QuickSense: Fast and Energy-Efficient Channel Sensing for Dynamic Spectrum Access Networks

Sungro Yoon†, Li Erran Li⋆, Soung Chang Liew∗,
Romit Roy Choudhury‡, Injong Rhee†, Kun Tan◦
North Carolina State University† Bell Laboratories⋆
Chinese University of Hong Kong∗ Duke University‡ Microsoft Research Asia◦

***Abstract*—Spectrum sensing, the task of discovering spectrum usage at a given location, is a fundamental problem in dynamic spectrum access networks. While sensing in narrow spectrum bands is well studied in previous work, wideband spectrum sensing is challenging since a wideband radio is generally too expensive and power consuming for mobile devices. Sequential scan, on the other hand, can be very slow if the wide spectrum band contains many narrow channels. In this paper, we propose an analog-filter based spectrum sensing technique, which is much faster than sequential scan and much cheaper than using a wideband radio. The key insight is that, if the sum of energy on a contiguous band is low, we can conclude that all channels in this band are clear with just one measurement. Based on this insight, we design an intelligent search algorithm to minimize the number of total measurements. We prove that the algorithm has the same asymptotic complexity as compressed sensing while our design is much simpler and easily implementable in the real hardware. We show the availability of our technique using hardware devices that include analog filters and analog energy detectors. Our extensive evaluation using real TV "white space" signals shows the effectiveness of our technique.**

I. Introduction

The exponential growth of mobile data is a major challenge to the operators of cellular networks, who expect that mobile data in their networks will grow 18-fold by 2016 [1]. As a result, many operators face a severe spectrum shortage, dubbed a spectrum crunch [2], if the problem is left unaddressed. Aware of and keen to resolve the problem, the US government has set the goal of freeing up 500 MHz of spectrum for wireless communication [3]. A large portion of this will have to come from incumbents. Some incumbents may only allow their spectrum to be used while they are not using it. This demands innovative dynamic spectrum allocation and access.

One fundamental problem in dynamic spectrum access (DSA) networks is to obtain the spectrum map, the occupancy of all channels and their usage overtime, as shown in Figure 1. The spectrum map provides DSA devices a global view on entire frequencies, thus contributing to the best selection of channels. For example, a DSA device in need of a reliable connection can pick a channel that has been the least interfered in the past. Another device, that prefers more link bandwidth, can try channel bonding [4] and choose a channel with more available neighboring frequencies. There are two known approaches to building the spectrum map. One is to query a geo-location database on occupied channels in a particular location; the other is to let DSA devices sense a wide band of radio frequency and survey the occupancy.

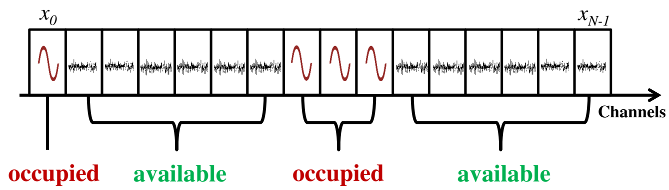

Fig. 1. Wideband spectrum sensing problem: determine $X = [x_0, x_1, \cdots, x_{N-1}]$ where $x_i = 1$ if Channel i is occupied, and 0 otherwise.

Spectrum sensing is an essential technique for DSA networks. Even though the FCC has mandated the geo-location database approach for secondary users [5], spectrum sensing helps in the following settings. First, the current geo-location database does not provide any information about other secondary users within the area. Thus, there are no means to arbitrate among multiple DSA devices in the same location. Second, the geo-location database can only approximately predict the signal reception quality at a location and could be inaccurate. While the geo-location database approach relies on certain propagation models for the prediction, e.g., [6], due to the complexity of radio propagation and the changing environment, the prediction model might not provide fine information. Finally, accurate and fast spectrum sensing technologies can help increase the spectrum efficiency. For example, a secondary user can opportunistically use a channel during the time interval between primary users' transmissions when a short silence is detected. Recently, President's Council of Advisors on Science and Technology (PCAST) has published a report in which use of the spectrum sensing is recommended [7]. Standard bodies such as 802.22 [8], IEEE SCC41, and companies [9] have already started working on such spectrum sensing technologies.

The channel sensing is challenging because the entire search space is very large. Recent measurements [10] reveal that, in the band between 30MHz and 3GHz, there are at least several hundred spectrum holes, each of which more than 2MHz wide. In the future many of these spectrum holes will be subject to dynamic spectrum access.

However the current spectrum sensing techniques have clear limitations. One is the sequential scan approach, e.g., used in WiFi, that investigates all of the narrow band channels one by one, trying to detect the signal energy or the existence of specific signatures [11]. But this approach is not scalable as it always requires N searches, where N is the number of channels. Another possible approach is to perform wideband scan using an advanced device such as high performance analog-to-digital converters (ADCs). But the high speed ADC is very expensive, power consuming and requires extensive computation for the signal processing [12]. For example, in order to detect 500MHz bandwidth 1 Giga samples per second ADC is required that consumes 2.15W and costs $775 [13]. Thus this wideband scan is not suitable for mobile devices that have limited energy. Recent proposals adopt the compressed sensing technique for wideband spectrum sensing [14], [15]. These proposals perform sub-Nyquist sampling and achieve $O(k \cdot logN)$ searches where k is the sparsity of the spectrum. However, prior knowledge of k is usually required, which is not feasible in DSA networks where mobile devices frequently join and leave. In addition, the compressed sensing requires dedicated hardware architectures [16], [17]. To the best of our knowledge, there is no low-cost implementation of the compressed wideband sensing.

In this paper, we propose a fast, low-cost and energy-efficient way of wideband sensing. The key insight is that consecutively located clear channels produce nearly zero signal energy, regardless of their total bandwidth. Therefore, it is not necessary to scan through all the narrow bands one by one within the clear channels. It just suffices to probe the energy level of an entire target bandwidth with one time measurement. One can skip looking into that bandwidth if no significance in the energy has been observed. Only if some energy has been detected, do we zoom in to measure energy at finer granularities. We deliberately choose analog hardware as the basic building block for our technique. Low-pass filters are used to adjust the target search bandwidth: the wideband signal filtered through the analog filter is separated at various sub-bands. Then we use energy detectors to measure the energy of each sub-band. We refer to our solution as QuickSense.

