive

[11]We elide a map of the driving path due to a lack of space
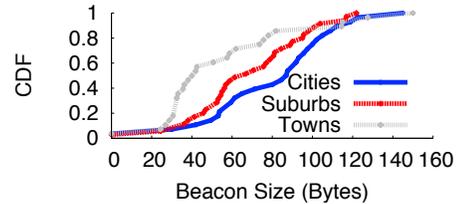
elevation technique results in zero false positives. Hence, it is important to adaptively switch between elevation sources when modelling.

C. Reducing Beacon Sizes

As discussed in Section IV-1, BSs need to beacon spectrum availability in their coverage area. We first measured the value of MCC_A, i.e. minimum number of channels, such that at least one of them is available at any point in the BS's coverage area. As seen in Table IV, the worst case MCC_A is 4. Therefore, 2 bits are sufficient to encode the channel availability at a grid point in the coverage area of a BS.

We then measured the efficacy of our beacon compression approach over uncompressed beacons. Figure 9 shows our technique results in a median beacon size of 40 bytes for towns, 70 bytes in suburbs and 85 bytes in cities. We also measured the reduction in beacon size due to our compression algorithm, and present the results in Figure 10. As seen in the figure, our compression algorithm reduces the beacon size in more than 50% of cities and suburbs by more than 62%. We note that we do not see as much benefit in towns since most towns we measured at have irregular terrain and hence, BSs in these areas have a smaller coverage area. Hence, as seen in Figure 9, the beacon size in towns is smaller as well.

D. System Scalability

We now evaluate two optimizations that improve the scalability of our system: (i) Using the terrain server's cache, and (ii) Pre-computing the white spaces, and then evaluate the scalability of our scheme for propagating spectrum updates.
Terrain Server Cache: To quantify the benefit of the terrain server's cache, we measured the time taken by the SenseLess Engine to compute white space availability at randomly chosen points in the state of Washington, with and without the terrain server's cache. We also varied the number of simultaneous requests that were sent to the SenseLess Engine, and plot the results in Figure 11. As we see in the Figure, increasing the number of simultaneous requests increases the latency in computing all the white spaces. Adding caching at a terrain
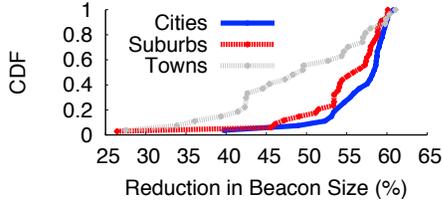
Fig. 10. CDF of reduction in beacon sizes due to beacon compression at different locations. Beacon compression can reduce beacon sizes by as much as approximately 62% in the median case.
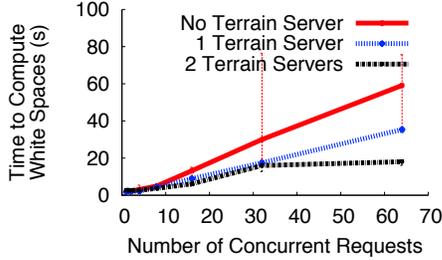

Fig. 11. Impact of using Terrain servers on the time taken to compute white spaces availability across the state of WA, as the number of concurrent requests vary (25th & 75th percentile error bars).
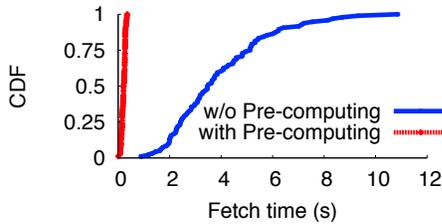

Fig. 12. Time taken to compute white spaces availability across 384 different locations using 32 concurrent requests, with and without precomputing.
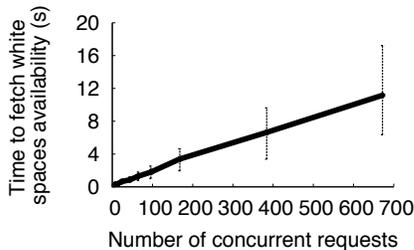

Fig. 13. Time taken to fetch white spaces availability across a varying number of concurrent requests, with precomputing enabled (25th & 75th percentile error bars).

server reduces this latency by more than 50%. Adding per-region terrain servers further reduces the time to compute white spaces.

