IQ-Hopping : Distributed Oblivious Channel Selection for Wireless Networks

Apurv Bhartia[1], Deeparnab Chakrabarty[2], Krishna Chintalapudi[2], Lili Qiu[3], Bozidar Radunovic[2], and Ramachandran Ramjee[2]

[1]Meraki Networks [2]Microsoft Research [3]Univ of Texas, Austin

ABSTRACT

Interference in WiFi deployments is a growing problem due to the increasing popularity of WiFi. Therefore it is important that APs find the right channel to operate upon. Through a large scale measurement study involving over 10,000 WiFi APs we show that channel measurements and selection are most effective when performed frequently (every few minutes). This is because of the highly dynamic nature of WiFi traffic congestion. Our key contribution in this paper is a novel approach to distributed channel selection – Ineffective time Quantum (IQ) Hopping, that is simple enough to be described in three lines and has provable optimality guarantees. IQ-Hopping does not require any explicit channel measurements and can react within a matter of several seconds to bad channel conditions, including microwave ovens, hidden interferers, or dynamically varying congestion. Through implementation and experiments on off-the-shelf WiFi routers (OpenWRT, MadWiFi), we demonstrate the effectiveness of IQ-Hopping.

CCS Concepts

•Networks → Wireless access points, base stations and infrastructure; Wireless local area networks;

1. INTRODUCTION

WiFi is ubiquitous, and found in almost all airports, cafes, malls, offices, etc. Recognizing that WiFi carries about 70%-85% of mobile internet data traffic today [3], most major cellular operators and ISPs such as AT&T, British Telecom, Comcast, China Mobile *etc.*are deploying millions of WiFi hotspots to provide cheap broadband access to outdoor users. It is predicted that there will one WiFi hotspot for every 20 people in the world by 2018 [1].

As a consequence of this mushrooming, a typical WiFi AP has to share the spectrum with a large number (5-30) of neighboring APs (Section 2). While some large businesses use centralized controllers to manage APs, most small and medium offices deploy controller-less APs due to their low cost [2]. Thus, in offices, homes and cafes, *the vast majority of APs are deployed and used independently.* Consequently, in order to best utilize available wireless channels, the problem of distributed channel selection in WiFi has been extensively studied [12] and asks the question *which channel should an Access Point (AP) choose to operate on without any centralized controller managing them?*

Almost all existing research in distributed channel selection falls into two classes (Section 6), those that perform explicit coordination by exchanging messages between APs or those where each AP periodically scans through all channels to measure congestion levels and chooses the least congested channel independently. The former schemes are not implemented in APs today since explicit coordination among autonomous APs requires incentives and standardization. Thus, the dominant practice today for most deployed APs is to select the least occupied channel using scanning at the time they are turned on or once a day (e.g., APs in our measurement study). Some existing research also suggests performing scans once every half hour [20], though this trades off increased measurement overhead for timeliness.

Large scale measurement study. In order to understand the nature of WiFi congestion, we conduct a large scale measurement study of over 10,000 APs deployed across various U.S retail stores, hospitals, WiFi hotspots and offices. By measuring minute by minute channel measurements for each of these APs, we arrive at the following conclusions. *Moderate to severe congestion exists in 2.4 GHz WiFi about 10% of the time. In these busy times, congestion is dynamic and measurements become obsolete within few minutes. Significant gains (60-80%) can be obtained for these busy APs by switching to the least congested channel every minute.*

Ineffective-time Quantum Hopping (IQ-Hopping). *The key contribution of this paper is IQ-Hopping, a novel and simple approach to channel selection that neither requires message exchanges nor channel scanning measurements.* In IQ-Hopping each AP continuously and independently performs the following three actions:

- **Step 1.** Hop to a random channel and generate a random (exponentially distributed) time quantum, τ.
- **Step 2.** While transmitting/receiving packets, keep track of *ineffective-time* ($t_{\mathtt{ineff}}$) and *effective time* ($t_{\mathtt{eff}}$). Here, $t_{\mathtt{ineff}}$ is the total time wasted since last channel change when the AP had packets to transmit but was being unsuccessful. This could be due to contention, collision, packet loss or even microwave oven interference. $t_{\mathtt{eff}}$ is the total time spent in successful packet transmissions.
- **Step 3.** When $\Gamma(\phi)(t_{\mathtt{eff}} + t_{\mathtt{ineff}}) > \tau$ where $\phi = \frac{t_{\mathtt{eff}}}{t_{\mathtt{eff}} + t_{\mathtt{ineff}}}$ choose a channel randomly, inform clients of impending channel change and go to Step 1. [1] Γ is a monotonically decreasing function, whose choice is discussed in detail in Section 3. In this paper, we use $\Gamma(\phi) = 3^{-10\phi}$.

