usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Beyond Sensing: Multi-GHz Realtime Spectrum Analytics

Lixin Shi, *Massachusetts Institute of Technology;* Paramvir Bahl, *Microsoft Research Redmond;* Dina Katabi, *Massachusetts Institute of Technology*

https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/shi

This paper is included in the Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI '15).

May 4–6, 2015 • Oakland, CA, USA

ISBN 978-1-931971-218

Beyond Sensing: Multi-GHz Realtime Spectrum Analytics

Lixin Shi* Paramvir Bahl† Dina Katabi*
*Massachusetts Institute of Technology, {lixin, dina}@csail.mit.edu
† Microsoft Research, Redmond, WA, bahl@microsoft.com

Abstract – Spectrum sensing has been an active research area for the past two decades. Nonetheless, current spectrum sensing systems provide only coarse occupancy data. They lack information about the detailed signal patterns in each band and can easily miss fleeting signals like radar.

This paper presents SpecInsight, a system for acquiring a detailed view of 4 GHz of spectrum in realtime. SpecInsight's design addresses the intrinsic conflict between the need to quickly scan a wide spectrum and the desire to obtain very detailed information about each band. Its key enabler is a learned database of signal patterns and a new scheduling algorithm that leverages these patterns to identify when to sample each band to maximize the probability of sensing active signals.

SpecInsight is implemented using off-the-shelf USRP radios with only tens of MHz of instantaneous bandwidth, but it is able to sense 4 GHz of spectrum, and capture very low duty-cycle signals in the radar band. Using SpecInsight, we perform a large-scale study of the spectrum in 7 locations in the US that span major cities and suburban areas, and build a first-of-its-kind database of spectrum usage patterns.

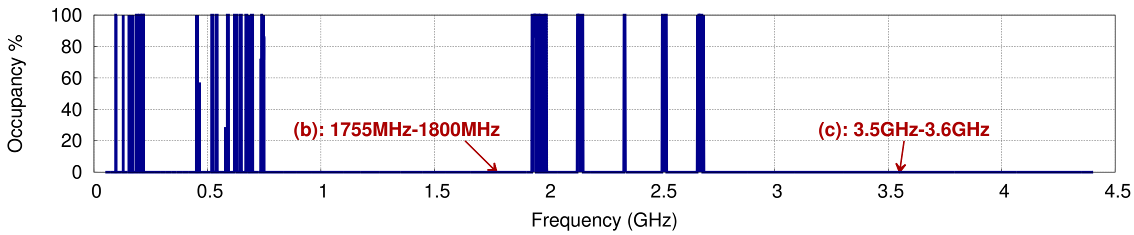
1 INTRODUCTION

There has been a significant interest over the past two decades in sensing the wireless spectrum and understanding how it is used [32, 34, 16]. Spectrum sensing has been a recurring topic not only for the research community [6, 26], but also for the government [29, 9], the military [1], and industry [20, 21]. Despite all of these efforts, our understanding of the wireless spectrum is still quite limited. State-of-the-art sensing equipment provide only coarse information of spectrum occupancy. Consider for example the Microsoft Spectrum Observatory (MSO), a state-of-the-art large-scale system for tracking spectrum usage [21]. Fig. 1(a) shows a typical MSO spectrum report. The figure reveals important information about spectrum occupancy, over a span of multiple GHz. Yet, the figure also misses informative details about how the spectrum is used. If one focuses the sensing resources on a single band and continuously listens to that band, one would discover that the above report has missed the fleeting (low duty-cycle) signal in the radar band around 3.5 to 3.6 GHz, which is shown in Fig. 1(c). In fact, not only did it miss the presence of the signal but it also missed how the signal uses the spectrum – i.e., its periodicity in time and its span in frequency. There are many signals that are missed in the MSO report. Fig. 1(b) shows another example. The band is used by the Air Force Satellite Control Network. The signal in the figure is difficult to catch since it hops in a 45 MHz band, occupying only 1 kHz at a time, i.e., its occupancy is 2×10^{-5}.

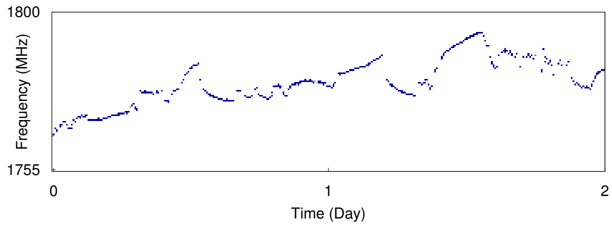
Learning the details of how the spectrum is used – e.g., the time-frequency utilization patterns in Fig. 1(b) and Fig. 1(c) – is fundamental to the design of dynamic spectrum access (DSA) systems as it can significantly increase the opportunity for spectrum sharing by leveraging signal periodicity. A band that has a periodic occupancy like the one in Fig. 1(c) can be easily time multiplexed with secondary users. The information can also reveal breaches of spectrum regulations by detecting abnormal utilization patterns, which would be invisible in coarse occupancy reports. The utilization patterns could also provide insight into the diverse technologies occupying the spectrum. The research community may know the technologies in the ISM and Cellular bands. Yet, the vast majority of the spectrum is occupied by undocumented technologies (e.g., radios in government bands), which are little known to the research community.

However, obtaining detailed spectrum utilization patterns is challenging, particularly for low occupancy signals like those in Fig. 1. Sensing hardware has limited bandwidth and cannot acquire multiple GHz in realtime. Therefore, spectrum sensing platforms like those used by Microsoft resort to sequential scanning of the spectrum; they hop from one band to the next, sensing only tens of MHz at any moment [21]. As a result, they obtain only high level occupancy statistics; but they can neither detect the low-occupancy signals nor identify their utilization patterns. Scaling the sensing system to a GHz-wide bandwidth, while obtaining fine-grained information about each band, is a significant challenge that remains unaddressed by past work.

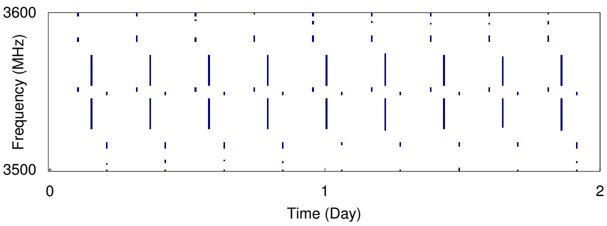
This paper introduces SpecInsight, a multi-GHz spectrum sensing system that reveals the detailed patterns of spectrum utilization in real-time. Underlying our design



(a) Microsoft Spectrum Observatory: the average occupancy at the Redmond station (with RfEye device) for one week (08/03/2013 – 08/09/2013)



(b) Army and Air Force Band (1755 MHz-1800 MHz)



(c) Radar Band (3.5GHz-3.6GHz)

Figure 1: Occupancy vs. Realtime Spectrum Patterns: The top graph shows an occupancy report obtained by the Microsoft Spectrum Observatory (MSO). Today's sensing reports can easily miss low occupancy signals. For example, the report in (a) has missed the Air Force Signal in (b) and the radar signal in (c). Graphs(b&c) are examples of SpecInsight's output, which captures the spectrum time-frequency patterns. The patterns are visualized as intensity maps, where the vertical and horizontal axes represent frequency and time respectively.

is a basic insight that any sensing system using a commodity radio is limited to tens of MHz at a time, and hence will have to sample the multi-GHz spectrum. The question, however, is: Which bands should we sample at what times in order to minimize the probability of missing active signals?

We address this question by observing that many spectrum bands are used according to some time-frequency patterns (e.g., always-on in time and frequency, always-on but hopping periodically in frequency, periodic in time but fixed in frequency, etc.). By learning these patterns, SpecInsight can schedule its scans of the various spectrum bands so as to maximize the probability that it will detect the presence, absence, and variation of spectrum utilization patterns, in every band.

SpecInsight implements this design principle in two phases. First, SpecInsight has an innovative algorithm for learning spectrum utilization patterns. In contrast to past work on detecting WiFi or other technologies in the ISM band, our algorithm has to search for previously unknown patterns without making assumptions about the technologies occupying a particular band. The output of the algorithm is used to populate a database of spectrum patterns and their locations. Second, SpecInsight has a smart scheduling algorithm that leverages the spectrum patterns in the database to sense multiple GHz using only tens of MHz of bandwidth, and still output the detailed spectrum utilization patterns as they occur in real-time. The algorithm is formalized as a multi-armed bandit game [11] in order to balance the tradeoffs between exploitation of known patterns and exploration of new and changing spectrum dynamics.