QuickSense builds upon an intelligent algorithm and the accompanying hardware design. We show that QuickSense achieves $O(k \cdot logN)$ searches on average, which is asymptotically the same searching speed as compressed sensing, and N searches in the worst case. But unlike compressed sensing, QuickSense is easily implementable and operates at any sparsity of the spectrum. We test the availability of our hardware design via the proof-of-concept experiment where off-the-shelf hardware is used. It shows that QuickSense can perform an efficient spectrum sensing in the real-world setups. We collect TV signals from UHF whitespaces and perform extensive trace-driven simulations to evaluate the performance. The results show that our technique can perform up to 3.4 times and 1.5 times fast searching when 10% and 30% of the channels are occupied, respectively. This saves up to 70.6% and 33.3% energy.

The rest of the paper is organized as follows. In Section II, we introduce our QuickSense algorithms and the hardware designs. In Section III, we discuss several additional issues. In Section IV, we present the proof of concept experiment and the trace-driven simulation. We discuss related work in V and conclude this paper in Section VI.

II. QuickSense: Analog-filter-based Spectrum Sensing

In this section, we first define wideband spectrum sensing problem. We then propose a basic algorithm that uses a single tunable analog filter and an analog energy detector. After that, we present an enhanced algorithm that uses $\log_2(N)$ number of fixed-bandwidth analog filters and analog energy detectors. For each algorithm, we provide the matching hardware design.

A. Problem Statement

Consider a spectrum consisting of N channels (subcarriers) each having fixed bandwidth as shown in Figure 1. Let variable $x_i \in \{0,1\}$ be an indicator that specifies whether Channel i is occupied or not. We first consider the basic problem of determining all the values of x_i, $\forall i = 0,1,\cdots,N-1$.

Let P_i be the power sensed on channel i. Let Th_i be the energy detection threshold on channel i. We have $x_i = 0$ if $P_i < Th_i$, 1 otherwise. The threshold Th_i can be set according to [18]. The noise power spectral density needed does not change very fast, so we can obtain this through infrequent measurements. We will look into the threshold setting in Section II-D.

Let f_i denote the left end of the frequency band of channel i. We have a frequency filter that we can position over the interval $[f_i,\ f_{i+b}]$ to separate the signal within the frequency band. We can set i and b arbitrarily.

After filtering and measurement, we can then obtain the signal power within the band: $P_{[i,i+b-1]} = P_i + P_{i+1} + \cdots + P_{i+b-1}$. Let $Th_{[i,i+b-1]}$ be the corresponding energy detection threshold. We can decide as follows: if $P_{[i,i+b-1]} < Th_{[i,i+b-1]}$, then $x_j = 0$ for all $j \in i,\cdots,i+b-1$. Otherwise if $P_{[i,i+b-1]} > Th_{[i,i+b-1]}$, then $x_j = 1$ for one or more $j \in \{i,\cdots,i+b-1\}$.

For the second case, all we know is that there exists at least one occupied channel between 1 and b. In particular, the transmitting stations may be at different distances to our detector so the value of $P_{[i,i+b-1]}$ does not tell us how many stations are transmitting. By using the filter to make a sequence of successive measurements, we want to derive x_i for all $i \in 0,\cdots,N-1$. Our goal is to minimize the number of measurements needed.

In general, there are 2^N possible combinations for $X = (x_0,x_1,\cdots,x_{N-1})$. This means that N bits of information is needed to describe the spectrum usage profile. Thus, to derive the profile information, N measurements will be necessary in general. This is due to the fact that each measurement gives one bit of information, either $x_j = 0$ or $x_j = 1$. However, as indicated by work in the arena of compressed sensing, when $Z = [P_0,P_1,\cdots,P_{N-1}]$ is sparse (i.e., there are many low energy readings in the vector), we can do much better. In particular, if we can make a series of measurements consisting of linear combination of P, as embodied by $AZ = Y$ where A is a matrix representing m linear combinations in m measurements Y, then it is only necessary for m to be of order $k\log_2(N/k)$, provided that Z is k-sparse (i.e., there are only k readings in Z that are above noise power).

The problem is that we do not have a ready way to implement the measurement matrix A, so we cannot fit our problem within the above traditional framework of compressed sensing. In the worst case, we may end up having to make order N filtered measurements all the time just to construct the linear combinations.

The issue is whether we can construct an alternative algorithm in which the number of measurements is $O(N)$ in the worst case, yet $O(k\log_2(N/k))$ when the vector X is sparse. In particular, we would like the algorithm to be robust and near optimal for different levels of sparsity.

B. Basic Algorithm

We now outline a basic algorithm based on a combination of linear search and bisection search, which is linear and logarithmic in complexity, respectively.

This basic design is based on a specific hardware component, tunable channel filter (tunable in terms of bandwidth) [19]. For now we assume that the tunable filter is available and it can completely separate out the only channels of interest. Then an energy detector can be used for the filter output to determine if a certain bandwidth is occupied.

The basic idea is that when $P_{[i,i+b-1]}$ is below a threshold, we can deduce in one shot that $x_i,\cdots,x_{i+b-1}=0$. Otherwise when $P_{[i,i+b-1]}$ is above a threshold, we do not have much information. In this algorithm, the linear part consists of the identification of contiguous x_i that are all 1's; and the logarithmic part consists of the identification of contiguous 0's. With many zeroes (the sparse case), the logarithmic part begins to dominate and the algorithm becomes efficient.

In more detail, as shown in Figure 2, the algorithm starts to find the first band of contiguous 0's. If $x_0 = 0$, then the algorithm measures $P_{[1,2]}$. If $P_{[1,2]}$ is small, the algorithm doubles the measurement interval again to measure $P_{[3,6]}$. If $P_{[3,6]}$ is greater than noise power, then the algorithm contracts the measurement interval by half to measure $P_{[3,4]}$ and so on. This binary search procedure is implemented by a recursive function $identifyZeroes(i,d)$. The function returns an integer j, such that $x_i,\cdots,x_{i+j-1}$ are all zeroes and $x_{i+j}=1$. If the returned $j=0$, then the implication is that $x_i=1$, and no zeroes are identified. The parameter d is such that $[i,i+d-1]$ is the initial interval to be explored. The returned j, however, can be less than or equal to d.