[1]Note that $t_{\mathtt{ineff}} + t_{\mathtt{eff}}$ is not equal to the total time elapsed since there may idle times when there are no packets to transmit.

 ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

The key intuition behind IQ-Hopping's design is that ϕ is high in good channels and low in poor channels. Thus, an AP will stay in a good channel for a long time (several minutes) and hop out of poor channels quickly (within a few seconds). In this paper, we use $\tau = 1s$. This implies that at a congestion level of 0.8, a busy AP will switch channels in 9s, thereby, reacting quickly to congestion.

IQ-Hopping fundamentally differs from existing channel selection schemes in two ways. First, APs and clients can be oblivious of their neighbouring networks and are not required to exchange messages or periodically scan channels to estimate congestion. Second, rather than deciding on the best channel, APs only decide when to leave the current channel and then simply choose a random channel. Besides its simplicity, IQ-Hopping has several key advantages:

Easy to implement on off-the-shelf WiFi routers without client side modifications. We have implemented and tested IQ-Hopping in OpenWRT and MadWifi based APs (Section 4). APs keep track of $t_{\texttt{ineff}}$ and $t_{\texttt{eff}}$ based on ACK and packet receptions and notify channel changes to clients by using 802.11h-based channel switch announcements, thus, enabling robust deployment with unmodified clients. We show that the cost of switching channel in modern wireless routers is modest – UDP average loss rate of at most 1.3%; TCP throughput reduced for at most 1-2 seconds (Section 5).

Provable self-organizing properties for multiple IQ-Hopping APs similar to graph colouring algorithms. When multiple IQ-Hopping APs co-exist and interfere with each other, we provide theoretical guarantees backed by simulations, that they will self-organize quickly and find channel assignments similar to centralized graph colouring algorithms. This is remarkable since IQ- Hopping APs achieve this without any knowledge of the interference graph or message exchanges between APs, as required by graph colouring schemes. Thus, when a transient congestion bottleneck occurs in a channel, APs react by spreading out across free channels to dissipate congestion bottleneck.

Robustness to immeasurable wireless effects. There are several wireless effects that are hard to predict and are not captured through passive channel scanning measurements *e.g.,*non-WiFi interference such as microwave ovens, hidden terminals, capture effects, channel fading, *etc.*In IQ-Hopping, APs measure and react to the overall time wasted irrespective of the underlying reasons. Consequently, IQ-Hopping automatically takes all these effects into account without incurring any overhead for channel scanning.

Sub-Minute adaptation to changing congestion and channel conditions. As demonstrated in (Section 5), IQ-Hopping APs can adapt in a matter of several seconds to non-WiFi sources of interference (*e.g.,*microwave ovens) and to changing WiFi traffic conditions. We also formally show that for interference graphs (disc graphs), IQ-Hopping has very fast convergence (Section 3).

Summary of contributions in this paper.

- A large scale measurement study comprising over 10,000 WiFi APs to show that channel selection should be performed frequently (in the order of minutes) during busy times.
- Ineffective-time Quantum Hopping: a novel approach to channel selection that does not require any channel measurements or inter-AP communication and performs well even in the face of hidden interferers and dynamic traffic conditions.
- Formal proofs showing that multiple co-existing IQ-Hopping APs can self-organize on par with centralized graph colouring schemes and also achieve fast convergence.
- Validation of IQ-Hopping through implementation and experimentation on off-the-shelf routers (OpenWRT, MadWiFi).

2. LARGE SCALE MEASUREMENT STUDY

A large number of channel selection schemes proposed in the WiFi literature are based on the measuring the number of interferers and congestion observed in various channels [12]. In this section, we conduct a large scale measurement study involving 10766 802.11n WiFi hotspots deployed in varied sectors such as retail stores, cafes, malls, schools, offices and hospitals spread across various locations in the U.S and answer the following questions:

- Does congestion exist in WiFi deployments today?
- How many interferers do WiFi APs typically see?
- Does a large number of interferers imply high congestion?
- If a channel is congested, how likely is it that a less congested channel is also available at the same time to switch to?
- How often should we make channel selection measurements and decisions and how much can we potentially gain?