Implementation & Results: We have implemented SpecInsight using two USRP radios [8], equipped with the SBX and WBX daughterboards.[1] Our prototype senses over 4 GHz of spectrum, from 50 MHz to 4.4 GHz. We have compared SpecInsight with a setup that uses exactly the same hardware but sequentially scans the spectrum (similar to the Microsoft Spectrum Observatory). Our results show that the probability of missing active signals is $10\times$ lower with SpecInsight when compared to sequential scanning.

We have used the prototype to sense the spectrum in seven locations, including three major US cities and four suburban areas. We report the results of analyzing one week of data from each location and comparing their spectrum patterns. Our main findings are:

- Large swaths of the spectrum may appear completely empty when they actually have active signals. In particular, about 39% of the bandwidth below 4.4 GHz is used by signals whose occupancy is less than 0.0001, and hence are typically invisible to sequential scanning.
- One may think that the common way the spectrum is used is highly dynamic – i.e., a source may transmit at any time. We found that about 65% of the spectrum utilization patterns are either always on, or transmit periodically. Further, among the dynamic patterns, only 5% are highly dynamic[2]. Thus, knowing the spec-

[1]The use of two radios is not fundamental to our design but rather imposed by the range of frequencies of the USRP daughterboards.

[2]Defined as having a standard deviation of when the signal will next appear that exceeds 200ms.

trum patterns is highly useful for smart scheduling of sensing activities.

Contributions

- SpecInsight is, to our knowledge, the first spectrum sensing system capable of detecting and tracking fleeting signals (whose occupancy is ~10^{-5}) in multi-GHz spectrum, while using only tens of MHz of instantaneous bandwidth. Past systems have not been able to combine specificity with scalability: they either provide detailed spectrum occupancy in a single band, e.g., ISM [25, 14], or they obtain coarse occupancy data but miss low-occupancy signals like those in Figures 1b and 1c [21].
- SpecInsight introduces an innovative algorithm for learning spectrum usage patterns, and a smart scheduling algorithm for tracking the presence, absence, and variations of these patterns in realtime over a wide bandwidth of 4 GHz.
- The paper presents a large scale study of spectrum usage patterns in 7 US locations that span urban and suburban areas, illustrating which signal patterns appear in which parts of the spectrum.

2 Related Work

Past work on spectrum sensing may be divided into narrow-band and wide-band techniques. Narrow-band techniques assume the radio bandwidth is at least as wide as the sensed band. They focus on ways to accurately detect a signal. They may use energy level [31], cyclostationarity [14], signal waveform [34], wavelet transform [27], or response to interference [23]. Wide-band sensing techniques try to cover a wide spectrum significantly larger than the radio's own bandwidth. The traditional approach scans the spectrum sequentially and reports average occupancy [21, 34]. Some recent proposals exploit the sparsity of spectrum utilization to sense the spectrum without sampling it at the Nyquist rate, leveraging techniques like compressive sensing [24, 4] or the sparse FFT [12, 10, 13]. For example, BigBand [13] is able to recover the full signals in the spectrum, but under a sparsity assumption that only a small fraction of the spectrum is occupied, so it cannot be used in crowded spectrums, e.g., under 1.5GHz. Another scheme, QuickSense [33], employs a hierarchical search algorithm and analog filters to sense the white spaces, which spans only hundreds of MHz where the wireless technologies are mostly documented.

SpecInsight is a wide-band spectrum sensing technology. SpecInsight, however, differs from the above work in that it does not need sparsity assumptions or custom analog filters. Additionally, SpecInsight covers a wider band than this prior work and provides details of the usage patterns in each band (frequency hopping, periodic, continuous in time but not in frequency, etc.).

SpecInsight also builds on past work that proposed the use of sensing history for dynamic spectrum access [34]. Specifically, a series of theory papers [17, 36] models the behavior of primary users as a Markov process [36] and predicts future opportunities for dynamic spectrum access. SpecInsight differs from these past proposals both in objective and technique. Specifically, while they focus on finding some portion of the spectrum that is idle, SpecInsight focuses on exhaustively characterizing all active signals in the entire spectrum. As a result, the algorithms SpecInsight uses for characterizing historical patterns and scheduling sensing operations differ from the models in past work. Also, SpecInsight is focused on practical system design and empirical data and is supported by a spectrum study that spans multiple locations in the US.

Another line of work focuses on collaborative sensing, where different nodes share spectrum data in order to cover a large geographical area. For example, SpecNet [16] uses spectrum analyzers in different locations to sense the spectrum and share their results; V-Scope [35] mounts spectrum sensors on public vehicles and leverages mobility to enable large-area sensing of the white spaces. SpecInsight complements these systems by enabling *multi-GHz* spectrum sensing on relatively low-cost and easily accessible USRP radios.

Our work is also related to past literature on signal feature extraction. Many of these systems are focused on the ISM band with the objective of identifying WiFi interferers [19, 25, 14]. SpecInsight builds on the idea of signal feature extraction. However, it differs both in the features it extracts and the algorithm it uses to extract them. These differences stem from SpecInsight's use of features to identify spectrum utilization patterns that can be leveraged for smart scheduling of sensing operations, rather than to identify particular technologies. Additionally, SpecInsight spans a 40× wider band than the ISM band, and hence has to deal with a greater diversity of wireless techniques, of which the majority are undocumented.

Finally, our work supplements past work on large-scale spectrum measurements [6, 18, 26, 15]. First, our findings about spectrum occupancy and usage confirm many past spectrum observations; Second, by enabling wide-band spectrum sensing on low-cost devices, we believe SpecInsight opens up the possibility of even larger scale spectrum measurements.

3 SpecInsight's Design

The goal in designing SpecInsight is to build a tool for sensing spectrum usage, extracting occupancy patterns,

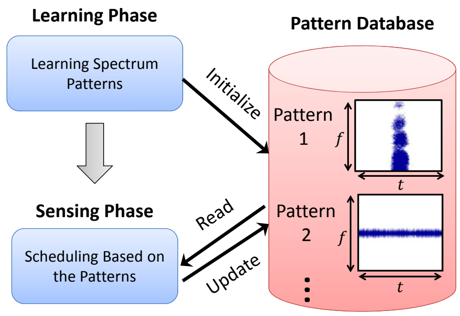


Figure 2: Flowchart of SpecInsight's Architecture: SpecInsight has two phases: the learning phase and the sensing phase. In the learning phase, SpecInsight extracts and learns the patterns in the spectrum and initializes the pattern database; in the sensing phase, SpecInsight uses the learned patterns to schedule when to sense each band. The pattern database stores and maintains the learned patterns, which are representative frequency-time blocks of the underlying signal.

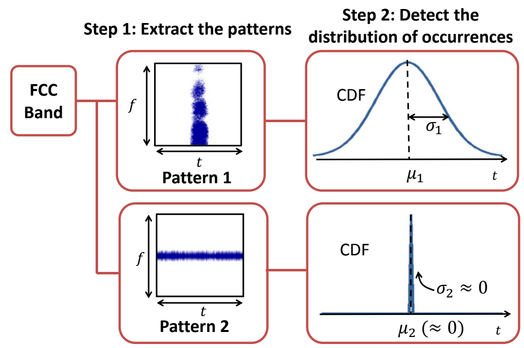


Figure 3: SpecInsight's Learning Phase: To extract pattern information in any given FCC band, SpecInsight employs two steps in the learning phase: 1) extract the patterns; 2) detect the distribution of occurrences of the patterns. The patterns extracted by SpecInsight, as well as the distributions of their occurrences are stored in the pattern database.

and detecting their repeated occurrences. Its key feature is the ability to provide realtime occupancy information of 4 GHz of spectrum using inexpensive commodity radios whose realtime bandwidth is limited to tens of MHz (e.g., USRPs). Anyone can download the SpecInsight software, deploy it on a USRP radio, and start sensing GHz of spectrum in their location.It not only senses a large bandwidth, but also provides finer details at each frequency, so that domain experts in each band can look into the spectrum patterns captured by SpecInsight for further analysis. We envision that such a system will help make wide-band spectrum sensing ubiquitous.