Once the algorithm identifies the first 1 after the first contiguous 0's, it begins the linear search to identify the first contiguous 1's. This is done by the function $identifyOnes(i)$. The function returns an integer j, such that $x_i,\cdots,x_{i+j-1}$ are all 1's and $x_{i+j}=0$. If the returned $j=0$, then the implication is that $x_i=0$, and no 1's are identified.

Example: As shown in Figure 3, suppose $N=64$ and only the channel 15 is occupied. The algorithm first identifies contiguous 0's by measuring P_0, $P_{[1,2]}$, $P_{[3,6]}$, $P_{[7,14]}$, $P_{[15,30]}$. Since $P_{[15,30]}$ is significant, the algorithm performs binary search for the occupied channel by shrinking its filter size and measures $P_{[15,22]}, P_{[15,18]}, P_{[15,16]}$ and finally P_{15}. Now it transitions to identify continuous 1's by measuring P_{15} and P_{16}.

```
initialize i,j to zero and the spectrum profile X to all zero,
    N is the total number of channels in the band.
    Measure the noise power and set the threshold value.

while (i < N) {
    j=identifyZeroes(i,1);
    set X[i,i+j-1] = 0 and x_{i+j} = 1;
    i = i+j+1;
    if (i < N) {
        j=identifyOnes(i);
        set X[i,i+j-1] = 1 and x_i = 0;
        i = i+j+1;
    }
}


// linear search to find X[i,i+j-1] with consecutive 1s
identifyOnes(i) {
    int j = 0;
    while (i+j < N) {
        measure P_{i+j};
        if P_{i+j} < Threshold, break;
        else j++;
    }
    return j;
}


// binary search to find X[i,i+j-1] with consecutive 0s
identifyZeroes(i,d) {
    int j = 0;
    if (i+d > N) d = N-i;
    measure P_{[i,i+d-1]};
    if (P_{[i,i+d-1]} ≥ Threshold) {
        if (d/2 ≥ 1)
            j = identifyZeroes(i, d/2);
        else    j = 0;
    }
    else j = d + identifyZeroes(i+d; 2*d);
    return j;
}
```

Fig. 2. QuickSense Basic Algorithm

Since channel 16 is empty, the algorithm again transitions to identify 0's by measuring P_{17}, $P_{[18,19]}$, $P_{[20,23]}$, $P_{[24,31]}$, $P_{[32,47]}$, and $P_{[48,63]}$. The number of measurements is 13 as compared to 64 for the linear scan. We show in Section II-B that our algorithm takes $O(k\log_2(\frac{N}{k}))$ measurements when the number of occupied channels k is small.

Hardware design: Figure 4 illustrates the schematic design of QuickSense. The basic idea of our hardware design is to bring the wideband signal down to baseband using a voltage controlled oscillator (VCO) and a mixer. Depending on the frequency band, we might need a preselector to reject out-of-band unwanted interference signals. Once we have the signal in the baseband, we can pass it through a tunable lowpass filter. The tunable lowpass filter then filters the signal at the target bandwidth of interest. The filtered signal is delivered to an analog energy detector (simply consisting of a squarer, amplifier, and an integrator), which determines the signal energy level. Then the control unit digitizes the energy reading

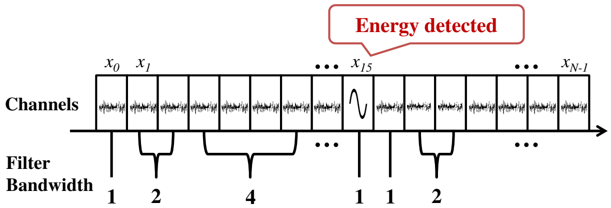

Fig. 3. QuickSense doubles the filtering bandwidth until it detects an occupied channel. Then it shrinks its filtering bandwidth to find the occupied channel. Once it finds an empty channel, it increases the filtering bandwidth again.

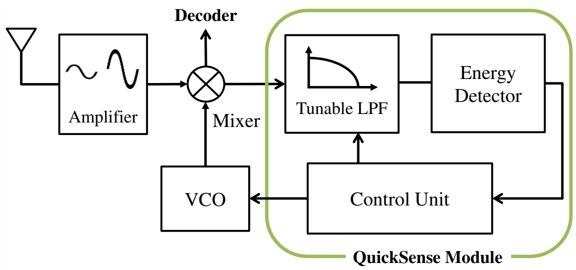

Fig. 4. Basic QuickSense design using a single pair of a tunable lowpass filter and an energy detector.

(using a very low-rate ADC) which is then used as an input to the implementation of our QuickSense algorithms as described in Sections II-B and II-C.

To improve component reuse, QuickSense can be designed to share the antenna and the amplifier with the DSA radio. During reception of DSA radio, a copy of the incoming analog signal is split and fed into the QuickSense module for building the spectrum map. During transmission of DSA radio, QuickSense module can be turned off. The QuickSense module works alongside of DSA radio and does not interfere with DSA transmission or reception. It can provide spectrum occupancy profile information to applications or the spectrum management layer [20] which in turn can instruct DSA radio to select appropriate channels to use.

Complexity Analysis: We consider a few extreme cases first. When X is all 1's, the algorithm is $\Theta(N)$. When X consists of alternating 1's and 0's, the algorithm is also $\Theta(N)$.

Suppose X is k-sparse. Let where $k = \rho N$ where $0 \leq \rho \leq 1$. Let us consider an extreme case where the k non-zero x_i are from 0 to $k-1$. Then, the number of measurements is $c_1 k + c_2 \log_2[(1-\rho)N]$ for some constants c_1 and c_2. The term $c_1 k$ is the linear part for identification of the 1's; the term $c_2 \log_2((1-\rho)N)$ is for the identification of the 0's. This is however not the worst case given sparsity.

