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

Open Access to the Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI ’15) is sponsored by USENIX
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 $2 \times 10^{-5}$.

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
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 dynamic2. Thus, knowing the spec-

---

1The use of two radios is not fundamental to our design but rather imposed by the range of frequencies of the USRP daughterboards.
2Defined as having a standard deviation of when the signal will next appear that exceeds 200ms.
analog filters. Additionally, SpecInsight covers a wider in that it does not need sparsity assumptions or custom
grology. SpecInsight, however, differs from the above work mostly documented.
hundreds of MHz where the wireless technologies are
Sense [33], employs a hierarchical search algorithm and
spectrum is occupied, so it cannot be used in crowded
a sparsity assumption that only a small fraction of the
able to recover the full signals in the spectrum, but under
sparse FFT [12, 10, 13]. For example, BigBand [13] is
exploit the sparsity of spectrum utilization to sense the
sensing techniques try to cover a wide spectrum signifi-
cost and easily accessible USRP radios.
using energy level [31], cyclo-
variations of these patterns in realtime over a wide
broadband mounting spectrum sensors on public vehicles and lever-
ages mobility to enable large-area sensing of the white
SpecInsight complements these systems by en-
ning multi-GHz spectrum sensing on relatively low-
cost and easily accessible USRP radios.

Another line of work focuses on collaborative sens-
ing, where different nodes share spectrum data in order
to cover a large geographical area. For example, Spec-
Net [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 lever-
ages mobility to enable large-area sensing of the white
SpecInsight complements these systems by en-
able

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 in-
ferers [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. Addition-
ally, 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 undocu-
mented.

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 be-
lieve 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,
What are the patterns? Spectrum patterns are a key concept in SpecInsight’s design. A pattern is a representative frequency-time 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 duration times. 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-
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.

---

\(^3\)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.
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 $\Delta f$ and $\Delta t$, we compute it only for a small range of $\Delta f$ and $\Delta t$. 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 $B_3$ and $B_4$ 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 $\tau$ 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 $\tau$. It can be characterized by its mean $\mu$ and standard deviation $\sigma$, which SpecInsight computes over multiple measurements.

These statistics $\mu$ and $\sigma$ are necessary to sense the spectrum efficiently. The mean $\mu$ determines the period of the pattern, and the standard deviation $\sigma$ measures how dynamic the signal is. Thus, $\mu$ can be used to decide
how often and at what time we expect to see the signal and $\sigma$ 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 ($\sigma$ 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 $\varepsilon$-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 $\varepsilon$ of the time she picks a lever at random. The choice
The dwell time is determined by \( \tau \), which SpecInsight extracts in the learning phase. The bigger \( \sigma \) is, the more dynamic the usage pattern is, for the following reason: The reward function in Eq. 2 is not well defined for frequency bands with no signals where \( \mu = 0 \) and for frequency bands with no signals where \( \mu = \infty \) (always idle). For these frequency bands, we set a minimum dwell time \( t_d \) such that the collected data contains at least a few time-frequency blocks.

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 \( \mu \) and \( \sigma \). SpecInsight randomly picks one of the usage patterns’ \( \mu \) and \( \sigma \) 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

\[ R_f(t) = 1 - \frac{T + \mu \left(\frac{t - T}{\mu}\right) - t}{\mu} \]  

where \( T \) is the last time the signal was observed and \( \mu \) 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 \( \mu \).

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 \( \mu \), we predict the signal will appear again at times \( T + \mu, T + 2\mu, T + 3\mu, \ldots \). Thus at time \( t \), we predict that the signal will appear next at time \( T + \mu \left(\frac{t - T}{\mu}\right) \) and we are \( T + \mu \left(\frac{t - T}{\mu}\right) - t \) away from it. Since the farthest we can be away from the next appearance is \( \mu \), we normalize by \( \mu \) 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 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 \( \sigma \) of the pattern interval \( \tau \) which SpecInsight extracts in the learning phase. The bigger \( \sigma \) 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 \( \mu \) and \( \sigma \). SpecInsight randomly picks one of the usage patterns’ \( \mu \) and \( \sigma \) 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 10 ms 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
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.

To get the ground truth, we collected by SpecInsight. We report the results below.

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. This gives us a total of $3 \times 3 = 9$ classes, 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: The results using the above two metrics are shown in Fig. 9. For always-on bands, SpecInsight spends $50 \times$ less time in these bands and still achieves the same accuracy as sequential scanning. For fixed-cycle bands, SpecInsight spends $10 \times$ less time in these bands and yet has $50 \times$ higher accuracy. For bands with more dynamics, SpecInsight can afford to spend $2.5 \times$ more time in these bands which translates into $4 \times$ higher accuracy. Finally, overall, SpecInsight has $10 \times$ 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.](image)

![Figure 9: Comparison of SpecInsight with Sequential Scanning](image)
(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 \( \sigma \) of the pattern intervals (described in §4.2) to distinguish dynamic patterns from periodic and always-on signals. Often higher \( \sigma \) 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 \( \sigma \). 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 \( \sigma \), 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 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 \( \sigma \) 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 \( \sigma \) (\( \sigma > 200 \text{ms} \)). 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 utilisations.

Acknowledgments: We thank the anonymous reviewers and shepherd Jacobus Van der Merwe for their constructive comments. We are grateful to Haitham Hasanieh, Anoop Gupta, Aakanksha Chowdhery, Omid Abari, Swarun Kumar, Deepak Vasisht, John MacDonald, Fadel Adib, Harirahan 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


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