SpecInsight operates in two phases: a learning phase and a sensing phase. During the learning phase, SpecInsight sequentially scans the entire spectrum. It uses the collected data to extract and learn the different usage patterns which it then stores in a pattern database as shown in Fig. 2. Once the database has been populated with the usage patterns of each frequency band, SpecInsight goes into the sensing phase. It uses a smart scheduling algorithm to pick the best frequency band to sense based on the learned patterns. SpecInsight then collects signals in the chosen band and uses a pattern recognition algorithm to decide if the signals belong to a known usage pattern. If not, SpecInsight continues sensing that frequency band for an extended period to learn new usage patterns and update the pattern database.

What are the *patterns*? Spectrum *patterns* are a key concept in SpecInsight's design. A pattern is a representative time-frequency block which characterizes the underlying signal in both time and frequency dimensions. In the example of Fig. 2, pattern 1 spans the whole frequency bandwidth but is narrow in time, while pattern 2 reveals a utilization that is continuous in time but occupies a narrow bandwidth in frequency. The question now is, how do we determine the frequency and time widths of these blocks? On the frequency axis, SpecInsight sets the frequency range of a given pattern equal to one block in the FCC spectrum allocation table [2]. On the time axis, SpecInsight is presented with a trade-off: a short duration allows us to better detect fleeting signals while a long duration allows us to capture longer signals that repeat at a much larger time granularity. To be able to capture both types of signals, SpecInsight uses both short and long time durations. Specifically, in our implementation, we use durations of 5 ms and 50 μs.

For each time-frequency block as defined above, SpecInsight normalizes its power so that the maximum power is equal to 1. This is necessary since two wireless users with the same usage pattern can have significantly different power levels due to different signal attenuations from these users to SpecInsight's sensing antenna. Thus, if we do not normalize, two time-frequency blocks with the same usage pattern can be misidentified as two different patterns. Normalizing also allows us to match time-frequency blocks measured at different spatial locations which allows us to discover similar usage patterns across different urban and suburban areas.

Next, we describe how SpecInsight learns these patterns and uses them to schedule its sensing of each band.

4 The Learning Phase

In the learning phase, SpecInsight extracts and learns information of the spectrum patterns. This process is summarized by Fig. 3. Since SpecInsight divides the frequency spectrum into FCC bands according to the FCC allocation table, we focus only on a single FCC band in the following discussions. First, SpecInsight extracts patterns that exist in this band. Because some FCC bands (e.g., the ISM band) are shared by different types of signals, there might be more than one signal pattern in the band. In this case, SpecInsight extracts and records all of the patterns it can capture. Second, as shown in Fig. 3, SpecInsight keeps track of when each pattern repeats it-

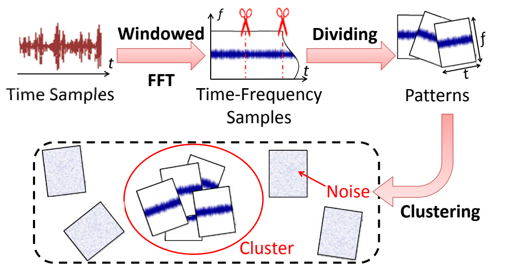
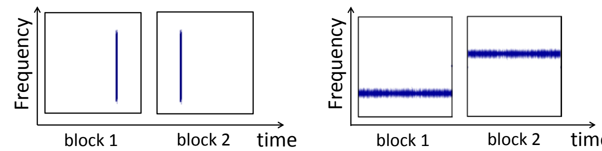


Figure 4: How SpecInsight Extracts Patterns in One FCC Band



(a) Two time-frequency blocks of a fleeting signal **(b)** Two time-frequency blocks of a frequency-hopping signal

Figure 5: Examples where Euclidean distance fails

self and draws the distribution of the time intervals between different occurrences of the same pattern. This distribution characterizes the timing properties of the underlying signal, e.g., a fixed-cycle signal would have a concentrated distribution while a dynamic signal would have a scattered distribution. SpecInsight stores the list of existing patterns and its corresponding distribution of occurrences in the spectrum pattern database. In the following two subsections §4.1 and §4.2, we describe these two steps in detail.

4.1 Extracting the Patterns

Fig. 4 outlines how SpecInsight extracts the patterns and identifies patterns from noises in a given FCC band. Since patterns are in the form of time-frequency blocks of signals, SpecInsight needs to first transform the I/Q time samples output [3] by the sensing hardware into two-dimensional time-frequency samples. SpecInsight does this by taking the FFTs over a sequence of successive time windows to obtain *time-frequency blocks*.[4] However, not all time-frequency blocks extracted by SpecInsight represent actual signals. Some of them might just be noise. So, how can SpecInsight tell signal patterns apart from noise? The intuition is that wireless signals intrinsically have certain regularities in the way that they use the spectrum, which are reflected by the time-frequency blocks SpecInsight extracts. On the other hand, noise is random. So if we run a clustering algorithm on the time-frequency blocks collected by SpecInsight, signal patterns will be clustered and noise will be filtered out. [5]

There may be multiple spectrum patterns in the same FCC band. In such scenarios, the clustering algorithm can also distinguish between the different patterns, i.e., blocks belonging to each utilization pattern are clustered together and separated from others. This is essential for SpecInsight's sensing phase, because the smart scheduling algorithm has different scheduling strategies for signals with different patterns (e.g., fixed-cycle or dynamic in time). Often each pattern ties to a specific technology, e.g., WiFi and Bluetooth are clustered to two different patterns; however, the goal of distinguishing different patterns is not to precisely identify wireless technologies, but to separate different patterns of spectrum utilization to sense the spectrum more efficiently.

4.1.1 Clustering Metric

Our clustering algorithm needs a distance metric in order to group time-frequency blocks into different clusters of usage patterns, where a small distance between two blocks means they are likely to be in the same cluster. A straightforward solution would be to use the Euclidean distance between two blocks. However, Euclidean distance does not work for some signals (e.g., the ones showed in Fig. 5 (a) and (b)), because it does not take into account possible shifts in the signals. For fleeting signals, the time pulse can appear at any time shift within each time-frequency block; for the frequency hopping signals, the center frequency in each time-frequency block can be different.

To solve this issue, we compute the shifted correlation between two time-frequency blocks. We shift the time-frequency blocks in both time and frequency and pick the minimum Euclidean distance across all shifts as our clustering metric. Formally, given two time-frequency blocks $B_1(f,t)$ and $B_2(f,t)$, our clustering metric is:

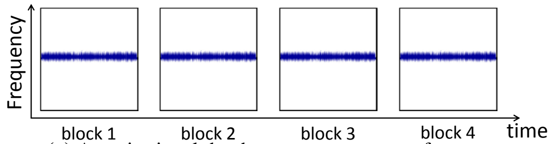
$$D(B_1,B_2) = \min_{\Delta f,\Delta t} \sum_{f,t} |B_1(f,t) - B_2(f-\Delta f, t-\Delta t)|^2 \quad (1)$$

where Δf and Δt represent any possible shift in frequency and time respectively. Using the above metric, we are now able to correctly cluster together the two time-frequency blocks in Fig. 5(a) and Fig. 5(b). Unfortunately, while the shifted distance metric solves the issue in Fig. 5, it creates a new problem that it can render two different usage patterns indistinguishable. For example, consider the two usage patterns in Fig. 6. Fig. 6(a) shows four time-frequency blocks of a frequency band with a static signal that has the same center frequency all the time and Fig. 6(b) shows four time-frequency blocks of a frequency band with a dynamic signal that hops from one center frequency to another. For any pair of time-frequency blocks in Fig. 6(a) and (b), the above distance metric will be small since the shifted correlation will

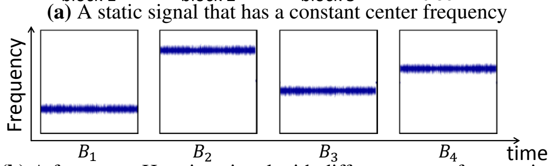
[3] I/Q samples are the real and imaginary parts of the time samples.

[4] SpecInsight also squares the magnitude since blocks are represented in terms of their powers.