2.1 Measurement Methodology

For our measurement study we used 10766 WiFi 802.11n APs, each equipped with two extra dedicated radios that continuously scan across all the 2.4 GHz and 5 GHz channels, spending 150ms in each channel[2]. The number of interferers is measured by decoding all WiFi beacons collected during the preceding 3 minute time window and counting the unique BSSIDs observed. For measuring congestion, we use air-time utilization counters provided by the WiFi chipsets that count the ticks during which the medium was found busy as a part of CSMA. Thus, congestion in a channel is measured as the fraction of time when there was enough energy in the channel (or preamble was detected) to cause the AP to defer transmission. A single congestion measurement is the average of last 10 measurements (spanning 1.5 seconds) in each channel.

The entire data deemed *ALL* comprises three sets of data collected on different days. *CAFE* was collected from 908 APs deployed in two popular fast-food chains in the U.S. over four consecutive days, Thursday through Sunday, for a total of 38Hrs during daytime. *OFFICE* was collected from 73 APs in a three floor office building with 700 employees on a Thursday and a Monday from 11:00 A.M to 7:00 P.M. Finally, *LARGE* was collected from 9785 APs between 8:00 A.M to 8:00 P.M on a Monday. Overall *ALL* comprised a total of 127,271 AP-Hours of data.

2.2 Congestion Levels

While at most locations and hours congestion levels may be moderate or low, channel selection matters the most during the busiest times. In order to understand, how high congestion levels can be during the day, in Figure 1 we plot the distribution of top 10% busiest times (based on average congestion across all channels over the entire day) from *ALL*, *CAFE* and *OFFICE* in 5 GHz and 2.4 GHz[3]. As seen from Figure 1, while 5 GHz channels are largely underutilized, 2.4 GHz channels experience moderate (40-70%) and severe levels (>70%) of congestion in the *ALL* and *CAFE*. Further, congestion can be extreme (80-90%) in the 2.4 GHz channels at certain pockets such as *OFFICE*. Even 5 GHz channels in *OFFICE* show moderate to high levels of congestion. It is interesting that even during high congestion levels in 2.4 GHz, clients do not switch to the lighter loaded 5 GHz channels in both *OFFICE* and *ALL* – perhaps due to the range limitations of 5 GHz channels that forces clients to remain in the congested 2.4 GHz band.

[2]The scanning radios skip the channel that the AP is operating on in order to avoid saturation and instead obtain data for the operating channels directly from the data radios.

[3]We could not obtain the 5 GHz measurement data for *CAFE*.

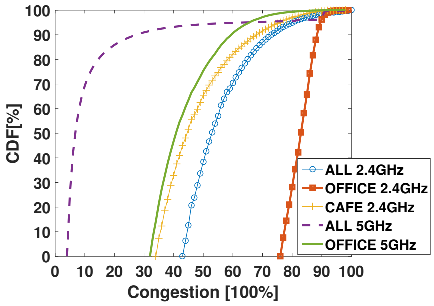

Figure 1: Distribution of congestion in *ALL*, *OFFICE*, *CAFE* in 2.4 GHz and 5 GHz at top 10% busiest times.

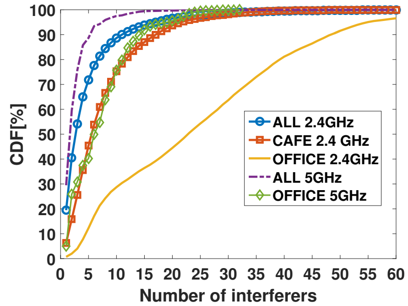

Figure 2: Distribution of number of interferers in *ALL*, *OFFICE*, *CAFE* in 2.4 GHz and 5 GHz.

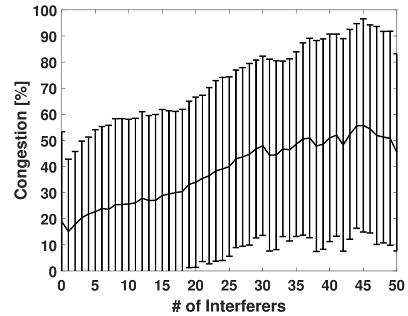

Figure 3: Distribution of congestion for different number of interferers for *ALL*.

2.3 Number of Interferers

Figure 2 depicts the distribution of the number of interferers for *ALL*, *CAFE* and *OFFICE* data sets. As seen from Figure 2, in 2.4 GHz, 90% of all the APs (*ALL*) have 10 or fewer neighbours, Cafes (*CAFE*) have about 18 neighbours or less while the office (*OFFICE*) has significantly larger number of neighbours at 50. The number of neighbours in the 5 GHz is typically less by a factor between 2-3, probably due to range limitations in the 5 GHz band.

