Detecting Spectrum Misuse in Wireless Networks

Kefeng Tan*, Kai Zeng†, Daniel Wu‡, Prasant Mohapatra*
*University of California, Davis. †University of Michigan, Dearborn. ‡Google
*{kftan, pmohapatra}@ucdavis.edu, † kzeng@umich.edu, ‡ danwu@google.com

***Abstract*—In contrast to conventional static fixed-width channel allocation, on-demand dynamic variable-width channel allocation has shown that it can effectively improve the fairness, throughput, and spectrum efficiency of wireless networks. Airtime utilization (the percentage of time spent on transmissions) is often used to characterize the spectrum demand of networks. The higher the airtime utilization of a network is, the more spectrum the network should be allocated to. Normally, if all wireless devices in a network utilize spectrum effectively, the airtime utilization can faithfully reflect the spectrum usage. In practice, however, spectrum can be ineffectively used due to the misconfiguration of wireless devices, such as inappropriate bit rate configuration, conservative transmit power setting, or mismatch between channel-width and bit rate. The misconfiguration not only degrades the performance of its local network, but also causes the inflation of local network's airtime utilization and thus results in an unfair spectrum allocation. To address the problem, we present Pinokio, a system that monitors spectrum usage at access points, detects spectrum misuse and improves spectrum efficiency. Our extensive evaluations suggest that Pinokio can accurately detect spectrum misuse, and limit the inflation of airtime utilization from more than 730% to less than 20%.**

I. Introduction

On-demand dynamic variable-width channel operation has emerged to replace conventional static fixed-width channel operation due to its improvements on throughput, fairness, power efficiency, and spectrum efficiency to wireless networks [7], [9], [14], [20]. In this new paradigm, the spectrum demand of each network is periodically assessed and the spectrum is reallocated among all subnetworks based on their needs. One of such an exemplar system is shown in Fig. 1, where a *spectrum allocation server* periodically collects spectrum usage information from access points (AP), based on which the server reallocates a piece of non-overlapping spectrum to each of APs to avoid interference.

Standards body has already adopted the concept of variable channel-width. IEEE 802.11 version 2007, for instance, proposes 5 MHz, 10 MHz and 20 MHz operations; IEEE 802.11n enables bonding two 20 MHz channels into a 40 MHz channel. 3GPP LTE also requires 6 different channel-widths, and WiMAX is able to accommodate 11 operational bandwidths. In addition, some off-the-shelf devices have also supported variable-width channel operations, such as the Atheros 11a/b/g Wi-Fi chipset, IEEE 802.11n chipset, and Intel WiMAX chipset [2], [5], [7].

To ensure wireless spectrum to be allocated efficiently, spectrum usage at each subnetwork has to be assessed accurately. Spectrum usage can be evaluated by aggregate throughput, the number of associated clients, or airtime utilization (percentage of time spent on transmissions). Compared with

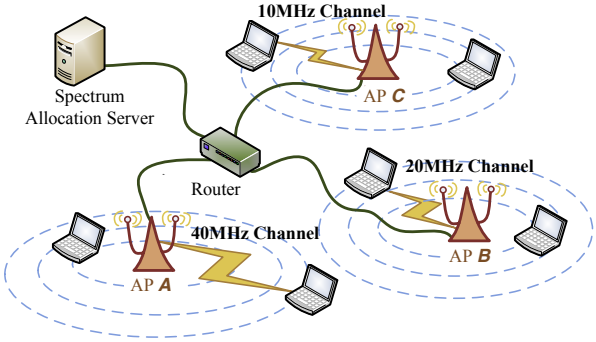


Fig. 1: Spectrum reallocatable WLANs: spectrum is periodically reallocated among subnetworks based on the demand.

the other two metrics, airtime utilization is more accurate in characterizing the spectrum usage because it directly measures the time spent on all transmissions over the air. Airtime utilization has also been traditionally used to ensure fair resource allocations [8], [18], improve the performance of commercial enterprise wireless LANs [1], and recently used to facilitate dynamic spectrum access [6], [21].

Airtime utilization can truthfully reflect spectrum usage only if all wireless devices utilize the spectrum to its full capacity. The premise, however, can be easily violated in practice. For example, if the auto bit rate adaptation on a wireless interface card is accidentally disabled and the bit rate is set to be fixed, it will result in ineffective spectrum usage. If the rate is set too low, the spectrum is underutilized. If, on the other hand, it is set too high, the channel will be busy with unnecessary retransmissions, and thus not be fully utilized either. In either case, the local airtime utilization will be inflated. Consequently, extra spectrum that is disproportional to the real time utilization will be allocated to it, which otherwise would be allocated to the neighboring networks.

To illustrate the impact of such misconfigurations on spectrum allocation, consider allocating a piece of spectrum of 85 MHz (ISM band at 2.4 GHz) to three neighboring basic service sets (BSS) - *A, B* and *C*. To avoid interference, each BSS should get a piece of non-overlapping spectrum. For simplicity, we do not consider guard bands and we assume no other interfering sources. Similar to the approach in [14], the spectrum is allocated proportional to the airtime utilization of networks. Initially, BSS *A* and *B* have heavy traffic loads, and the airtime usage of both is 90%; BSS *C* is idle, and its channel is busy only 10% of time. Proportionally, BSS *A, B* and *C* get 40 MHz, 40 MHz and 5 MHz of spectrum, respectively. If a client at BSS *C* is misconfigured and its local

airtime utilization is increased up to 75% (we will show the feasibility in Section II), the spectrum will be reallocated and their shares are changed to 30 MHz, 30 MHz and 25 MHz, respectively. Because BSS *C* usurps 10 MHz spectrum from each of its neighbors BSS *A* and *B*, the network performance of the latter two is doomed to be degraded.