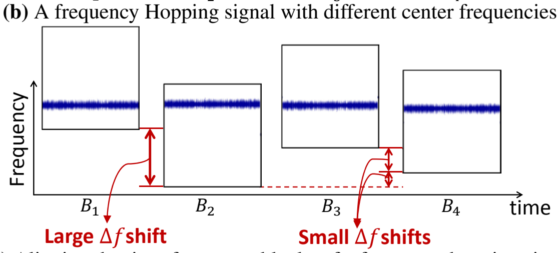
[5] Some signals like the direct spread-spectrum signals which are below the noise floor will not be captured by SpecInsight. However, without prior knowledge of the spreading codes, any energy-based detection will likely miss these signals.



(a) A static signal that has a constant center frequency



(b) A frequency Hopping signal with different center frequencies



(c) Aligning the time-frequency blocks of a frequency hopping signal

Figure 6: The Shifted Euclidean Distance

align the center frequencies in the blocks with the hopping signal (demonstrated in Fig. 6(c)). Hence, all these time-frequency blocks will be clustered together as the same pattern, while they are actually different signals.

To solve this problem, we constrain the time and frequency shift of the time-frequency block to a small range. Instead of computing the Euclidean distance in Eq. 1 for all values of Δt and Δf, we compute it only for a small range of Δt and Δf. To see how this approach can solve this problem, consider again the four blocks (B_1, B_2, B_3, and B_4) which contain a frequency hopping signal shown in Fig. 6(b). By constraining the shift, the distance metric between blocks B_1 and B_2 now becomes large because the center frequencies in B_1 and B_2 are far apart and cannot be aligned with a small shift as can be seen from Fig. 6. However, the distance metric between blocks B_1 and B_3 remains small since the center frequencies are near and can be aligned with a small shift. Thus, for a frequency hopping pattern, some pairs of blocks will have a small distance metric and some pairs will have a large distance metric. This will allow us to distinguish this usage pattern from the static usage pattern shown in Fig. 6(a) where all pairs of blocks have the same small distance metric.

The main question, however, becomes: *If two time-frequency blocks like B_1 and B_2 in Fig. 6(b) have a large distance metric, how can we still cluster them together?* Although B_1 and B_2 have a large distance metric, they are linked together via a chain of blocks that have small distance metrics. In other words, B_1 has a small distance metric with B_3 which in turn has a small distance metric with B_4 which has a small metric with B_2. Thus, although some of these blocks have large distance metric, they are still linked together via a *chain structure* which allows us to cluster them correctly as we will discuss in §4.1.2.

4.1.2 Clustering Algorithm

Machine learning provides us with a rich body of clustering algorithms. However, many of the well-known clustering algorithms such as the k-means do not work for this application. These algorithms are going to cluster together time-frequency blocks that have a small distance metric. As a result, they are not capable of capturing the *chain structure* cluster described above, where two blocks can have a large distance metric and yet belong to the same cluster. Thus, we need a clustering algorithm that is capable of clustering these chain structures.

To this end, we use the OPTICS algorithm [5]. This algorithm achieves exactly the above goal. At a high level, OPTICS is built on the concept of "reachability". Two time-frequency blocks are directly linked together if they have a small distance metric. Two other blocks B_1 and B_2 belong to the same cluster if there is a path of blocks that links B_1 to B_2. For example, in Fig. 6(b), the path was B_1, B_3, B_4, B_2. Thus, a cluster can be interpreted as a set of time-frequency blocks such that any pair of blocks can reach each other. Another advantage of the OPTICS algorithm over the k-means is that it does not require the number of clusters as an input. For the exact details of the OPTICS algorithm, we refer the reader to [5].

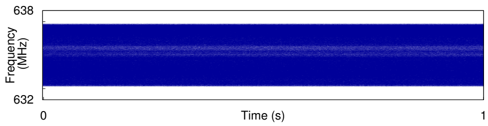
SpecInsight uses the OPTICS algorithm in two places:

- During the learning phase: SpecInsight runs the full OPTICS algorithm to cluster the collected usage patterns and establish a pool of patterns. The number and types of classes is data dependent. In §8, we describe the classes of usage patterns which are revealed by our experiments.
- During the sensing phase: SpecInsight uses OPTICS to cluster the newly sensed usage pattern and determine whether they belong to an already learned cluster of usage patterns or they form a new cluster of patterns that needs to be added to the pattern database.

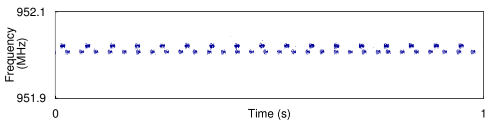
4.2 Detecting the Distribution of Occurrences

Once SpecInsight extracts and identifies a specific pattern, it tracks the different times when the pattern recurs and builds an occurrence distribution (step 2 in Fig. 3). SpecInsight defines the *pattern interval* τ as the time between two consecutive occurrences of the pattern, and the distribution of occurrences is defined as the statistical distribution of the pattern interval τ. It can be characterized by its mean μ and standard deviation σ, which SpecInsight computes over multiple measurements.

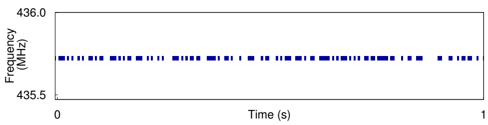
These statistics μ and σ are necessary to sense the spectrum efficiently. The mean μ determines the period of the pattern, and the standard deviation σ measures how dynamic the signal is. Thus, μ can be used to decide



(a) An Always-on Signal (White Spaces Channel 41)



(b) Fixed Cycle Signals (951.9 MHz - 952.1 MHz)



(c) A Dynamic Signal (940 MHz - 940.5 MHz)

Figure 7: Examples of Usage Patterns Over Time: Three types of signals are shown according to their timing characteristics: always-on signals, fixed-cycle signals and dynamic signals.

how often and at what time we expect to see the signal and σ tells us how precise our prediction is and can be used to decide the duration over which we should sense the band.

These distributions of pattern occurrences, as well as the pool of patterns that SpecInsight extracts and identifies, are stored in SpecInsight's spectrum pattern database (Recall Fig. 3 for an outline of what is in the database). In the following section, we will expand on how SpecInsight's sensing phase can utilize this database to sense the spectrum efficiently.

5 The Sensing Phase

After the pattern database is initialized in the learning phase, SpecInsight goes into the sensing phase and uses a smart scheduling algorithm to decide which frequency band to sense at each given time. Before we delve into the details, we will start with an example that gives some intuition behind SpecInsight's smart scheduling algorithm.

5.1 Intuition

SpecInsight's scheduling strategy builds on the following key intuitions. First, if a signal appears regularly every period, it will be much easier to catch this signal at its next predicted period even if it is a short fleeting signal. Second, we should spend more time sensing frequency bands with dynamic usage patterns and minimize the time we spend sensing bands with usage patterns that are static or have little uncertainty.

To better understand why this makes sense, let us consider three simple examples of usage patterns that have different time properties (i.e., their distributions of occurrences are very different): 1) always-on signals ($\mu \approx 0, \sigma \approx 0$) as in Fig. 7(a), 2) fixed cycle signals ($\sigma \approx 0$) as in Fig. 7(b), and 3) dynamic signals (σ is large) as in Fig. 7(c). Intuitively, for always-on signals, we can scan the frequency band less often in order to check from time to time that the signal is still there. For fixed-cycle signals, we can predict exactly when the signal is going to appear and sense the band precisely at that time. We also might want to check at times when we predict the band to be idle in case our prediction is wrong and there is another user using the band with a different usage pattern. For dynamic signals, the best strategy would be to sense the band at random times but for longer durations. We can afford to sense these bands for longer time given the time we saved on bands with always-on and fixed-cycle signals.

This gives the intuition. In the following section we will formalize this intuition into the *smart scheduling algorithm* that SpecInsight employs in its sensing phase.

5.2 The Smart Scheduling Algorithm

The smart scheduling algorithm needs to answer two main questions:

- Which frequency band f to sense next?
- How long to stay in a frequency band f?

Which frequency band f to sense next? Answering this question requires balancing a trade-off between exploitation and exploration. On one hand, we can exploit the information we learned from the sensing history to schedule brief checks on the next occurrence of a signal in some frequency band. On the other hand, due to the dynamics of the spectrum, the history information we have might not be accurate. So we need to keep exploring the spectrum in order to discover new usage patterns.