2.4 Number of Interferers vs congestion

A large number of channel selection schemes use number of interferers (neighbours) as a measure of degree of congestion. In order to understand how valid this measure is, we measure the correlation between congestion and the number of interferers in the *ALL* data set. Figure 3 depicts the dependence of congestion and its variation (error bars at 85%ile value) as a function of number of interferers seen over *ALL*. As seen from Figure 3, there is an increase in average congestion levels from 20% to 40% as the number of neighbors increases from 1 to 40. However, the range of congestion values seen for a given number of interferers is also very large. For example, if the observed congestion level were 30%, the number of interferers could be any value between 1-50. Thus, number of neighbours may not be a good measure of congestion.

2.5 Opportunities for channel selection

If all channels are equally congested, switching channels will not provide any benefit. How often is the case that when the current channel is congested, there is another free channel that could potentially be used? In order to answer this question, we find the average difference in congestion levels between all channels at a certain minute and the least congested channel in that minute. This difference corresponds to the average congestion reduction that could be achieved by an AP switching its channel to the least congested one. Figure 4 depicts the mean congestion reduction for channels in various ranges of congestion levels. For *ALL* and *CAFE* the mean reduction in congestion level, even when the congestion in a channel is severe (>70%), is about 50%. This means that in most cases, there exists at least one channel with much lower congestion than the congested channel. Even in the severely congested *OFFICE*, an average congestion reduction of 30% is possible when a channel is highly congested. This indicates that there are opportunities for channel selection to significantly reduce congestion.

2.6 How quickly must we change channels?

How quickly does one need to find and change to the least congested channel in cases of high congestion? In order to answer this question we consider an oracle that chooses the least congested channel each minute. However, to avoid frequent channel changes due to small changes in congestion levels the oracle switches channels only if the reduction in congestion level is greater than $\Delta\mu$. We then define run-length $R_{\Delta\mu}$ of the duration for which the oracle does not change channels. Figure depicts the CDF of various run lengths in minutes for various values of $\Delta\mu$ for 10% of the busiest WiFi APs (based on average utilization over the entire day over all channels). As seen from Figure 5, the best channel decision becomes obsolete within 4 minutes 80% of the time. With decision being made every 10 minutes, we lose out on opportunities where congestion would have been 20% less about half of the time. *Thus, quick and frequent channel selection decisions, in the order of a few minutes, can lead to significant gains by avoiding congestion.*

2.7 Potential gains of switching channels

It is in general extremely hard to determine the exact throughput gains of a channel selection scheme without performing active measurements. In this section we quantify the potential gains based on *free time* which is simply (1 - measured congestion level). Free time is the upper limit of time that a device could potentially use for communication given the current level of congestion.

As baseline, we use the prevalent practice of selecting the least congested channel once when the AP is turned on or at the beginning of the day (e.g., all APs in our measurement data did not change their channel during the entire day). Next we consider seven different channel switching strategies. The first four strategies comprise switching to the least congested channel once every 5, 10, 30 and 60 minutes (labeled as Least Cong 5-60 in Figure 6). While in practice, these schemes will incur various measurement costs, in this analysis, we assume zero measurement overhead for these schemes. The fifth scheme chooses the channel with the minimum number of interferers once every 5 minutes (changing this interval to 10, 30, 60 minutes did not alter the results significantly). Scheme labeled OPT (Figure 6) computes gains of an oracle that uses the least congested channel every minute. Finally, we compute the gains from IQ-Hopping (labeled IQHop in Figure 6).

For each of these strategies, Figure 6 depicts the average percentage gains obtained over the baseline for various levels of average congestion (over all channels) for *ALL*. Recall that the top 10% busiest times for *ALL* comprise 12,727 AP-hours and have congestion levels of 40% or higher (Figure 1). As seen from Figure 6, for these moderate to high congestion levels (40%+), most of the schemes provide gains of 20% or more in free time, thereby demonstrating the value of channel selection. The oracle (OPT) gains up to 80% over baseline during high congestion times. The least congested channel schemes achieve gains of 30-40%, with the 5 minute version performing the best (albeit, assuming zero mea-

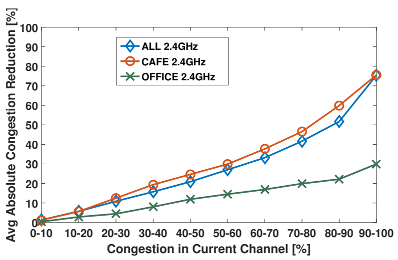

Figure 4: Average congestion level reduction opportunity for various levels of congestion in current channel.