The worst case is where the k 1's of x_i are not contiguous so that there are either $k-1$, k, or $k+1$ sections of contiguous 0's. Note that given sparsity k, it is not possible to have more than $k+1$ sections of contiguous zero x_i. Let l_h denote the length of section h. The complexity in this case is $\sum_h c_2 \log_2(l_h)$ subject to $\sum_h l_h = (1-\rho)N$. Because of the concavity of log function, the maximum $\sum_h c_2 \log_2(l_h)$ is obtained when the l_h's are equal for all h. Therefore, $\max \sum_h c_2 \log_2(l_h) = c_2(k+1)\log_2(\frac{(1-\rho)N}{k+1})$.

01. initialize i,j to zero and the spectrum profile X to all zero,
02. N is the total number of channels in the band.
03. Measure the noise power and set the threshold value.

04. **while** $(i < N)$ {
05. j=identifyZeroes(i);
06. set 0 to $X[i, i+j-1]$ and 1 to x_{i+j};
07. $i = i+j+1$;
08. **if** $(i < N)$ {
09. j=identifyOnes(i);
10. set 1 to $X[i, i+j-1]$ and 0 to x_i;
11. $i = i+j+1$;
12. }
13. }

// same as the basic QuickSence algorithm
14. identifyOnes(i)

// binary search to find $X[i, i+j-1]$ with consecutive 0s
15. identifyZeroes(i) {
16. **int** $j = 0$;
17. **if** $(i+d > N)$ $d = N-i$;
18. BW_{Max}: The maximum bandwidth to be sensed.
19. measure $P_{[i]}$, $P_{[i,i+(2^1-1)]}$, $P_{[i,(i+2^2-1)]}$,...$P_{[i,i+BW_{Max}]}$;
20. **if** $(P_{[i,i+2^k-1]} \geq$ Threshold for $0 \leq k \leq log(BW_{Max})$ {
21. $j = k$;
22. }
23. **else** $j = BW_{max}$ + identifyZeroes($i + BW_{Max} + 1$);
24. **return** j;
25. }

Fig. 5. QuickSense Enhanced Algorithm

Thus, the complexity is the same as that in traditional compressed sensing, except that there is a simple implementation for our algorithm, whereas directly applying the compressed sensing framework requires a high-rate ADC and complex signal reconstruction algorithms.

C. Enhanced Algorithm using Fixed Bandwidth Analog Filters

Performing the search using the single pair of an analog filter and an energy detector leads to exhaustive search when the spectrum is densely populated. We advance our design to deal with such a case. We carefully combine $log_2 N$ number of fixed-bandwidth filters and energy detectors such that they work in parallel and prevent the exhaustive search. Note that the lowpass tunable filter in the basic design is required to filter only $\log_2(N)$ different bandwidths, i.e. $1, 2, \cdots, \log_2(N)$ times a single channel. Thus, we replace the single tunable lowpass filter with a $\log_2(N)$ number of fixed lowpass filters. With the $\log_2(N)$ energy readings per each measurement, we update entries in the spectrum map in parallel (see line 18 in Figure 5). When the spectrum is sparse, this enhanced design achieves $O(loglogN)$ searches. The proof is intuitive so we omit here due to the space limit.

Hardware design: The down-converted signal output from the mixer is fed into an RF splitter. This RF splitter directs the RF signal to $log_2(N)$ fixed-bandwidth lowpass filters. $log_2(N)$ energy detectors are used to get the energy readings out of each filter. Figure 6 shows the schematic design of our enhanced

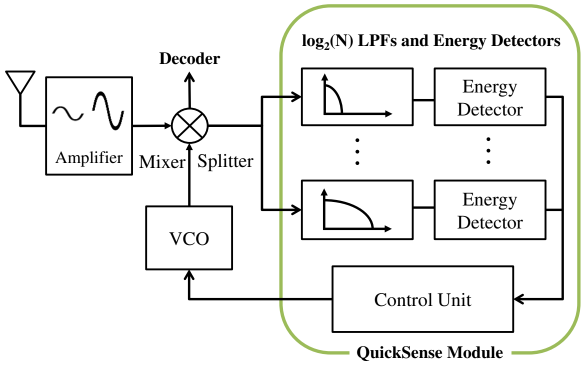

Fig. 6. Enhanced QuickSense design using $\log_2(N)$ lowpass filters with different fixed bandwidths and energy detectors.

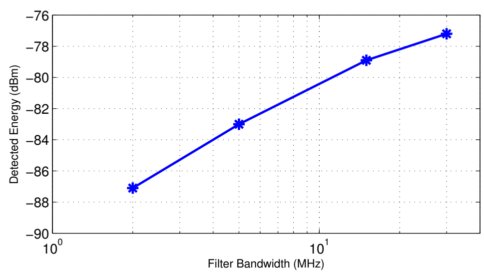

Fig. 7. Energy readings from empty channels. Filters with 2MHz, 5MHz, 15MHz and 32MHz bandwidths are used.

QuickSense algorithm.

One possible concern is that the number of channels could be hundreds or thousands. But only $log_2 N$ devices are needed (i.e. 10 devices when detecting 1,000 channels). Also the cost of off-the-shelf analog filters and energy detectors is already very cheap. For example, AD8310, a high performance energy detector, costs only $4.62. Costs of lowpass filters vary depending on their filtering bandwidths, but they are usually less than $5 (e.g., MiniCircuit's LFCN-80 is $3.24).

D. Energy Detection Threshold

QuickSense detects energy from variable-bandwidth channels. It is needed to deal with varying noise power because of this variable bandwidth. Now $P_{[j,k]}$, the energy sensed within channel $[j,k]$ is given by $P_{[j,k]} = S_{[j,k]} + \Omega_{[j,k]}$, where $S_{[j,k]}$ is the sum of signal energy and $\Omega_{[j,k]}$ is the background noise within the channel $[j,k]$, respectively. The noise energy $\Omega_{[j,k]} = \sum_{i=j}^{k} \Omega_i$ grows monotonically with the filtering bandwidth, $k-j+1$. With the larger filter bandwidth, an energy detector will naturally detect more energy due to the noise power. We measure the noise power in empty channels using a spectrum analyzer with different lowpass filters. Figure 7 shows the increasing noise power with larger filtering bandwidths.