To mitigate the aforementioned problem, we propose and develop Pinokio[1] , a system to detect airtime inflation based on certain characteristics of bit rate behaviors that fundamentally differentiate spectrum misuse from normal usage. Briefly, our contributions are:

- Identify a new source causing the degradation of spectrum efficiency in wireless networks. While many existing works focus on optimizing spectrum allocation algorithms, they alone cannot guarantee a fair spectrum allocation. The effectiveness of spectrum allocation also depends on the accuracy of the metric used for allocations. To the best of our knowledge, no prior work has ever pointed out this problem.
- Develop a new monitoring system, Pinokio, that can automatically detect spectrum misuse. The detection algorithm of Pinokio is based on the statistical algorithm of intrusion detection expert system (IDES) developed by SRI [11]. We identify a small set of key characteristics of bit rate behavior to help anomaly detection, and propose corresponding strategies to deal with anomalies.
- Evaluate Pinokio system with trace-driven simulations as well as our software-defined radio (SDR) based platform experiments. Our evaluations are based on data trace consisted of 54+ million transmission records and 720+ off-the-shelf devices from a real operational network. Our results show that the proposed system can detect spectrum misuse accurately, and improve wireless spectrum efficiency.

The rest of this paper is organized as follows. We show a series of misconfigurations that cause the airtime utilization inflation in Section II, and explain the components of Pinokio system in Section III. In Section IV, the performance of Pinokio system is evaluated. The related works are discussed in Section V. Conclusions are drawn in Section VI.

II. Spectrum Misuse

In this section, we first introduce the system model used in this study, then describe four possible spectrum misuses, and finally illustrate their impacts on airtime utilization with experimental results.

A. System Model

Airtime utilization is defined as the time spent on transmitting data frames (T_{data}) over the total time (T_{total}).

$$\text{Airtime usage} = \frac{T_{data}}{T_{total}} \quad (1)$$

We consider an enterprise wireless LAN depicted in Fig. 1, in which the spectrum is reallocatable among all BSSs based on the needs. Once the channel-width is set at a BSS, all operations are bounded by it. All other PHY and MAC layer properties are assumed to be compliant with generic IEEE 802.11 standard, such as CSMA/CA and bit rate adaptations. We also make the following assumptions about the system:

1) Only the bit rate behavior of mobile clients is suspicious, but that of APs is trustworthy. The assumption is realistic because APs in enterprise networks are typically configured and managed by IT professionals, and thus less likely to be misconfigured.
2) We only consider misconfigurations that can cause spectrum misuse at wireless interface cards. Modules like bit rate adaptation, transmit power settings can possibly be malfunctioning.
3) We assume that the spectrum allocation server is always allocating spectrum fairly based on airtime utilization. It is the lack of examination on the accuracy of airtime utilizations that leads to unfair spectrum allocations.

Our approach is not targeting at any specific spectrum allocation algorithm. Rather, it solves the problem of airtime inflation regardless of the spectrum allocation algorithm used.

B. Possible Spectrum Misuses

We identify four possible misconfigurations at a client side that can cause spectrum misuse, and explain them in detail as follows.

(a) *Conservative bit rate*: If the bit rate at a client is set too conservatively, it will not only slow down its own transmissions, but also intensify contentions among all other clients. As a result, the airtime usage increases.

(b) *Aggressive bit rate*: If the bit rate at a client is set too aggressively, it will incur many unnecessary retransmissions. Much of the airtime will be wasted on retransmissions. The possible airtime increase is bounded by the maximum retransmission limit, which is set to 7 in MadWifi driver [3].

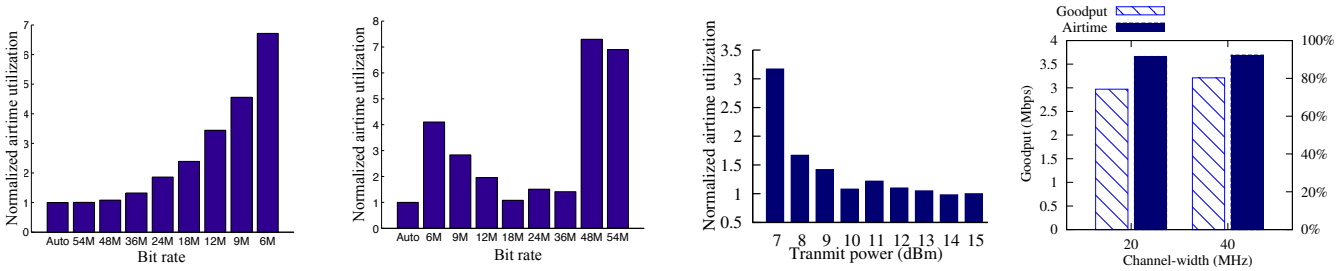
(c) *Low transmit power*: Instead of setting the bit rate, lowering the transmit power also decreases the transmission rate. Its effect is similar to dropping the bit rate, except that the bit rate can still be adjusted within a possibly reduced range, rather than be fixed.

(d) *Bit rate and channel-width mismatch*: Assume a wide channel is allocated to an AP and its clients, over which all of them are operating properly. If one of the clients moves further away from the AP, its frames will have to be delivered at a lower bit rate to ensure reliable transmissions. However, this is not an efficient way to utilize a wide channel. The same capacity can also be achieved over a narrower channel with increased bit rates (we will demonstrate this in Section III).

C. Impact of Spectrum Misuse on Airtime Utilization

To quantify the impact of spectrum misuse on airtime utilization, we conduct experiments to emulate each of the misuse scenarios and compare the inflations of airtime usage. The experiments are conducted with two laptops equipped with IEEE 802.11b/g/n wireless card and MadWiFi driver v0.9.4. We let one laptop send the other UDP traffic. Each

[1]It is analogous to the fictional wooden puppet, Pinocchio, whose nose becomes longer when he is telling an exaggerated fact.



(a) Conservative bit rate (b) Aggressive bit rate (c) Low transmit power (d) Bit rate and channel-width mismatch

Fig. 2: Impact of spectrum misuse on airtime utilization

round of transmission lasts about 2 minutes and the average throughput of 5 rounds is reported. In each scenario, all the airtime increases are normalized with regard to that of the normal case.