To address this trade-off, we formulate the problem as a *multi-armed bandit game* [11]. The multi-armed bandit game is a well studied problem in decision theory. In this game, the gambler needs to iteratively choose from K bandit machines, each of which will give her random rewards according to an unknown distribution. Her goal is to maximize the rewards in a given number of rounds. The gambler could learn the distribution by repeatedly pulling the levers. She then needs to decide whether to exploit the information she learned and choose the lever that maximizes her expected payoff or to just explore more in order to better learn the distribution.

There is a large literature of solutions to the multi-armed bandit game [30, 11]. In our implementation, we adopt a simple but very effective solution called *the ε-greedy strategy* which provides a very good approximation to the optimal decision [30]. In this solution, gambler simply chooses the lever that maximizes her expected payoff for $(1-\varepsilon)$ of the time and for the remaining ε of the time she picks a lever at random. The choice

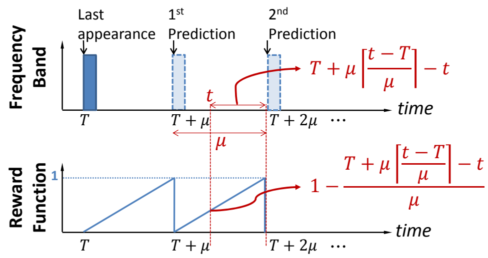


Figure 8: The Reward Function: The reward function $R_f(t)$ shows how near we are to the next signal appearance. It is zero at the beginning of a predicted period and one at the end, while linearly increasing as we get nearer to time we predict the signal will appear.

of ε defines the degree to which we rely on the learned information and ε is traditionally set to 0.1 [30].

Thus, 10% of the time, SpecInsight is going to pick a random frequency band to sense and 90% of the time, it will pick the band that gives it the maximal reward. But what is the reward function that SpecInsight needs to maximize? SpecInsight avoids missing a signal by going to its frequency band just before it expects the signal to appear. As a result, SpecInsight uses an indication of how near we are to the next expected appearance of a signal in the frequency band as its reward function. Formally, we calculate the reward function for a frequency band f at time t as:

$$R_f(t) = 1 - \frac{T + \mu \lceil (t-T)/\mu \rceil - t}{\mu} \tag{2}$$

where T is the last time the signal was observed and μ is the mean value of the pattern interval time as described in §4.2. The reward function is normalized to 1 in order to compare bands with different mean pattern interval μ.

To better understand this reward function, consider the example shown in Fig. 8. Given the last appearance of a signal at time T and the expected cycle μ, we predict the signal will appear again at times $T+\mu, T+2\mu, T+3\mu, \cdots$. Thus at time t, we predict that the signal will appear next at time $T+\mu\lceil (t-T)/\mu \rceil$ and we are $T+\mu\lceil (t-T)/\mu \rceil - t$ away from it. Since the farthest we can be away from the next appearance is μ, we normalize by μ and subtract it from 1 so that the nearer we are, the larger the reward function is.

How long to stay in a frequency band f? Once SpecInsight decides which frequency band to sense, it needs to decide how long to stay in that band. We refer to this as the dwell time t_d. The dwell time is determined by the number of measurements (time-frequency blocks) we need to collect in each band. It is directly related to the dynamics of the pattern, for the following reason: The more dynamic the usage pattern is, the more uncertain we are of our predictions, so that the offset between the predicted occurrence of the signal and the actual occurrence is bigger. To compensate for that, we need to have

Algorithm 1: Smart Scheduling Algorithm

Procedure SMARTSCHEDULING($\{f\}, \{\mu\}, \{\sigma\}, \{T\}$)
  $t \leftarrow$ Current Time
  if RAND([0,1]) $< \varepsilon$ **then** ▷ The ε-greedy strategy
    $f^* \leftarrow$ RAND($\{f\}$) ▷ Pick random frequency
  else
    for f in $\{f\}$ **do**
      $\mu, T \leftarrow \{\mu\}_f, \{T\}_f$
      if $\mu \neq 0, \infty$ **then**
        $R_f(t) \leftarrow 1 - \frac{T+\mu\lceil (t-T)/\mu \rceil - t}{\mu}$
      else
        $R_f(t) \leftarrow$ RAND([0,1])
    $f^* \leftarrow \arg\max_f R_f(t)$
  $t_d \leftarrow \min\{6\{\sigma\}_{f^*}, \text{small constant}\}$
  return $\{f^*, t_d\}$

longer measurement time in order to capture the signal. As a result, the number of measurements needs to be proportional to the uncertainty in our predictions of when the signals are going to appear.

The dynamics of the pattern, i.e., the level of uncertainty, is captured by the standard deviation σ of the pattern interval τ which SpecInsight extracts in the learning phase. The bigger σ is, the more dynamic the usage pattern is. SpecInsight uses the *3-Sigma Rule* [28] to determine the dwell time t_d. The rule states that a $\pm 3\sigma$ interval centered at the mean of the distribution covers most of the cases. For example, in a Gaussian distribution, it covers 99.7% of the probabilities. More generally, for any distribution it covers at least 90%. Based on this rule, SpecInsight sets the dwell time to be $t_d = 6\sigma$.

A few points are worth noting:

- The reward function in Eq. 2 is not well defined for frequency bands with always-on usage patterns where $\mu = 0$ and for frequency bands with no signals where $\mu = \infty$ (always idle). For these frequency bands, we pick the reward function randomly between 0 and 1.
- Frequency bands with fixed-cycle signals, always-on signals, or no signals have $\sigma \approx 0$. For these bands, we set a minimum dwell time t_d such that the collected data contains at least a few time-frequency blocks.
- Some frequency bands might contain multiple patterns, where each pattern has its own μ and σ. SpecInsight randomly picks one of the usage patterns' μ and σ to calculate the reward function and the dwell time.
- In the case of fixed-cycle signals, SpecInsight is able to track the signals while sequential scanning only detects the signal with some probability. Our ability to track the signals is important in the case of fleeting periodic signals like the one in Fig. 1(c), which are very easy to miss using sequential scanning.
- Finally, SpecInsight is a best-effort system and might miss sensing deadlines if pattern dynamism in the entire spectrum is very high. In the worst case, if all

of the bands in the spectrum were equally highly dynamic, it would degrade to randomly sampling the bands but would still be no worse than sequential scanning. Fortunately, as we will show in section §8, only very few ($< 5\%$) of the patterns are highly dynamic in today's spectrum and SpecInsight works well.

Finally, a pseudocode of SpecInsight's smart scheduling algorithm is shown in Alg. 1.

6 Implementation

We implement SpecInsight on USRP software radios [8]. Since each USRP daughterboard works in a particular frequency range, we use two USRPs that simultaneously run SpecInsight: the first USRP is equipped with an SBX daughterboard, and works in the frequency range from 400 MHz to 4.4 GHz, and the second USRP is equipped with a WBX daughter-board and works in the frequency from 50 MHz to 2.2 GHz. We connect the two USRPs to the same antenna using a power splitter. We use an ultra-wideband omni-directional outdoor antenna that works from 25 MHz to 6 GHz [22].

In order to maximize the USRP capabilities, we tune the bandwidth and sampling rate to their maximum (40 MHz and 50 Ms/s). We set the two USRPs to sense non-overlapping frequency ranges, i.e, 50 MHz to 2.2 GHz and 2.2 GHz to 4.4 GHz. Each of them runs an independent version of SpecInsight's sensing algorithm, and their spectrum pattern databases are combined together. Thus, SpecInsight senses a total spectrum bandwidth of 4.35 GHz, from 50 MHz to 4.4 GHz. SpecInsight divides this spectrum into 171 bands based on the FCC spectrum allocation table [2]. For each band, it learns its spectrum patterns and schedules when to sense the band according to the algorithms in §4 and §5.

Implementing SpecInsight in realtime is challenging. SpecInsight needs to process a data stream over a Gbit/s. In order to support such high data rates, we implement all major computations using Intel's streaming SIMD extension (SSE2) instruction set, which provides instruction-level parallelization. We also use the FFTW library [3] for fast FFT implementation. Consequently, we are able to run SpecInsight in realtime on a machine with an 8-core Intel-i7 processor and 8 GB of RAM.