Our idea is to learn about Ω_j for all $0 \le j < N$ before the actual channel sensing operation. Then we set $\Omega_{[j,k]}$ as the energy detection threshold for the channel $[j,k]$. As the noise power varies depending on each frequency, we use the data structure which we call *noise map*. The noise map stores the noise power Ω_j for $0 \le j < N$. The noise measurement can be done infrequently as the noise power within a frequency band changes slowly. But for building an accurate noise map, QuickSense needs to gather enough number of energy readings per noise measurement. This is due to the randomness of the noise. In the following, we show analytically how it works.

Let M denote the total number of energy readings and s_j the received signal sample at channel j where $s_j = x_j + \omega_j$. Here, x_j and ω_j are complex variables representing the received signal and the noise, respectively. $E[P_j]$, the average energy reading at channel j is given as follows:

$$\begin{aligned} E[P_j] &= \frac{1}{M}\sum^{M}|s_j|^2 = \frac{1}{M}\sum^{M}|x_j+\omega_j|^2 \\ &= \frac{1}{M}\sum^{M}(|x_j|^2+|\omega_j|^2+x_j'\omega_j+x_j\omega_j'). \end{aligned}$$

Assume M is sufficiently large such that $\frac{1}{M}\sum^{M}|\omega_j|^2 = \Omega_j$. As the two random variables x_j and ω_j are mutually independent and $E[\omega] \simeq 0$ in AWGN channel, $\sum x \cdot \omega = E[x]E[\omega] \simeq 0$. Thus we have

$$E[P_j] = \frac{1}{M}\sum^{M}|x_j|^2 + \Omega_j = S_j + \Omega_j.$$

Now the energy reading $P_{[j,k]}$ will be given by

$$P_{[j,k]} = \sum_{i=j}^{k} E[P_i] = S_{[j,k]} + \Omega_{[j,k]}.$$

If $\Omega_{[j,k]}$ is known from the noise map, we can detect the signal energy $S_{[j,k]}$ accurately. QuickSense assumes that a channel is occupied if $P_{[j,k]} > \Omega_{[j,k]}$. Otherwise the channel is assumed to be clear. In the above, M is tightly related to the energy detection accuracy. In Section IV-A2, we experimentally observe the relationship between M and the energy detection accuracy. We verify how many energy readings QuickSense needs to achieve the accurate sensing.

III. Discussions

Non-uniform channel width: Each channel might have a different bandwidth. Our basic algorithm in Figure 2 does not make any assumption on channel width. So it works with no change. However, enhanced QuickSense design, which uses $\log_2(N)$ analog filters, is no longer valid in this case. It is required to replace the fixed analog filters with tunable filters.

False negative or false positive detections: QuickSense basically relies on energy detectors to determine the occupancy of channels. Here the energy detection threshold establishes a trade-off between false positive and false negative detections. In our opinion the false negative is more harmful than the false positive, as it can interfere with ongoing primary signals. So the basic policy is to set the threshold conservatively such that the false negative detection is minimized.

Still, false negatives could be incurred in case primary signals are very weak, thus letting a device attempt to use an already occupied channel. In this case accurate narrow band sensing can be used [11]. These methods are complex and require the knowledge of underlying signal structures. But to make sure that a channel is truly empty, it is preferable to

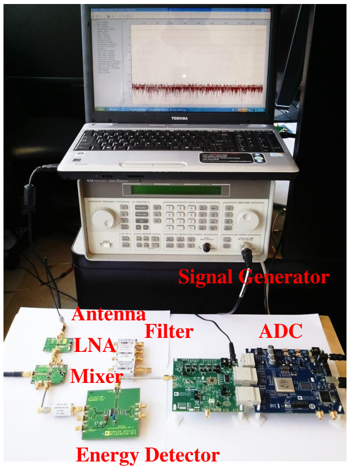

Fig. 8. QuickSense benchmark device consists of analog hardware and a signal generator. An ADC is used to read the energy detector output.

perform one of the narrow band sensing right before the actual transmissions, once the target band of use is decided.

Energy from adjacent channels: An analog filter does not completely remove out of band frequencies. This residual energy from adjacent channels might increase overall energy readings thus incurring false positives. Note that the steepness of the filtering function is defined by the roll-off factor and the value is accurately known a priori. Using the knowledge, it is possible to calibrate the energy reading values such that the false positive due to the residual energy is minimized.

Forbidden channels: Certain channels within a wide band might not allow dynamic access, e.g., due to government policies. We can handle this straightforwardly, by changing lines 15 and 24 of the algorithm in Figure 2 as follows. Whenever we measure a sub-band that lies within a forbidden channel, we skip the measurement and simply set the measured power to exceed the threshold.

Finding available channels only: Some application might want to find available channels rather than building a complete spectrum map. QuickSense algorithm can be modified such that it terminates as soon as it finds the available channels required by the application. However, this may not be optimal all the time. If we know the noise power of each channel accurately, e.g., from noise map, we can run a binary search algorithm. This will result in at most $log_2(N)$ measurements.

IV. Evaluation

We first perform proof of concept experiment using real hardware devices. Then we perform trace-driven simulation using the real traces collected at UHF TV band.

A. Proof of Concept Experiment

The main goal of this experiment is to demonstrate the availability of QuickSense using hardware implementations in real environments. We do not implement the control unit in this initial prototype but manually run our algorithm instead. The hardware costs below \$100 and consumes less than 200mW.

With the implementation, we perform benchmarks to see how faster QuickSense can perform channel sensing compared to the sequential scan in UHF TV band at 470 – 698MHz. The result is that the enhanced QuickSense design performs 2.9X faster searches compared to the sequential scan.

We also experiment to see the amount of time required per channel sensing, which is directly related to channel sensing delay and accuracy. QuickSense reliably detects signals around -80dBm with only 50μs sensing delay.

1) Hardware Components: We use hardware components that include multiple analog devices and a signal generator as shown in Figure 8. The final output is DC voltage that represents the detected energy at filters. The voltage values are finally recorded in the PC. Note that we do not use preselector in this design as we do not need to demodulate the input signal but only to measure the energy level. Hardware components are listed in the order of RF processing in the following.