Conservative bit rate: We intentionally lower the bit rate to examine its impact on airtime usage. An UDP traffic of 4 Mbps is loaded over a line-of-sight link between two laptops about 20 feet apart. The bit rate is fixed at the sender, but is set to default `auto` at the receiver. We plot the normalized airtime utilizations of all IEEE 802.11g bit rates, with regard to that of the auto bit rate. Fig. 2(a) shows how the airtime usage increases with the decrease of the bit rate. Through analysis to its trace, we find that frames are sent at 54 Mbps most of the time (95%) when the auto bit rate is used. In addition, its packet retransmission ratio is less than 0.02%, which indicates that the wireless link is highly reliable. This explains the similarity between the case of 54 Mbps and the case of auto bit rate. As the bit rate is gradually decreased, the airtime utilization is increased by 7x at 6 Mbps. However, the increase is not inversely linear proportional to the bit rate because the bit rate only affects the payload. Other overhead like backoff time, SIFS, DIFS, ACK and preambles also have to be taken into consideration.

Aggressive bit rate: We deliberately raise the bit rate to measure its impacts on airtime usage. An UDP flow of 1 Mbps is loaded over a link of about 50 feet with walls in between, which makes the link much less reliable than that in previous case. Fig. 2(b) shows how the normalized airtime usage with regard to that of the auto bit rate varies over all bit rates. In this setting, the auto bit rate stays at 18 Mbps most of the time (99%), which results in similar airtime usage with that of 18 Mbps. As the bit rate is increased from 18 Mbps to 36 Mbps, the retransmission rate jumps from 1.4% to 39.6%, but the airtime utilization just increases slightly because the overhead of retransmissions can be compensated by the increased bit rates. At 48 Mbps and 54 Mbps, the retransmission rate increases up to 90.7% and 90.1% respectively, which results in 7.3x and 6.9x airtime inflation respectively.

Low transmit power: Similar effects can be achieved by reducing the transmit power, instead of adjusting the bit rate. All other settings are maintained the same as in the aggressive bit rate case, except this time we do not fix the bit rate but decrease the transmit power from 15 dBm to 7 dBm, 1 dBm at a time. As a result, the airtime usage gradually increases with the decrease of transmit power, as shown in Fig. 2(c). The increase, however, is not linear. It rises sharply when the transmit power is lowered below a certain level, which is 7 dBm in this case.

Bit rate and channel-width mismatch: Setting an inappropriate channel-width can also decrease spectrum efficiency. An UDP traffic of 4 Mbps is loaded over a 50 foot link with walls in between. We first set the width of operational channel to 40 MHz through channel bonding. About 3 Mbps goodput is observed, which consumes about 92% of the total airtime. Its data trace shows that most of the transmissions use 6 Mbps. Without changing other settings, we then switch the channel-width to 20 MHz. The auto bit rate primarily chooses 12 Mbps this time, and similar goodput is achieved at the cost of the same amount of airtime. Obviously, the spectrum utilization of the 20 MHz channel is higher than that of the 40 MHz channel, in terms of Mbps per MHz. The difference can be attributed to the mismatch between the bit rate and the channel-width.

III. Design of Pinokio System

In this section, we propose a design of spectrum misuse detection system - Pinokio, to address the aforementioned problems. We first introduce the IDES statistical algorithm, and then explain how to apply the algorithm in detecting spectrum anomaly. Specifically, we identify a small set of key features on bit rate behavior, and present each of their statistical profiles to help understand why and how they can help spectrum anomaly detections.

A. IDES Statistic Algorithm

The IDES algorithm maintains a statistical profile on network system's normal behavior based on training data [11]. When the network's recent behavior significantly deviates from its normal statistical profile, an alarm will be triggered about possible anomaly.

Specifically, the IDES algorithm works as follows. The detection system first needs to identify a series of behaviors of network, say events E_1, E_2, ... E_k, whose expected occurrence probability is p_1, p_2, ... p_k, respectively. The probability profile can be acquired from a long-term training. The system then

monitors the same aspects of network behavior and gets a different set of probabilities p'_1, p'_2, ... p'_k. To verify if the two sets follow the same distribution, we test the following hypothesis:

$$H_0(\text{Normal}) : p'_i = p_i,\ i = 1, 2, \ldots, k$$
$$H_1(\text{Abnormal}) : H_0 \text{ is not true}$$

To this end, construct a statistic Q. Let

$$Q = \sum_{i=1}^{k} \frac{(X_i - N \times p_i)^2}{N \times p_i}$$

Where X_i represents the number of occurrence for event E_i, and N is the total number of all events. It has been proven that Q has an approximate χ^2 distribution with $(k-1)$ degree of freedom for a large N (>50), provided the events E_i ($i = 1, ..., k$) are independent. Therefore, Q measures the *deviation* of the observed values over the expected values. Formally,

Reject H_0 :
if $Pr(Q > q) < \alpha$ OR Equivalently if $q > \chi^2_\alpha(k-1)$;
Accept H_0 :
if otherwise;

Where q is an instance of Q, and α is the desired significance level of test. The two conditions in the second line are equivalent, meaning that the profile under test is statistically different from the normal profile. Therefore the hypothesis H_0 should be rejected, which indicates that an anomalous behavior has occurred.

B. Bit Rate Behavior for Anomaly Detection

In this Section, we explain how to apply the algorithm for spectrum anomaly detections. As we have shown in Section II, potential spectrum misuses can influence normal bit rate behavior. We therefore identify four key features of bit rate behavior for spectrum anomaly detections. Adding more features may increase the sensitivity of the detection system, but can also introduce performance overhead.

In addition, we do not choose any features unique to generic bit rate adaptations, but select those that can fundamentally characterize normal bit rate behavior. This is because many bit rate adaptations on off-the-shelf devices are proprietary and are implemented either in device drivers or in firmware. Considering the sheer number of different devices, reverse engineering all adaptation algorithms, even if it is possible, is a time-consuming process. Even worse, such an approach may limit its applications. As we show in Section IV, the features we select are very effective in detecting spectrum misuses. In the following, we introduce each of the features respectively.