7 USRP Calibration

SpecInsight is not hardware specific, and can be used with various radios. The radio hardware, however, may have its own spurs, i.e., fake signals generated by hardware noise, which might be recognized by SpecInsight as patterns. Thus, when running SpecInsight on a particular hardware platform, the radio should be calibrated to identify hardware-specific spurs and eliminate them.

We calibrate the USRPs used in our prototype. All of our calibration experiments are conducted in a Faraday shield room which blocks all signals from the outside.

Calibration in the absence of signals: We put our sensing setup in the shield room, and collect measurements in the absence of any transmission. Since all active signals from the outside are blocked by the room, every received signal that is above the noise floor is a spur from the hardware. We noted two types of USRP spurs: 1) the USRP always shows power at the baseband DC frequency, 2) the time samples received during the first 10ms after power-on are corrupted. We add filters to SpecInsight to remove these spurs before running the algorithms. After adding these filters, SpecInsight does not detect any pattern in the samples collected by the USRPs in the shield room. This complies with the fact that there are no active signals in the environment, and random noise is discarded by the pattern clustering algorithm.

Calibration in the presence of transmission: USRPs do not adapt the receiver's gain with the signal power. As a result, signals whose power is higher than the ADC's maximum quantization level are clipped at the receiver. Clipping distorts the received signal and changes its frequency representation (creating harmonics). To ensure that the received signal's frequency representation matches that of the signal over the air, the receiver should be operating in its linear range without clipping.

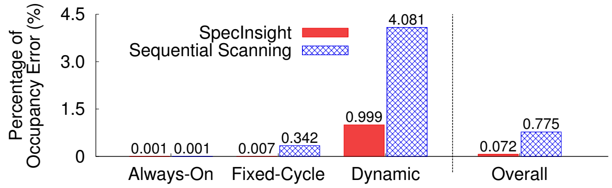
The common approach to avoid clipping is to add automatic gain control (AGC) to the receive chain [7]. USRPs however do not implement AGC. To address this issue, SpecInsight detects the occurrences of clipping by counting the number of time samples that are equal to the maximum quantization value. Once clipping is detected, SpecInsight drops the samples and sends out alerts. During our experiments, which encompass 7 locations and a total of 49 days, we noted only 7 occurrences of clipping, which were removed from the data. Please note that the clipping problem is specific to our sensing hardware but not fundamental to the algorithm; to avoid it, one could use a more expensive hardware that implements AGC.

We run experiments in the shield room with a transmitter to check SpecInsight's ability to detect a pattern correctly and eliminate clipping events. We let the transmitter transmit continuously, but vary its transmission power. We confirm that SpecInsight detects the signal in the correct frequency band as long as there is no clipping, and generates an alert whenever the signal clips.

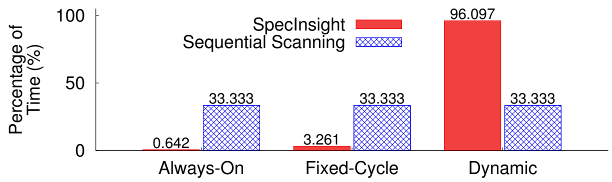
8 Empirical Results

8.1 SpecInsight's Accuracy

We compare SpecInsight with a setup that uses exactly the same USRP hardware but sequentially scans the



(a) Percentage of Occupancy Error



(b) Percentage of Time Spent in Each Class

Figure 9: Comparison of SpecInsight with Sequential Scanning: (a) shows that overall SpecInsight reduces errors by 10× in comparison to sequential scanning; (b) shows that SpecInsight uses its time wisely spending less time on always-on and fixed-cycle bands and more time on dynamic bands.

spectrum, as typical in today's systems [21]. For sequential scanning, the dwell time of each band is set to 50ms, which matches the average dwell time of SpecInsight.

To compare the accuracy of the two systems, we need the ground truth. However, existing sensing hardware does not have 4 GHz of instantaneous bandwidth thus cannot provide the ground truth for such a wideband. To address this issue, we use 10 USRPs to continuously monitor a subset of the bands within the 4 GHz spectrum, and obtain their ground truth. This provides us with the ground truth needed to calculate the accuracy of SpecInsight and sequential scanning for this particular sub-set of bands. We then repeat the experiment for different subsets of bands.

We categorize the bands based on their usage patterns to: always-on (on for $> 95\%$ of the time), fixed-cycle ($\sigma < 5$ms), and dynamic ($\sigma > 100$ms). In our experiments, we consider equal number of bands (20) of each type; for each band we run the experiment for 1 hour. For both SpecInsight and sequential scanning, we compute the following two metrics for each type of bands:

- **Percentage Occupancy Error:** This is the percentage difference between the ground truth occupancy of a band and the occupancy reported by SpecInsight and sequential scanning. We define occupancy as the percentage of time the band is occupied.
- **Percentage of Sensing Time:** This is the percentage of the total amount of time that the sensing algorithm spends in each type of band.

Results: The results using the above two metrics are shown in Fig. 9. For always-on bands, SpecInsight spends 50× less time in these bands and still achieves the same accuracy as sequential scanning. For fixed-cycle bands, SpecInsight spends 10× less time in these bands and yet has 50× higher accuracy. For bands with more dynamics, SpecInsight can afford to spend 2.5× more time in these bands which translates into 4× higher accuracy. Finally, overall, *SpecInsight has 10× higher accuracy than sequential scanning for the same time budget.* This is due to its smart scheduling algorithm, which spends as little time as needed on always-on and fixed-cycle signals, and saves its time for dynamic signals.

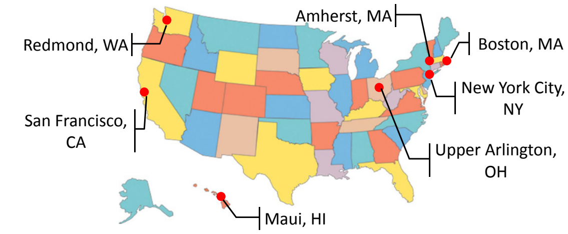


Figure 10: SpecInsight's Measurement Locations.

8.2 Real-World Spectrum Analytics

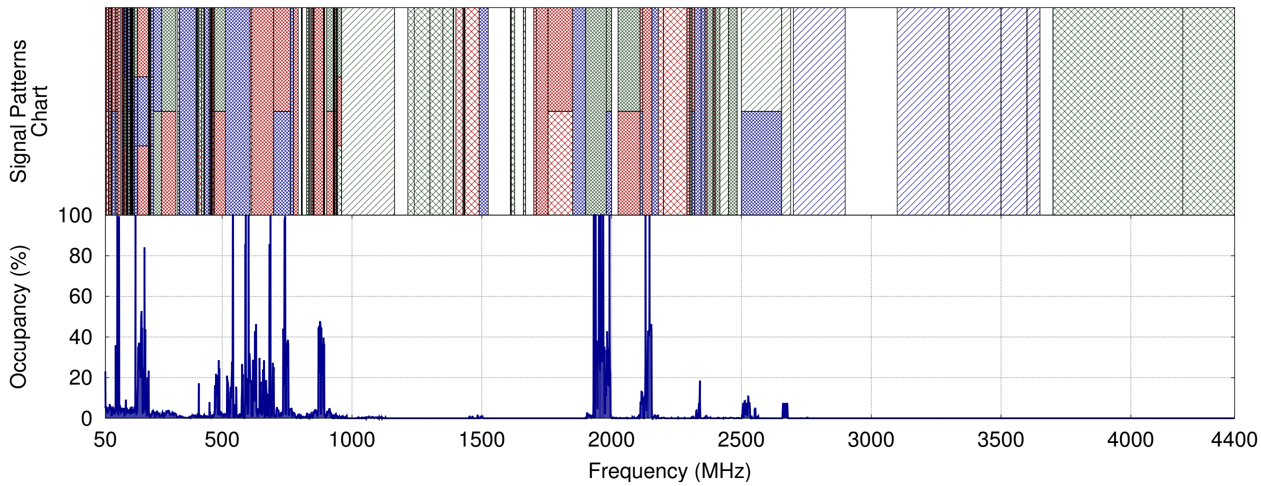
We deployed SpecInsight in seven locations in the US, including three major cities and four suburban areas, which cover the East Cost, West Cost and Pacific islands (Fig. 10). In each location, we analyzed one week of data collected by SpecInsight. We report the results below.