- Antenna: A7U UHF antenna from Lectrosonics for UHF TV band is used.
- Low noise Amplifier: LNA increases input signal strength while maintaining low noise figures. Analog Device's ADL5521 is used. It has 20dB gain.
- Mixer: The mixer is used to down-convert signals at carrier frequencies into baseband signals at DC. The mixer takes two inputs, one from LNA and another from the signal generator. We use Analog Device's AD8342.
- Filter: Since the ideal bandwidths as required in QuickSense are not available in the market, we try to make the best approximation based on the availability. We use MiniCircuit's lowpass filters SLP-5, SLP-15, SLP-30 and SLP-50. Bandwidths of these filters are 5MHz, 15MHz, 32MHz and 48MHz, respectively.
- Energy Detector: The energy detector measures the signal energy from filter outputs in dBm. We use Analog Device's AD8306. The output is 0.4V when the input signal is around -87dBm and it linearly increases until the signal strength becomes 13dBm. At this point, the output becomes 2.2V.
- ADC: The ADC reads the energy detector's voltage output into digital samples. We use Analog Device's AD9649 which has the 14bit bandwidth and the 20MHz sampling rate. At PC, VisualAnalog software stores the digital samples into files.

2) Energy Detection Threshold: The energy detection threshold is derived from the noise power measurements. The noise power is measured using a spectrum analyzer and we build the noise map using the information. The noise spectral density differs at different frequencies. In UHF band, for example, its average is -155.6dBm/Hz in our lab. It translates into -88.6dBm, -83.8dBm, -82.2dBm and -78.8dBm with 5 MHz, 15MHz, 32MHz and 48MHz filter bandwidth, respectively.

3) Benchmark: We filter the received signal using different filters. The energy outputs from the filters are measured at every fixed frequency intervals. Based on the data, we see how many channel sensings our basic/enhanced QuickSense design

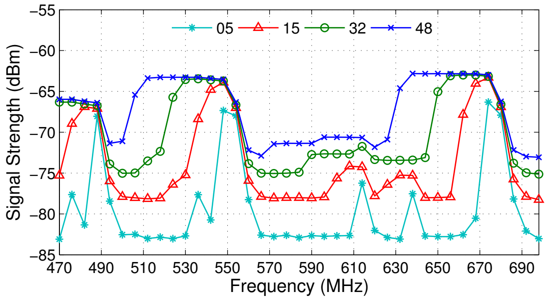

Fig. 9. Energy measurements of UHF TV whitespace at different filter bandwidths.

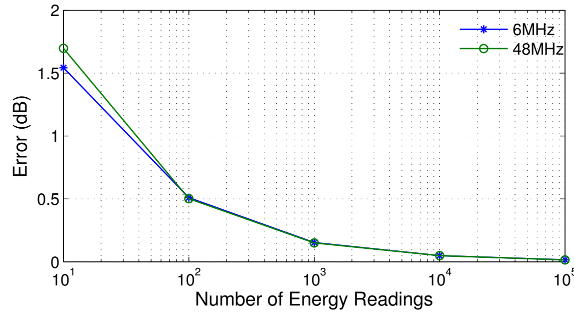

Fig. 10. As the number of energy readings is around 1,000, the noise estimation error becomes very low, around 0.5dB.

Channel	Frequency (MHz)	Distance (miles)
15	476 - 482	31
17	488 - 494	10
25	536 - 542	32
27	548 - 554	9
28	554 - 560	10
38	614 - 620	26
42	638 - 644	31
48	674 - 680	8
49	680 - 686	10

TABLE I
LIST OF 9 TV STATIONS WITHIN THE AREA AND THEIR DISTANCES TO OUR BUILDING.

Filter	QuickSense-B	QuickSense-E
5MHz	28	13
15MHz	15	13
32MHz	6	13
48MHz	1	13
Total	50	13

TABLE II
NUMBER OF CHANNEL SENSING PER FILTER WITH QUICKSENSE-B AND QUICKSENSE-E IN UHF TV BAND.

would require to build the spectrum map. For convenience, we refer each as QuickSense-B and QuickSense-E.

There are total 38 channels in UHF TV whitespaces where each channel occupies 6MHz. This means that the sequential scan would require 38 channel measurements. Among these, 9 channels are being used by TV stations operating within our area. As a ground truth, we list those TV stations and their respective distances to the our lab in Table I.

Figure 9 shows the energy outputs from the filters having 5MHz, 15MHz, 32MHz and 48MHz bandwidths. They are measured at every 6MHz frequency intervals. We use this data to emulate QuickSense-B and QuickSense-E and compute the number of channel sensing operations required by each. Table II shows the result. QuickSense-B completes the sensing with 50 measurements in total. In this case, the performance of QuickSense-B is even worse than the sequential scan. As we will revisit this issue in Section IV-B3, QuickSense-B begins to perform poorly if one fourth of the channels are occupied. On the other hand QuickSense-E builds the spectrum map with only 13 measurements. This is 2.9X gain over the sequential scan. The main reason is that all the filters go through the channel sensing in parallel.

4) Accuracy vs. Delay: In reality, an accurate channel detection needs multiple energy readings. The main reason is the background noise. A single measurement can be distorted by the background noise. So it is required that the receiver gathers the sufficient number of energy readings to cancel out the contribution of background noise.

Using the hardware implementation, we observe the noise estimation error $|E[\omega^2] - \Omega|$ in dB with regard to the number of energy samples. Here $E[\omega^2]$ is the average energy reading from a vacant channel and Ω is the actual noise power accurately measured using the spectrum analyzer. We vary the number of energy measurements to see the change in the estimation error. Two different filter bandwidths (6MHz and 48MHz) are tested. Figure 10 shows the result. From the figures, we observe that the estimation becomes accurate as the number of energy samples increases. If we collect 1,000 energy samples and average them, the estimation error becomes 0.1dB.

Analog hardware does not have an impact on the total sensing delay. The response delay of the analog filters and the analog energy detector is very short. Analog filters have response delays of around hundreds ns. SLP-5 has the longest response delay among the filters that we have, and it is $327ns$ [21]. An energy detector is even quicker, having the maximum delay of around $0.6ns$ [22]. Given that 1,000 measurements are required per channel sensing, the total delay incurred by analog hardware part will not exceed $500\mu s$.

Suppose an ADC is used to collect energy readings from the energy detector. If a device is to collect 1,000 energy readings per channel sensing, it takes $0.05\mu s$ with 20MHz ADC and $1ms$ with 1MHz ADC, respectively.