Contiguous adjustment of bit rate. We are interested in knowing how much the bit rate changes at a time. In a normal case, the bit rate will only decrease one step at a time in case the channel deteriorates, or increase one step at a time when the channel quality improves. The consecutive adjustment of bit rate can avoid two things. On one hand, a dramatic bit rate jump can cause the loss of connectivity. On the other hand, a large bit rate reduction results in the delay of transmission. In an abnormal case, the bit rate may be set too high or too low, thus not change in a contiguous fashion.

Response to transmission failure. Transmissions in WLANs may be failed in two cases. A frame can be lost over the air due to the weak reception signal. Or it can collide with other frames at a receiver, and thus be decoded incorrectly. In either case, retransmissions should be slowed down upon a few transmission failures. However, if the bit rate is fixed, the bit rate in retransmissions are remained the same regardless of the failures. In IEEE 802.11, a `Rx` bit in MAC layer packet header indicates whether the packet is a *retry* frame. Alternatively, a retransmission frame can be distinguished from the rest by its sequence number at the MAC layer, whose sequence number is shared with some of the previous frames transmitted.

Reciprocity of bit rates. It is widely believed that wireless channels exhibit reciprocity [16]. Recent experimental results have also shown that the received signal strengths (RSS) between a communicating pair are highly correlated, whose similarity can even be exploited for generating secret keys [12], [23]. The bit rates between a communicating pair should therefore also exhibit certain reciprocities. However, this reciprocity will be altered when anomalous behavior is present.

Percentage of low bit rates. If an assigned channel is fully utilized, the percentage of low bit rate should not be too high. However, if a majority of transmissions use low bit rates, it means that the channel is underutilized. We therefore use the percentage of low bit rate to distinguish the two cases.

C. Building Statistical Profile for Normal Behavior

In IDES algorithm, building the statistical profile for normal behavior is through a long-term learning from normal behavior. In the context of spectrum misuse detection, the training can be divided into two steps. One is to learn an empirical distribution for Q statistics, and the other is to acquire a proper threshold for each feature. Because factors, such as signal propagation and AP transmit power, can strongly influence the bit rate behavior. The process of training can make the system better adapt to different networks and wireless environments. The initial training to obtain proper distributions and thresholds is thus essential to the detections.

The detection system also needs to adapt to another change - the evolution of network behavior. New bit rate schemes are constantly evolving and may not be matched with the historical profile. Fortunately, the IDES algorithm provides a scheme to allow the gradual adaptation to behavior changes. A fading factor is defined such that the normal statistical profile will eventually "forget" the ancient data. If, on the other hand, the system behavior changes abruptly, the detection system can discard all historical data and rebuild the profile through the training.

To see why the proposed features can help detect spectrum misuse, we present some statistics for each of them in the next. Our results are based on a thorough analysis to the data trace of SIGCOMM'08, which consists of 47+ million transmission records and 600+ unique MAC addresses [17]. For the cases

where it is insufficient to validate, we also supplement our own experimental results to illustrate.

Bit rate change profile: We compare all bit rates changes in consecutive transmissions. In any two consecutive records from the same source, if the bit rate changes more than one step, it is termed as a non-contiguous change. Table I lists the percentage of non-contiguous bit rate changes. Since SIGCOMM'08 trace (the bottom row of the table) is not acquired in a highly congested situation, we also conduct our own experiments (top three rows of the table) on a testbed containing 4∼6 laptops, in which all laptops are contending with each other to transmit. Our results indicate that collisions may slightly increase the percentage of non-contiguous bit rate changes, but is unlikely to change the nature of bit rate adaptations. To summarize, *bit rate is adjusted one level at a time most of the time.*

Number of nodes	Non-contiguous bit rate change	Total bit rate changes	Percentage
4	35,350	1,326,858	2.60%
5	27,686	1,204,044	2.25%
6	49,476	900,074	5.21%
613	1,283,642	47,798,680	2.69%

TABLE I: Analysis to bit rate changes

Retransmission profile: We analyze bit rate changes in all retransmission frames. Fig. 3 shows that after how many retransmissions the bit rate starts to decrease. Among all retransmissions, 43% of them drop the bit rate after the first failure, and almost all slow down before the fourth attempt. Fig. 4 shows that how much the bit rate decreases at a time during the retransmissions. 90% of retransmissions drop one level, and merely 5% drop two or more. The result is consistent with the bit rate change profile. The profile can be summarized as that *bit rate is most likely to decrease within three consecutive frame losses.*

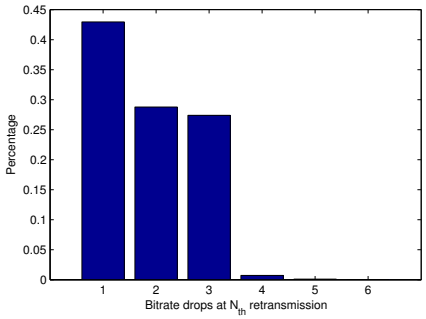


Fig. 3: Distribution of when bit rate starts to decrease in retransmissions

Reciprocity profile: We correlate the bit rate between communicating pairs. From the trace, we pick 234 pairs from more than 3000 communicating pairs who send at least 1000 frames to each other. As the reciprocity generally holds within a certain channel coherence time, we only analyze pairwised packets interleaved less than 100 ms.

Intuitively, the correlation of bit rates between a communicating pair could be used to characterize the reciprocity. However, from the cumulative distribution function (CDF) of

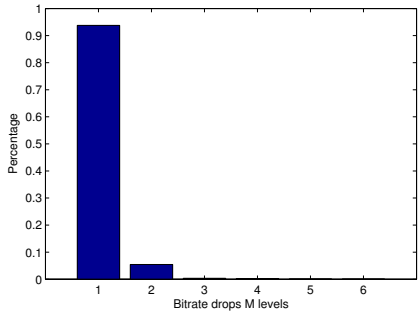


Fig. 4: Distribution of how much bit rate is decreased at a time in retransmissions

correlation coefficient shown in Fig. 5, no strong correlation can be inferred. The correlation coefficient is almost evenly distributed between 0 to 0.8, with a mean about 0.2. This suggests that the bit rates at the two ends of the same link do not necessarily adjust the bit rate in the same fashion, due to their different bit rate adaptation strategies.