8.2.1 The Spectrum Pattern Chart

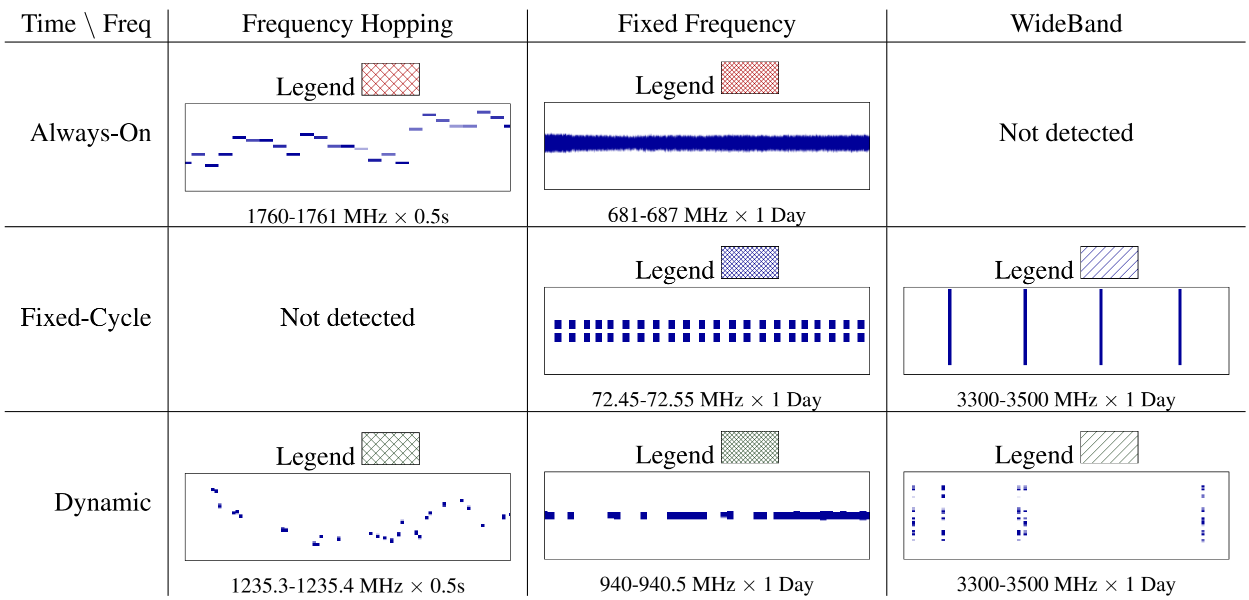
In this section, we want to analyze how the spectrum usage patterns are distributed across frequencies. Over one week and seven locations, SpecInsight detected a total of 312 different patterns corresponding to different technologies. To be able to visualize these patterns, we group them into classes according to their time and frequency properties. In the time dimension, we divide the patterns into always-on, fixed-cycle and dynamic. In the frequency dimension, we divided the patterns into frequency-hopping, fixed frequency, and wideband [6]. This gives us a total of 3×3=9 classes [7], where Fig. 11 (b) shows one usage pattern example for each class. Based on these usage patterns, we constructed the first-of-its-kind *spectrum pattern chart* shown in Fig. 11 (a). In a similar fashion to the FCC's spectrum allocation chart, the spectrum pattern chart shows the types of spectrum usage patterns seen in different frequency bands. Please note that we group the patterns into these rough classes just for the purpose of visualization; SpecInsight's database contains the exact and detailed patterns in each FCC band, in the form of time-frequency blocks.

Results: Fig. 11(a) shows the spectrum pattern chart (top) and the average spectrum occupancy chart (bottom)

[6] We label signals with bandwidth larger than 50MHz as wideband.
[7] Note in all of the experiments we did not see wideband signals that are always on, or frequency hopping signals that repeat in a fixed cycle. Hence, we ended up with a total of 7 classes.



(a) The spectrum pattern chart and average occupancy side by side: *Top*: Spectrum pattern chart drawn in the same fashion as the FCC allocation chart. Each of the small rectangle represents an active signal pattern type, out of the seven active types in Fig. 11(b). Frequency bands are arranged horizontally according to their frequency; and for bands with multiple types of patterns, the rectangles are piled up vertically. Different fillings for the rectangles represent different types of patterns (listed in Fig. 11(b)). *Bottom*: The average occupancy over 1 week and 7 locations.



(b) Legend for rectangle fillings: We divide the patterns according to their time and frequency properties, which are the rows and columns of this chart and each intersection defines a class of patterns. So there are a total of $3 \times 3 = 9$ types. We give examples for 7 types of signals, while the other two (wideband always-on and frequency-hopping fixed-cycle signals) were not detected in any of the 7 locations.

Figure 11: The Spectrum Pattern Chart

over one week and seven locations. The bottom graph is computed by averaging occupancy across locations and the top graph is a superposition of the patterns across all locations. The figure shows that although there are many bands in the occupancy chart that are empty or nearly empty, the pattern chart reveals that these bands are actually being used. For example, the occupancy in the frequency ranges 1.2 GHz–1.85 GHz and 2.9 GHz–4.4 GHz is less than 0.0001 (almost zero). However, SpecInsight detected in these bands some frequency hopping signals and some wide-band fleeting periodic signals. In fact, the figure shows that *although large swaths of the spectrum may appear completely empty, they actually have active signals*. In particular, about 39% of the bandwidth below 4.4 GHz is used by signals whose occupancy is less than 0.0001. Moreover, the usage patterns in these band are mostly of two types: 62.6% are frequency hopping signals and 33.5% are wideband fleeting signals.

To better understand how much bandwidth each type of pattern spans and how much it contributes to the spectrum occupancy, consider Fig. 12. The figure shows the distribution of bandwidths and occupancies of the patterns in government-owned bands, non-government bands and shared bands (where both government and

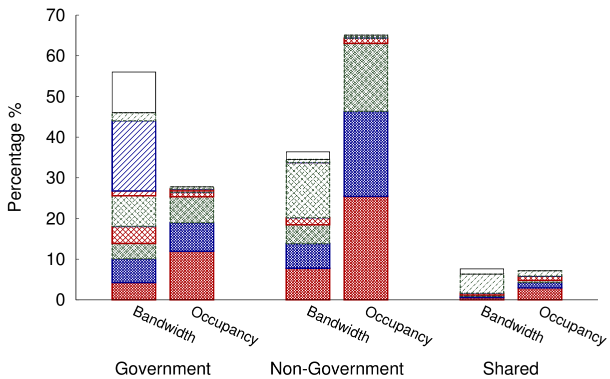


Figure 12: Spanned bandwidth vs. contributed occupancy: We check the spanned bandwidth vs. contributed occupancy for different types of patterns in government-owned, non-government and shared bands. For the legend of this figure, see Fig. 11(b).

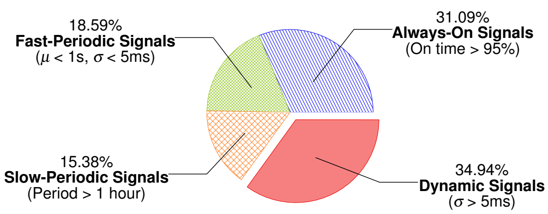


Figure 13: Statistics of Patterns According to the Timing Characteristics: The figure shows that more than half of the patterns (65%) have some timing regularities, either always-on or periodic.

non-government usage coexist). The results reveal that usage patterns like frequency hopping and wideband signals occupy 53.3% of the bandwidth but only contribute 6.8% to the total spectrum occupancy. This is more apparent in government-owned bands since these technologies are typically used in security applications. Particularly, the government owns 56% of the spectrum but only contributes 27.8% to the total occupancy.

8.2.2 *Timing Analytics*

In our timing analysis, we aim to answer the following questions: How many of the spectrum patterns are dynamic? How many are highly predictable (periodic or always-on) signals? We use the standard deviation σ of the pattern intervals (described in §4.2) to distinguish dynamic patterns from periodic and always-on signals. Often higher σ reveals a more dynamic usage pattern. However, this is not always true. Some periodic patterns have a very large period (hours-days), and hence can have a large standard deviation σ. Fig. 14 shows a usage pattern in the government-owned 152 MHz band that repeats every day. In particular, it has a signal that is always present, but at night, it is turned off in every other channel. To accommodate such periodic patterns with large σ, we distinguish between fast periodic and slow periodic.