B. Trace-Driven Simulation

In this section, we perform sets of simulation using real signal captures. We first investigate how accurately QuickSense can sense a channel. Then we compare the algorithmic performance of QuickSense with the sequential scan and the compressed sensing.

1) Simulation Setup: For realistic evaluation, we use live over-the-air signal captures. The signals in all TV white spaces, ranging from 470MHz – 698 MHz, are captured with SDR with an UHF indoor antenna attached. As the sample rate of the radio is 25 MHz, it cannot cover the whole white space. We sample each TV channel separately after down-

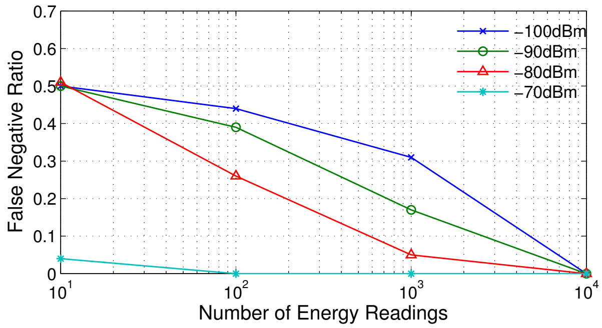

Fig. 11. The ratio of false negative signal detection is shown with different number of energy readings used for the channel sensing. Also the signal strength is differentiated.

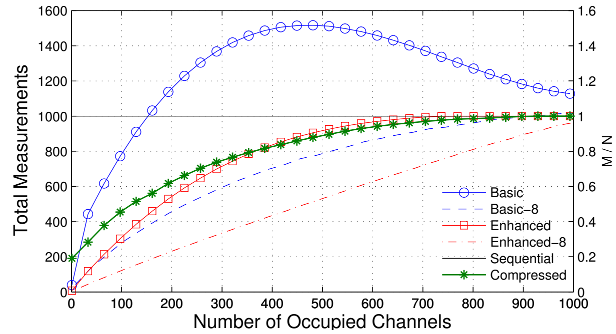

Fig. 12. Performance of sequential scan, QuickSense-B, QuickSense-E and compressed sensing. Basic-8 and Enhanced-8 show the cases where every 8 channel is grouped together.

converting to DC. The captured TV signal strength range between -66 dBm to -84 dBm in our measurement. Once all the TV channels are captured, we synthesize the separate signal samples in the following way. The signal samples from each channel are interpolated and then oversampled by a factor $2 \times B/R$, where B is the total bandwidth of the white space and R is the sample rate. Now we aggregate these samples to have the synthesized broadband signal $S(t)$ as $S(t) = \sum_k s_k(t) \cdot e^{2\pi j f_k t}$, where $s_k(t)$ is the baseband signal at the kth TV channel at time t and f_k is the carrier frequency of the kth TV channel. The term $e^{2\pi j f_k t}$ represents carrier signal. In the process, it is possible to arbitrarily vary the number of occupied channels within the synthesized signal. We can simply choose signal samples from either sets of occupied or vacant channels when creating the synthesized signal.

The behavior of analog filters is emulated using a software implementation. We generate $\log(N)$ fixed-bandwidth Chebyshev lowpass filters. Chebyshev filter is chosen because it has analog implementation counterparts and provides good separation between sub-bands. The unit bandwidth of the filters is 6MHz following the actual TV channel bandwidth.

For the energy detection threshold, we use the measured data as in Section IV-A2. The noise spectral density -155.6 dBm/Hz and it translates into -87.8 dBm in 6 MHz TV channel. It increases by 3dB as the filter bandwidth is doubled.

We implement [14] as the compressed sensing algorithm. It basically assumes the use of an analog DFT device, a hardware that obtains entire wideband spectrum in the analog domain. Then a low-rate ADC performs random-sampling over the analog buffer and the results are linearly combined afterwards. For the comparison with QuickSense, we calculate M/N, where M is the number of samples and N is the number of Nyquist-rate samples, that yields 99% detection accuracy. The sparsity information is required and we assume it is completely known at each run.

2) Channel Sensing Accuracy: The false negative ratio is measured while we vary the number of energy readings per channel sensing (as in Section IV-A2) and signal strength. The false positive is almost negligible in the result and we omit the result here. Figure 11 shows the result. With -80dBm or higher signal strength, 1,000 samples yield reasonably accurate estimation of the signal power. Note that the signals with -70dBm or higher strength can be sufficiently detected with only 10 energy readings. In DSA networks where signal strength is usually very strong, number of energy readings can be significantly saved.

3) Channel Sensing Speed: We count the total number of channel measurements for building a complete spectrum map. Spectrum bandwidths with 1,000 channels are tested. The number of occupied channels within the spectrum is varied from 1 to 1,000. The simulation is run 100 times. At each run, occupied channels are located at random positions.

The result is plotted in Figure 12. The sequential scan always scans through all the channels and thus spend constant number of measurements. On the other hand, the compressed sensing always completes the channel sensing with less sampling than Nyquist-rate samples.

QuickSense-B outperforms the sequential scan when the spectrum is sparse. For example, when 50 channels are occupied, it takes less than 500 measurements, which is 1/2 of the sequential scan. But the performance soon deteriorates due to the following reason. When vacant channels are detected QuickSense-B moves to next searching space and expands its searching window. Otherwise if a channel usage is detected it shrinks the window while staying at the current searching space. As the spectrum usage becomes denser, it will repeatedly expand and shrink its window while moving back and forth around the same frequency. We observe in Figure 12 that QuickSense-B performs the channel sensing most when the spectrum is around half-occupied. As the spectrum gets denser, the sensing time decreases because QuickSense-B performs sequential scan on occupied channels most of the time.

QuickSense-E shows significantly better performance. Compared to QuickSense-B, QuickSense-E requires as few as 20% of the total measurements. Especially when only one channel is occupied, it performs the measurement operation less than 5 times in average. This superior performance comes from the parallelized channel detection using $log_2(1,000) = 10$ filters. QuickSense-E does not experience the same problem as QuickSense-B; QuickSense-E never goes back to previous searching space when it senses an occupied channel. While it is hard to generalize, in our simulation the performance of QuickSense-E is even better than that of the compressive sensing within sparse channels. Yet QuickSense-E does not require the prior knowledge of channel sparsity or specially designed hardwares.