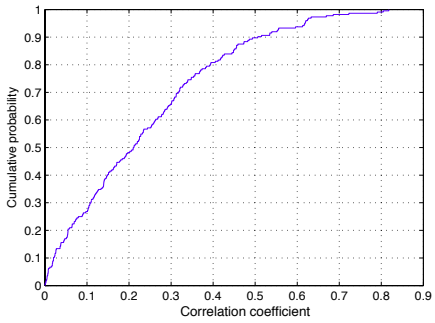


Fig. 5: CDF of correlation coefficient of bit rates

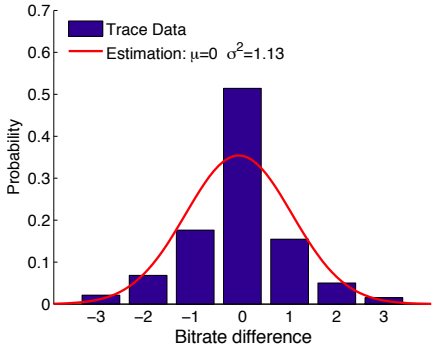


Fig. 6: Distribution of bit rate difference based on 1.2 million records

Bit rate is typically set based on signal-to-noise (SNR) ratio. If two RSS values between a pair are close to each other, their corresponding SNRs should also be similar. Even if the SNR thresholds for bit rates are set differently due to the differences in devices, the bit rates selected should not differ greatly. We then examine the bit rate differences of all transmission pairs. We map 6 Mbps∼54 Mbps to 1∼8 level, and set the maximum bit rate difference to 7. The distribution of bit rate difference over 1.2 million records has been plotted in Fig. 6, which roughly follows the normal distribution, with

an exception that the data is more concentrated in the center. We estimate its mean is 0 and its variance is 1.13. In addition, 50% of transmission pairs have less than one level differences, and another 40% pairs have less than two level differences. In another word, 90% of communicating pairs have less than two level bit rate differences. Thus the reciprocity of bit rates can be summarized as that *bit rates between a communicating pair are not likely to differ significantly.*

Percentage of low bit rate: We are interested in the percentage of low bit rate in all transmitted frames. It is trivial to show its percentage across all traces. Yet, we confirm that the percentage is very low (<1%) in normal data traces. We will, however, explain how to deal with the situation when a channel is dominated with low bit rates in the next.

D. Dealing with Spectrum Misuses

Once an anomaly is detected, the system can identify its source because an anomalous bit rate behavior is always correlated with a specific wireless device. For a device with inappropriate bit rate or transmit power configurations, the device should be limited from accessing the network, or even be removed completely from the network until it is reset properly. For the bit rate and channel-width mismatch, the spectrum allocation server should adjust the width of operational channel to improve the utilization. We now explain how to improve the spectrum utilization.

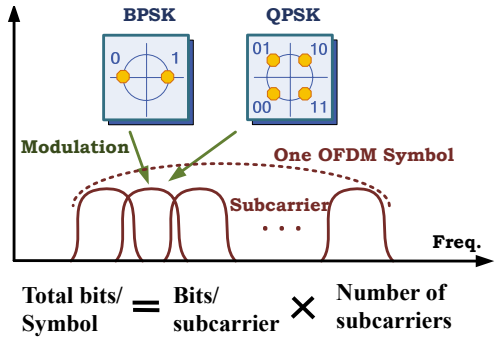


Fig. 7: Total bits over an OFDM symbol

If a user consistently transmits at low bit rate over a channel, the spectrum allocation server should reclaim some of the spectrum from the underutilized channel and re-allocate it to other subnetworks. The problem, however, is how to avoid hurting the performance of the subnetwork with a reduced channel-width. Shrinking the channel-width will improve SNR per Hz. This is because the signal's power is redistributed over a narrower channel, and thus is more concentrated. The improved SNR can be used to compensate the throughput loss due to the reduction in channel-width. As in the well-known Shannon theorem, the channel capacity can be expressed as $C = B\ log(1 + SNR)$. If bandwidth B decreases but SNR increases, channel capacity C may still maintain the same. Halving the channel width, for example, can yield 3 dB SNR gain, provided that transmit power is the same. Communication theory as well as previous experiments have shown that 3 dB SNR gain is sufficient for delivering one more bit per modulated symbol over the air. For example, upgrade BPSK (1 bit/modulated symbol) to QPSK (2 bit/modulated symbol) [15], [19].

In an OFDM system, the relationship between channel-width and modulation can be explained as follows. For a given OFDM system, each OFDM symbol is consisted of a certain number of subcarriers. Assuming no channel coding is used, the number of total bits is the product of the number of subcarriers and the number of bits per subcarrier, as shown in Fig. 7. Without considering other MAC/PHY layer overhead, the PHY layer throughput of an OFDM system can be calculated by dividing the total bits per OFDM symbol by the duration of sending each symbol. For example, if it takes 8ms [2] to send an OFDM symbol containing 48 bits (1 bit per data subcarrier), its throughput is 6 Mbps.

We now use an example to show the improvement on spectrum efficiency. In IEEE 802.11n, a 40 MHz channel is consisted of 108 data sub-carriers, which roughly doubles the number of sub-carriers of a 20 MHz channel (48 data subcarriers). Assuming a 40 MHz channel can only support BPSK modulation and thus its width is shrunk to 20 MHz, the SNR will be increased by 3 dB and the bit rate adaptation will be able to enable QPSK modulation. The total throughput over both channels should be similar, but the spectrum efficiency over the 20 MHz, in terms of Mbps per MHz, is almost doubled. In the context of our application, if low spectrum utilization is detected, the spectrum allocation server should take away half of the spectrum and redistribute it to other users. It is then up to the bit rate adaptation scheme at devices to ensure a proper usage over the narrower channel (e.g., increasing its bit rate).