Precomputations: Precomputations help the SenseLess to quickly retrieve the white space availability for a given location. To quantify this benefit, we computed the time taken by SenseLess to compute the white space availability at 384 locations in WA, with and without precomputations. Terrain cache was enabled in these experiments. We sent requests for these locations over 32 parallel threads. Figure 12 shows the CDF of the time to fetch the white space availability in these locations. The median time to retrieve the white space

availability reduces by more than an order of magnitude when using precomputed white space availability. We also evaluated this time when varying the system load, and present the results in Figure 13. We used 4 blade servers to issue the concurrent requests. As we see in the Figure, the mean time to compute white space availability for even 684 parallel requests is 10 seconds. As discussed in Section II, for a device moving at 60 mph, this corresponds to a extra protection range of 0.1 mile, which we showed in Figure 4 to cause only few white spaces to be lost.

Propagating Spectrum Updates: An important aspect of scalability for SenseLess, is its ability to compute those grids impacted by a wireless microphone and quickly push updates about these grids to those BSs and clients that 'subscribe' to these grids. The latest FCC ruling requires microphone operators to either use channels dedicated to them or to register the microphones well in advance in the database. However we also perform experiments to demonstrate SenseLess's bility to scale as a function of the number of events. Further results are presented in our technical report [22].

VII. DISCUSSION

We now discuss some features and potential limitations of our design, as well as future work.

Dynamic RF Management: In SenseLess, the database also stores information on actively operating WSDs. Hence, employing techniques similar to [23], SenseLess also provides a framework within which the database administrator provides additional services such as channel assignment, power control, client-BS association, and handoff algorithms. All of these factors affect how the RF spectrum is used by the WSDs. The SenseLess administrator may customize the performance of the network by programming deployment specific policies. Alternatively, the SenseLess service can also expose the raw data it collects on actively operating WSDs, to all WSDs serviced by the database. Each WSD can in turn use this information to optimize its decisions on how it uses the RF spectrum. For example, using the information on all WSDs exposed by SenseLess, BSs may perform various optimization locally and manage their clients. Hence, SenseLess permits wireless network management at various points in the network.

Determining Location: As described in Section II, it is important to know the location of the devices within a few hundred meters; otherwise too much white space has to be conservatively blocked off by the service to ensure protection of primary users. Currently, we use GPS to determine the location. However, we realize that GPS is power-hungry and does not work indoors. To solve this problem, we plan to use other approaches, for example TV tower localization technology, e.g. ROSUM [5]. Such a solution neither requires extra radios/antennas nor does it consume more energy.

Disconnected Operation: Our current implementation of SenseLess works when the clients and BSs can connect to the service over the Internet. However, in some scenarios, such as in peer-to-peer ad-hoc mode, or in the worst case of Internet failure, this connectivity might not be available. We

are currently working on extending SenseLess to work even when the clients cannot connect to the Internet. Our approach is to have clients carry a compressed form of the precomputed spectrum map for the region they are located. This can be downloaded when they have connectivity, say at night. The main challenge in this approach is to ensure consistency across multiple replicas of the database.

VIII. Related Work

The FCC has mandated the use of a geo-location service [3] and consequently several Internet websites offer the ability to view the TV coverage at a given location [4]. Spectrum Bridge recently released a web site, called Show My White Space [6], that shows the white spaces available at a given location. Although, we support a similar front end to SenseLess it differs in several key aspects. First, the focus of SenseLess extends beyond discovering nearby primaries. SenseLess is a complete system for *enabling* white spaces networking without relying on sensing. This also includes a set of design principles we espouse in our system. We examine the systems and networking challenges that arise when driving a white spaces using a database. SenseLess also exposes a set of APIs (Table I) that are not exposed by these services. Second, to the best of our knowledge, we are unaware if our elevation switching strategy is adopted by these other services, which we have shown as being crucial to reduce false positives and negatives. In [17], the authors propose an API framework for cognitive radio operation.