What would happen if the channels are grouped and used as a single channel, e.g., for channel bonding? We group every 8 channels and make them consecutively located. In this case, there are less number of fragmented channels and QuickSense becomes very efficient. QuickSense-B saves up to 51% measurements. We can see that the channel sensing of QuickSense-E is even less than the actual number of occupied channels in the consecutive channel setup.

V. Related Work

There is a large body of work on narrow band sensing [11] which are either based on detecting the signal energy or the signal signature. The signatures include waveform or cyclostationality of the signals. When it is known, the signature-based techniques can detect signals with low SNR. However, these techniques are complicated. Also prior knowledge of received signals is required, which is not always available.

The most related works are compressed-sensing-based techniques [14], [15], [16], [17]. Typically, a lower rate ADC is used to carry out sub-Nyquist sampling. These techniques propose to separate the "sample-and-hold" stage of the ADC from the "quantization". It requires compressed sensing projections without quantizing. This has to be done in the analog domain using special chips. For instance, analog buffers are used to store the signal input before the quantization stage. Also the special hardware is required that is capable of doing matrix multiplication in the analog domain. Real-world implementation of compressed-sensing-based spectrum sensing is thus very challenging. The accuracy of the spectrum sensing depends on the compression ratio (higher means retaining more samples). Prior works [14], [15] usually assume that the sparsity of the signal is known. In contrast, QuickSense naturally adapts to the sparsity of underlying spectrum and the accuracy does not depend on the compression ratio.

Recently Meng et al. [23] proposed a collaborative compressed spectrum sensing approach. Each node is equipped with a frequency selective filter, which linearly combines multiple channel information. The linear combinations are sent as reports to the fusion center, where the occupied channels are decoded from the reports by compressed sensing algorithms. By exploiting the sparsity of occupied channels, the number of reports needed by the fusion center is reduced. Gathering reports introduces significant delay.

VI. Conclusion

In this paper, we have proposed a fundamentally new approach to fast wideband spectrum sensing. Unlike compressed sensing or wideband radio approaches, QuickSense does not require complicated hardware devices. Instead, we use off-the-shelf analog filters and energy detectors that are cheap and energy-efficient. We combine the hardwares with novel and efficient algorithms. The complexity of QuickSense is asymptotically same as that of compressed sensing. We have performed proof of concept experiments using real hardwares. Simulation results demonstrate that the performance of QuickSense is comparable to compressed sensing while QuickSense does not require prior knowledge of sparsity of the spectrum.

VII. Acknowledgements

Part of Li Erran Li's research and Sungro Yoon's summer internship at Bell Labs in 2011 for this work are supported by CNS-1017871. This work is partially supported by the General Research Funds (Project No. 414812), established under the University Grant Committee of the Hong Kong Special Administrative Region, the China NSFC grant (Project No. 61271277) and 973 Program (Project No. 2012CB315904).

References

[1] Cisco, "Cisco visual networking index forecast projects 18-fold growth in global mobile Internet data traffic from 2011 to 2016."

[2] CNN, "Sorry, America: Your wireless airwaves are full." http://money.cnn.com/2012/02/21/technology/spectrum_crunch/index.htm.

[3] Whitehouse, "President Obama details plan to win the future through expanded wireless access," 2011.

[4] L. Jiao, V. Pla, and F. Li, "Analysis on channel bonding/aggregation for multi-channel cognitive radio networks," in *Proceedings of EW 2010*.

[5] FCC, "Second Memorandum Opinion and Order," http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-10-174A1.pdf, 2010.

[6] Spectrum Bridge, "Show my white space," http://whitespaces.spectrumbridge.com/whitespaces/home.aspx.

[7] President's Council of Advisors on Science and Technology, "Realizing the full potential of government-held spectrum to spur economic growth," http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast_spectrum_report_final_july_20_2012.pdf.

[8] "IEEE standard 802.22," Jan 2009.

[9] Shared Spectrum, "Full speed ahead traffic to increase 18x by 2016."

[10] ——, "General survey of radio frequency bands – 30MHz to 3GHz."

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

[12] B. Le, T. W. Rondeau, J. H. Reed, and C. W. Bostian, "Analog-to-digital converters: A review of the past, present, and future," *Signal Processing Magazine, IEEE*, 2005.

[13] Texas Instruments Inc., "12-bit, 1000 MSPS ADC with analog input buffer."

[14] J. N. Laska, W. F. Bradley, T. W. Rondeau, K. E. Nolan, and B. Vigoda, "Compressive sensing for dynamic spectrum access networks: Techniques and tradeoffs," in *Proceedings of DySpan*, 2011.

[15] M. Rashidi, K. Haghighi, A. Panahi, and M. Viberg, "A NLLS based sub-Nyquist rate spectrum sensing for wideband cognitive radio," in *Proceedings of DySpan*, 2011.

[16] J. Tropp, M. Wakin, M. Duarte, D. Baron, and R. Baraniuk, "Random filters for compressive sampling and reconstruction," in *Proceedings of ICASSP*, 2006.

[17] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, "Toeplitz-structured compressed sensing matrices," in *Proceedings of SSP*, 2007.

[18] H. Kim and K. G. Shin, "In-band spectrum sensing in cognitive radio networks: Energy detection or feature detection?" in *Proceedings of ACM MobiCom*, 2008.

[19] V. Saari, M. Kaltiokallio, S. Lindfors, J. Ryynanen, and K. Halonen, "A 240-MHz low-pass filter with variable gain in 65-nm CMOS for a UWB radio receiver," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 56, no. 7, 2009.

[20] K. Tan, Y. He, H. Shen, J. Zhang, and Y. Zhang, "Spectrum virtualization layer," Microsoft Research, Tech. Rep. MSR-TR-2011-108, 2011.

[21] Mini Circuits, "SLP-05," http://www.minicircuits.com/pdfs/SLP-5.pdf.

[22] Analog Devices, "AD8306," http://www.analog.com/en/AD8306.

[23] J. Meng, W. Yin, H. Li, E. Hossain, and Z. Han, "Collaborative spectrum sensing from sparse observations in cognitive radio networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 2, 2011.