One question raised naturally is whether *the spectrum efficiency can be further improved in a similar fashion?* Unfortunately, this is not feasible because the 3 dB margin is merely sufficient to add one more bit per modulated symbol. Doubling the throughput from 12 Mbps to 24 Mbps, for instance, needs two more bits per modulated symbol (from QPSK to 16-QAM) and takes extra 6 dB. Therefore the approach only applies to upgrading the low bit rate, such as BPSK, to QPSK or 4-QAM. To validate it, we present the throughput comparison over 10 MHz and 20 MHz in Section IV.

IV. Evaluations

In this section, we evaluate the detection performance of Pinokio system. We use trace-driven simulations to evaluate the effectiveness of the proposed system under various settings. In addition, we conduct SDR platform based experiments to validate the feasibility of shrinking spectrum for spectrum efficiency improvement.

A. Experiment Configurations

(a) **Trace files:** In Section III, we analyze the data trace of SIGCOMM'08 and present the statistical profile for each

[2] A real OFDM symbol is transmitted at a much higher speed. This example is simply for illustration purpose.

feature. To show the applicability of our approach, we use a different data trace - OSDI'06 trace to evaluate Pinokio. OSDI'06 trace is consisted of 54+ million transmission records and 720+ unique MAC addresses. Again, no bit rate anomaly is assumed in the trace.

(b) **WARP:** The feasibility of reducing channel-width and improving spectrum efficiency is validated by using the WARP platform [4]. WARP is an SDR platform. Its radio board is operated at a bandwidth of 10 MHz (as opposed to 20MHz in IEEE 802.11a/g). Its PHY and MAC layer are loosely based on IEEE 802.11g, using the same OFDM technique with standard 52 subcarriers and CSMA/CA protocol. It provides an accessibility to software, firmware and hardware. Therefore, we can easily change the channel-width by adjusting the number of subcarriers.

(c) **Laptops:** Laptops are used for emulating misconfigured devices and for generating misuse trace. The model is HP nc6000, equipped with Intel Pentium M 1.6 GHz processor with 512 MB DDR SDRAM, and HP W500 802.11a/b/g wireless LAN card with an Atheros chipset. The operating system is Linux with kernel version 2.6.25 and WLAN driver is MadWifi (version 0.9.4) [3].

B. *Efficacy of Detections*

Detection speed, detection rate, and false alarm rate are three compromising objectives. Since our detection algorithm uses passive monitoring, the detection speed will depend on the amount of traffic monitored. Pinokio's detection performance is thus evaluated by detection rate and false alarm rate.

Although our system is capable of detecting multiple misconfigurations at a time, in the simulations we have focused on the case that only one type of misconfiguration is present to simplify the presentation of results.

Trace-drive Simulations: We divide all traces into 400 subfiles, each of which contains 3∼5 minutes audit data and 200k∼300k records. Among them, 100 files are used for creating normal statistical profiles and acquiring necessary thresholds for detections. The rest of the 300 files are evenly split into two halves, one for simulating the normal case (to get false alarm rate) and the other for simulating the spectrum misuse (to get detection rate). Misuse traces are generated via experiments on laptops, and then are injected into the normal data trace to produce audit data. Since the detection algorithm only needs the statistical information about bit rate, the timing information of records does not affect the detection results. The airtime utilization after injecting misused data trace can be calculated as follows.

$$\frac{\text{Airtime usage}_{misused} + \text{Airtime usage}_{normal}}{\text{Total time}_{normal}}$$

Where the total time is remained unchanged, and the airtime usage of two parts are summed up. This is feasible because the wireless medium in normal data trace is underutilized.

The more the audit date deviates from the normal statistical profile, the easier Pinokio can detect the anomaly. To create different deviation cases, we inject different amounts of misused trace into the same normal data trace, and term the ratio of misused trace to all data trace as *percentage of misuse*. In reality, the ratio can be changed due to the traffic variations of the misconfigured user.

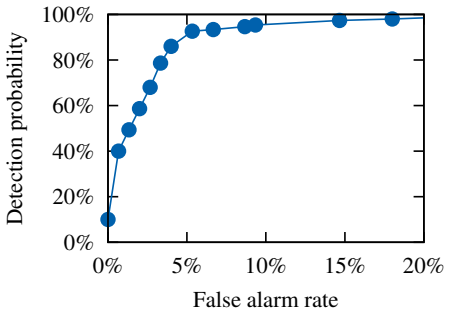


Fig. 8: Conservative bit rate: ROC curve of detection performance when the airtime increases by 20% with 40% of misuse.

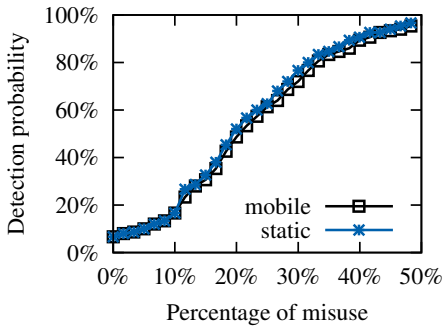


Fig. 9: Conservative bit rate: detection probability versus percentage of misuse.

Metrics: For each type of misuse, we plot the ROC curve when the injected misuse trace is at a certain level such that the false alarm rate is around 5% and the detection rate is around 90% (unless this level of detection performance is unachievable). In practice, this level of detection performance is regarded as acceptable. For the cases where the percentage of misuse is even lower, the airtime inflation is very limited (<20%) even if the detection performance will be degraded as well. In addition, to get a complete view of how system performance varies with the percentage of misuse, we also present a second type of results - detection probability versus percentage of misusage.

1) Detecting Conservative Bit Rate Settings: Over a wireless link that can support 54 Mbps bidirectionally, we intentionally drop the bit rate to 24 Mbps (4 level difference) at one laptop to emulate the conservative bit rate setting. Over the same link, the bit rate adaptation at the other laptop maintains automatic and is observed to use 54 Mbps mostly. To assess the robustness, we also move around the misconfigured laptop at a walking speed (about 1 meter/second).