Results: Fig. 13 shows the percentage of patterns that are always-on, fast-periodic, slow-periodic and dynamic, out

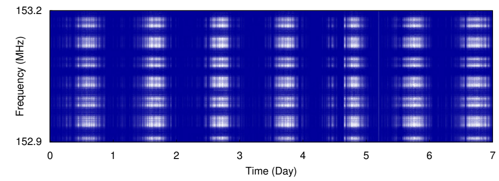


Figure 14: Example of a Slow Periodic Signal: Every other channel of the signal is turned off at night for a fixed duration.

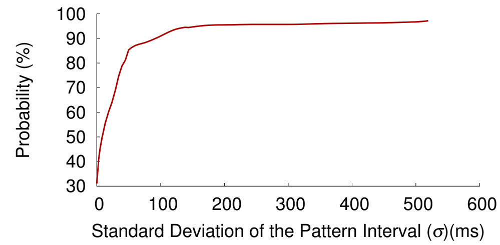


Figure 15: The CDF of the standard deviation of the pattern interval (σ). Only less than 5% of the signals have very large σ.

of the 312 detected patterns. It reveals that *only 35% of the detected usage patterns are actually dynamic*.

To gain more insight into how dynamic the frequency bands are, we compute the CDF of the standard deviation σ of signal intervals. Fig. 15 shows this CDF and reveals that *less than 5% of the patterns are highly dynamic*, i.e., having a very large σ (σ > 200ms). These results show that knowing the spectrum patterns is highly useful for smart scheduling of sensing activities, and hence the benefits of SpecInsight.

9 Conclusion

This paper presents SpecInsight, a system that can acquire the detailed utilization patterns over 4 GHz of spectrum in real time. We implement SpecInsight using off-the-shelf USRP radios and perform a large-scale study of spectrum analytics in 7 US locations including urban and suburban areas. Consequently, we build the first-of-its-kind spectrum pattern database characterizing how the spectrum is utilized. We believe that SpecInsight enables multiple applications such as dynamic spectrum access, finding breaches of spectrum regulations, and understanding undocumented spectrum utilizations.

Acknowledgments: We thank the anonymous reviewers and shepherd Jacobus Van der Merwe for their constructive comments. We are grateful to Haitham Hassanieh, Anoop Gupta, Aakanksha Chowdhery, Omid Abari, Swarun Kumar, Deepak Vasisht, John MacDonald, Fadel Adib, Hariharan Rahul, Zach Kabelac, Ezz Hamed, and Ranveer Chandra for their constructive feedbacks. We also thank the NETMIT group for their support. This research is funded by NSF. We thank members of the MIT Wireless Center: Amazon, Cisco, Google, Intel, Mediatek, Microsoft and Telefonica for their interest and general support.

References

[1] Army use of the electromagnetic spectrum. http://www.apd.army.mil/pdffiles/r5_12.pdf.

[2] United states frequency allocations. http://www.ntia.doc.gov/files/ntia/publications/2003-allochrt.pdf.

[3] FFTW. http://www.fftw.org/.

[4] Abari, O., Lim, F., Chen, F., and Stojanovic, V. Why analog-to-information converters suffer in high-bandwidth sparse signal applications. *Circuits and Systems I: Regular Papers, IEEE Transactions on* (2013).

[5] Ankerst, M., Breunig, M. M., peter Kriegel, H., and Sander, J. OPTICS: Ordering points to identify the clustering structure. In *SIGMOD* (1999), pp. 49–60.

[6] Chen, D., Yin, S., Zhang, Q., Liu, M., and Li, S. Mining spectrum usage data: a large-scale spectrum measurement study. In *Mobicom* (2009).

[7] Drentea, C. *Modern Communications Receiver Design and Technology*. Artech House, 2010.

[8] Ettus. Inc. USRP. http://ettus.com.

[9] FCC. Second memorandum opinion & order 10-174.

[10] Ghazi, B., Hassanieh, H., Indyk, P., Katabi, D., Price, E., and Shi, L. Sample-optimal average-case sparse fourier transform in two dimensions. In *Allerton* (2013).

[11] Gittins, J. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society* (1979).

[12] Hassanieh, H., Indyk, P., Katabi, D., and Price, E. Nearly optimal sparse fourier transform. In *STOC* (2012).

[13] Hassanieh, H., Shi, L., Abari, O., Hamed, E., and Katabi, D. Ghz-wide sensing and decoding using the sparse fourier transform. In *INFOCOM* (2014).

[14] Hong, S. S., and Katti, S. R. DOF: A local wireless information plane. In *ACM SIGCOMM* (2011).

[15] Islam et al., M. Spectrum survey in singapore: Occupancy measurements and analyses. In *CrownCom* (2008).

[16] Iyer, A. P., Chintalapudi, K., Navda, V., Ramjee, R., Padmanabhan, V. N., and Murthy, C. R. SpecNet: spectrum sensing sans frontières. In *NSDI* (2011).

[17] Kim, H., and Shin, K. Efficient discovery of spectrum opportunities with mac-layer sensing in cognitive radio networks. *Mobile Computing, IEEE Transactions on* (2008).

[18] Kone, V., Yang, L., Yang, X., Zhao, B. Y., and Zheng, H. On the feasibility of effective opportunistic spectrum access. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement 2010, Melbourne, Australia - November 1-3, 2010* (2010).

[19] Lakshminarayanan, K., Sapra, S., Seshan, S., and Steenkiste, P. RFDump: An architecture for monitoring the wireless ether. In *CoNEXT* (2009).

[20] McHenry, M. A. NSF spectrum occupancy measurement project summary, 2005.

[21] Microsoft Spectrum Observatory. http://spectrum-observatory.cloudapp.net.

[22] MP. Ultra base station antenna. http://www.mpantenna.com/products/product/08-ant-0861-vhf-uhf-basestation-antenna/.

[23] Rahul, H., Kushman, N., Katabi, D., Sodini, C., and Edalat, F. Learning to share: Narrowband-friendly wideband networks. In *ACM SIGCOMM* (2008).

[24] Rashidi, M., Haghighi, K., Panahi, A., and Viberg, M. A NLLS based sub-nyquist rate spectrum sensing for wideband cognitive radio. In *IEEE DySpan* (2011).

[25] Rayanchu, S., Patro, A., and Banerjee, S. Airshark: Detecting non-wifi rf devices using commodity wifi hardware. In *ACM IMC* (2011).

[26] Taher, T., Bacchus, R., Zdunek, K., and Roberson, D. Long-term spectral occupancy findings in chicago. In *IEEE DySPAN* (2011), pp. 100 – 107.

[27] Tian, Z., and Giannakis, G. A wavelet approach to wideband spectrum sensing for cognitive radios. In *Cognitive Radio Oriented Wireless Networks and Communications, 2006. 1st International Conference on* (2006).

[28] Triola, M. F. *Essentials of Statistics*. 2009.

[29] PCAST: Realizing the full potential of government held spectrum to spur economic growth, 2012.

[30] Vermorel, J., and Mohri, M. Multi-armed bandit algorithms and empirical evaluation. In *ECML* (2005).

[31] Yang, L., Hou, W., Cao, L., Zhao, B. Y., and Zheng, H. Supporting demanding wireless applications with frequency-agile radios. In *NSDI* (2010).

[32] Ying, X., Zhang, J., Yan, L., Zhang, G., Chen, M., and Chandra, R. Exploring indoor white spaces in metropolises. In *ACM MobiCom* (2013).

[33] Yoon, S., Li, L. E., Liew, S., Choudhury, R. R., Tan, K., and Rhee, I. Quicksense: Fast and energy-efficient channel sensing for dynamic spectrum access wireless networks. In *IEEE INFOCOM* (2013).

[34] Yucek, T., and Arslan, H. A survey of spectrum sensing algorithms for cognitive radio applications. *Communications Surveys Tutorials, IEEE 11*, 1 (2009).

[35] Zhang, T., Leng, N., and Banerjee, S. A vehicle-based measurement framework for enhancing whitespace spectrum databases. In *Mobicom* (2014).

[36] Zhao, Q., Tong, L., Swami, A., and Chen, Y. Decentralized cognitive mac for opportunistic spectrum access in ad hoc networks: A POMDP framework. *Selected Areas in Communications, IEEE Journal on 25*, 3 (2007).