The closest work to SenseLess was proposed by Gurney et. al. [21] which presented the design of a geo-location database for white spaces networking. Their system computed the available white spaces based on the transmit power of the white space device and without using terrain information. They presented a limited evaluation in the Chicago area. In contrast, we use a more accurate propagation model that uses terrain data (Section II). We consider various challenges in interfacing with white space client devices that arise due to location inaccuracy, bootstrapping and client mobility. We also provide a thorough evaluation of SenseLess.

IX. Conclusions

In this paper, we present an alternative technique, called SenseLess, that leverages propagation modeling at a server over the Internet to determine white space availability. As part of our system, we demonstrate the need for good propagation models, accurate knowledge of client location and quick dissemination of spectrum updates. We build such a system and it evaluate its efficacy via ground truth measurements. We also show how such a system can be extended to serve up future extensions and needs in a white spaces network.

Acknowledgments

References

[1] Authorized Ex Parte Contact - Unlicensed Operation in the TV Broadcast Bands (ET Docket No. 04-186), FCC.
[2] FCC Adopts Rule For Unlicensed Use Of Television White Spaces, http://www.fcc.gov/.
[3] FCC Frees Up Vacant TV Airwaves for "Super Wi-Fi" Technologies and Other Technologies, http://www.fcc.gov/.
[4] http://tvfool.com/.
[5] Rosum corporation announces successful dvb-h positioning trial in collaboration with uks national grid wireless, http://www.rosum.com/pdfs/rosum_nationalgridwireless_dvb-h_technologytrial.pdf.
[6] Show My White Space, http://showmywhitespace.com/.
[7] Shuttle Radar Topograph Mission (SRTM), http://www2.jpl.nasa.gov/srtm/.
[8] Skyhook Wireless, http://skyhookwireless.com/.
[9] The Global Land One-Km Base Elevation Project (GLOBE), http://www.ngdc.noaa.gov/mgg/topo/globe.html.
[10] The ITS Irregular Terrain Model Algorithm, NTIA, Department of Commerce.
[11] OET BULLETIN No. 69, Longley-Rice Methodology for Evaluating TV Coverage and Interference. February 2004.
[12] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh. White Space Networking with Wi-Fi like Connectivity. In *SIGCOMM*, 2009.
[13] R. E. D. Borth and B. Oberlie. Considerations for Successful Cognitive Radio Systems in US TV White Space. In *DySpan*, 2008.
[14] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. PARADIS: Physical 802.11 Device Identification with Radiometric Signatures. In *MOBICOM*, 2008.
[15] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols (1998). In *MOBICOM*, 1998.
[16] S. Deb, V. Srinivasan, and R. Maheshwari. Dynamic Spectrum Access in DTV Whitespaces: Design Rules, Architecture and Algorithms. In *MobiCom*, 2009.
[17] G. Denker, D. Elenius, M.-O. S. Rukman Senanayake, C. Talcott, and D. Wilkins. Cognitive policy radio language (coral). a language for spectrum policies xg policy language. 2007.
[18] FCC Media Bureau. TVQ TV Database, http://www.fcc.gov/mb/video/tvq.html.
[19] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-Blog: sharing and querying content through mobile phones and social participation. In *MOBISYS*, 2008.
[20] A. Goldsmith. Design and Performance of High-Speed Communication Systems over Time-Varying Radio Channels. *PhD Thesis, University of California at Berkeley*, 1994.
[21] D. Gurney, G. Buchwald, L. Ecklund, S. Kuffner, and J. Grosspietsch. Geo-location Database Techniques for Incumbent Protection in the TV White Space. In *DySpan*, 2008.
[22] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl. SenseLess: A Database Driven White Spaces Network. Technical Report MSR-TR-2010-127, 2010.
[23] R. Murty, J. Padhye, A. Wolman, and M. Welsh. Dyson: An Architecture for Extensible Wireless LANs. In *USENIX ATC*, 2010.
[24] S. Narlanka, R. Chandra, P. Bahl, and I. Ferrell. A Hardware Platform for Utilizing the TV Bands with a Wi-Fi Radio. In *IEEE LANMAN*, June 2007.
[25] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara. Accurate gsm indoor localization. In *UBICOMP*, 2005.