Fig. 8 shows the ROC curve of dropping the bit rate with 40% of misuse, which roughly increases the airtime by 20%. While maintaining the same false alarm rate of 5%, Pinokio can detect the misuse at the probability of more than 90%. Fig. 9 demonstrates the detection rates across different percentages of misuse. It also indicates that the system performs stably in both static and mobile scenarios. In this case, 50% of the misuse can increase the airtime by 25%.

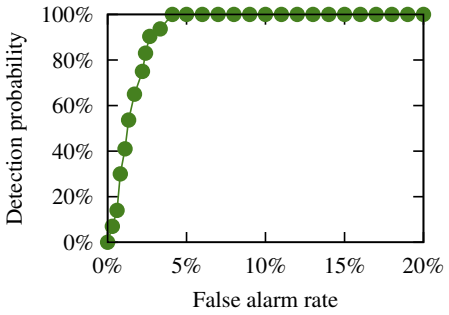


Fig. 10: Aggressive bit rate: ROC curve of detection performance when the airtime increases by 20% with 10% of misuse.

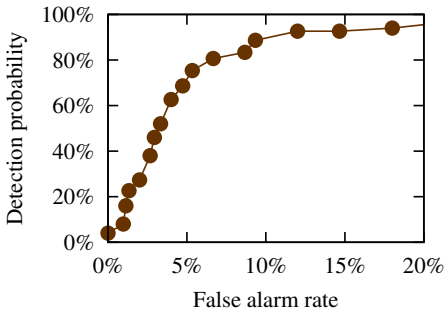


Fig. 12: Low transmit power: ROC curve of detection performance when the airtime increases by 12% with 60% of misuse.

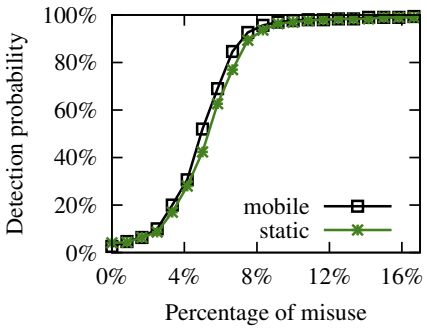


Fig. 11: Aggressive bit rate: detection probability versus percentage of misuse.

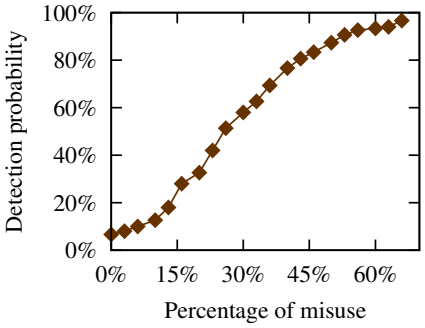


Fig. 13: Low transmit power: detection probability versus percentage of misuse (static only).

2) Detecting Aggressive Bit Rate Settings: Similarly, we fix the bit rate to 54 Mbps at one laptop over a wireless link that can support 24 Mbps to emulate the case of the aggressive bit rate settings, which incurs a large amount of retransmissions. We also limit the maximum retransmission to 3, instead of 7, to create a challenging scenario to detect. In addition, the misconfigured laptop is also moved around at a walking speed to emulate the mobile scenario.

Fig. 10 shows the ROC curve of detecting the aggressive bit rate setting with 10% of misuse. Pinokio can detect the misuse at the accuracy of more than 90% and at the false alarm rate of 3.33%. Fig. 11 demonstrates how the percentage of misuse as well as the mobility affects the detection rate. In this scenario, the airtime can be increased by 20% and 25% at the misuse ratio of 10% and16% respectively. Again, Pinokio's performance is not affected by the mobility.

3) Detecting Low Transmit Power: The transmit power at one of the laptops is lowered from 15 dBm to 10 dBm to emulate the low transmit power case. Yet, its auto-rate adaptation is unchanged. As the link becomes less reliable due to the reduction of transmit power, both the percentage of retransmissions and the percentage of transmissions at low bit rates increase.

Fig. 12 shows the ROC curve of lowering the transmit power with 60% of misuse. Compared with the previous two scenarios, the detection performance degrades slightly. The misuse is detected at the probability of 75% and with the false alarm rate of 5%. The adaptiveness of the bit rate makes the audit data profile deviate less significantly from the normal profile. In addition, the airtime utilization increases less than the previous two cases. Even with 60% of misuse, the airtime is just inflated by 12%, which is almost the half of the previous two cases. Fig. 13 demonstrates how the detection rate varies with the misuse ratios. We do not conduct the mobility test in this case as the channel has already become less predictable due to the decrease in the transmit power.

C. Shrinking Spectrum to Improve Efficiency

We use WARP to study the relationship between bit rate and channel width, for its accessibility to related parameters. Our goal is to verify the proposed strategy in Section II-B, halving spectrum but increasing bit rate will not affect the throughput in some cases. To this end, we vary the channel-width between two WARP nodes and measure their UDP goodput. To minimize the dependence on environment, we also choose multiple indoor locations to repeat the experiment. At

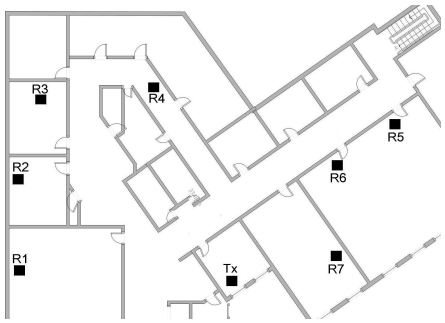


Fig. 14: Measurement map shows node locations. The point marked Tx is a Transmitter. The other points shows the receivers' locations.

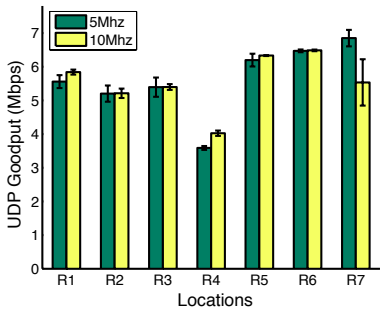


Fig. 15: Goodput comparisons: QPSK over 5 MHz versus BPSK over 10 MHz.

each location, the experiment lasts 5 minutes and is repeated 5 times. Fig. 14 shows the locations of experiments, where the node labeled `Tx` transmits and the rest of the nodes receive.

Fig. 15 shows that QPSK over a 5 MHz channel delivers similar amount of traffic as BPSK over 10 MHz channel does. Particularly, the goodput in both cases are comparable over links with different qualities, from highly reliable channels (R5, R6, and R7) where the goodput approaches to the maximum, fairly well channels (R1, R2 and R3), to poor channel (R4) where the goodput goes down to 2/3 of the maximum. Our results indicate that bit rate and the spectrum can be traded at low bit rates, while the goodput can be maintained almost unaffected.

V. Related Work

Bit rate features. Previously, bit rate behaviors have also been clustered into groups to fingerprint 802.11 devices [13]. Our work differs in that we do not assume any rate adaptation schemes as *priori*. Instead, we apply a statistical method to distinguish a series of anomalous bit rate behavior from normal bit rate behavior.

Channel-width adaptation and allocation. A case of channel-width adaptation between one communicating pair has been made in [7]. A joint spectrum and time block allocation is studied in [22], and a dynamic spectrum access scheme over white space is proposed in [6]. An efficient load-aware spectrum redistributable algorithm for WLANs is shown to improve the overall performance in [14]. In contrast, we do not target at improving the performance of one specific spectrum allocation algorithm, but rather to ensure the accuracy of the metric for spectrum allocations.

Anomaly detection in IEEE 802.11. Anomaly in 802.11 is referring to the medium access unfairness among contending nodes [10]. Nodes using low bit rates can occupy a channel much longer than those employing higher rates, and thus degrade the performance of local network. Instead, we are trying to detect the bit rate misuse that can potentially hurt the performance globally.

VI. Conclusion

In this study, we identify a new vulnerability in the spectrum reallocatable WLANs. Airtime utilization, as an important metric for spectrum redistribution, can lead to an unfair spectrum allocation if not examined carefully. We demonstrate a series of spectrum misuses as well as their impacts on the system performance. We also present Pinokio, a system that monitors spectrum usage of clients, detects spectrum misuse, and improves spectrum efficiency. Based on a widely accepted statistical algorithm, we incorporate four key features of bit rate behavior for anomaly detections and propose corresponding strategy to deal with anomalies. We evaluate the efficacy of Pinokio extensively, and our results show that Pinokio can successfully detect inflation of airtime utilization with high accuracy and low false alarm rate in both static and mobile scenarios.

References

[1] Aruba's enterprise wlan solution. http://www.arubanetworks.com/pdf/solutions/AB_ENT.pdf.
[2] Intel wimax/wifi link 5350 and link 5150 product briefs. http://www.intel.com/support/wireless/wmax/5350_5150/sb/CS-029594.htm.
[3] Madwifi project. http://MadWifi.net.
[4] Rice university warp project. http://warp.rice.edu.
[5] M. Arslan, K. Pelechrinis, I. Broustis, S. Krishnamurthy, S. Addepalli, and K. Papagiannaki. Auto-configuration of 802.11 n WLANs. In *ACM CoNEXT*, 2010.
[6] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh. White space networking with Wi-Fi like connectivity. In *ACM SIGCOMM*, 2009.
[7] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl. A Case for Adapting Channel Width in Wireless Networks. In *ACM SIGCOMM*, 2008.
[8] C.-T. Chou, K. G. Shin, and S. Shankar N. Contention-Based Airtime Usage Control in Multirate IEEE 802.11 Wireless LANs. *IEEE/ACM Transactions on Networking*, 2006.
[9] R. Gummadi and H. Balakrishnan. Wireless networks should spread spectrum based on demands. In *HotNets*, 2008.
[10] M. Heusse, F. Rousseau, G. Berger-sabbatel, and A. Duda. Performance Anomaly of 802.11b. In *IEEE INFOCOM*, 2003.
[11] H. Javitz and A. Valdes. The SRI IDES statistical anomaly detector. In *IEEE Computer Society Symposium on Research in Security and Privacy*, 1991.
[12] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *ACM MOBICOM*, 2008.
[13] M. Mirza, P. Barford, X. Zhu, S. Banerjee, , and M. Blodgett. Fingerprinting 802.11 rate adaptation algorithms. In *IEEE INFOCOM*, 2011.
[14] T. Moscibroda, R. Chandra, Y. Wu, S. Sengupta, P. Bahl, and Y. Yuan. Load-aware spectrum distribution in Wireless LANs. In *IEEE ICNP*, 2008.
[15] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini. Frequency-aware rate adaptation and MAC protocols. *ACM MobiCom*, 2009.
[16] T. Rappaport. *Wireless Communications: Principle and Practice, 2nd edition*. Prentice Hall, 2001.
[17] A. Schulman, D. Levin, and N. Spring. Dataset of wireless network measurement in the SIGCOMM 2008 conference. http://crawdad.cs.dartmouth.edu/umd/sigcomm2008, Mar. 2009.
[18] G. Tan and J. Guttag. Time-based fairness improves performance in multi-rate WLANs. In *USENIX Annual Technical Conference*, 2004.
[19] J. H. J. Terry. *OFDM Wireless LANs:A Theoretical and Practical Guide*. Sams Publishing, 2001.
[20] L. Yang, W. Hou, L. Cao, B. Zhao, and H. Zheng. Supporting demanding wireless applications with frequency-agile radios. In *USENIX NSDI*, 2010.
[21] Yiping Xing. Dynamic spectrum access in open spectrum wireless networks. *IEEE JSAC*, 2006.
[22] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, and Y. Wu. Allocating dynamic time-spectrum blocks in cognitive radio networks. In *ACM MobiHoc*, 2007.
[23] K. Zeng, D. Wu, A. Chan, and P. Mohapatra. Exploiting multiple-antenna diversity for shared secret key generation in wireless networks. In *IEEE INFOCOM*, 2010.