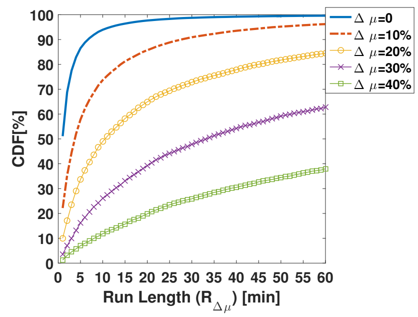

Figure 5: Distribution of how long the best channel decision is valid in busy scenarios.

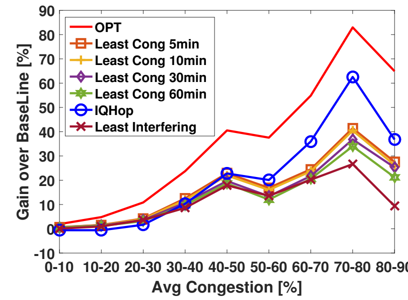

Figure 6: Potential gains in free time for *ALL* for various congestion levels.

surement overhead). Channel selection based on least number of interferers provides gains of only up to 20%. Finally, our proposed scheme, IQ-Hopping, outperforms all measurement based schemes during moderate to high congestion levels with gains up to 60% and comes closest to the performance of the oracle.

2.8 Summary

In summary, we arrive at the following conclusions.

- Moderate to severe congestion scenarios do exist in 2.4 GHz about 10% of the time (this might increase significantly in the future).
- Often when one channel is congested there will typically be another channel that has significantly lower congestion.
- Traffic is dynamic and the least congested channel changes in a matter of few minutes.
- Channel selection schemes that can rapidly find and switch to good channels can increase available free time by up to 60%.

3. IQ-HOPPING

In this section we describe IQ-Hopping – a distributed channel selection scheme where APs move away from congestion bottlenecks and spread over uncongested channels in under a minute. IQ-Hopping does not require APs to make any channel congestion measurements or coordinate with other APs or snoop ongoing traffic to determine the number of interferers, thus making it easy to implement and deploy. Further IQ-Hopping APs even react to hidden interferers such as microwave ovens and dynamic channel conditions – effects that are hard to measure.

We start by describing the essence of IQ-Hopping. The goal being to explain the idea and provide proofs of convergence and performance optimality. Later in section 4 we describe the details of the implementation and discuss other practical issues.

3.1 The IQ-Hopping Algorithm

The pseudo-code for IQ-Hopping is provided in Algorithm 1 which is an expanded version of the three steps described in Section 1.

Step 1. [Lines 2,3] In IQ-Hopping, APs always select a new channel randomly from the set of available channels, $Channels$ (Line 2). Upon arriving at a new channel, the AP generates a random deadline τ for itself drawn from an exponentially distributed random variable (Line 3). The mean of the exponential distribution determines how long an AP is willing to tolerate a congested channel. We discuss how to choose τ later.

Step 2. [Lines 4-16] The AP keeps track of ineffective time $t_{\texttt{ineff}}$ and effective time $t_{\texttt{eff}}$ during it normal course of transmitting and receiving packets as follows. Let $\Psi(P)$ represent the effective time

Algorithm 1 Pseudo-code for IQ-Hopping

1: **repeat**
2: HopToRandom($Channels$)
3: $\tau = Exp\,(\lambda)$
4: $t_{\texttt{ineff}} = 0$
5: $t_{\texttt{eff}} = 0$
6: Initialize a timer to fire once every δ
7: **repeat**
8: Upon each timer fire event
9: **If** Packets in Queue **then**
10: $t_{\texttt{ineff}} = t_{\texttt{ineff}} + \delta$
11: **end if**
12: On every successful reception of ACK for packet (P) event
13: $t_{\texttt{ineff}} = t_{\texttt{ineff}} - \Psi(P)$
14: $t_{\texttt{eff}} = t_{\texttt{eff}} + \Psi(P)$
15: On every successful data packet (P) reception event
16: **If** Packets in Queue **then**
17: $t_{\texttt{ineff}} = t_{\texttt{ineff}} - \Psi(P)$
18: $t_{\texttt{eff}} = t_{\texttt{eff}} + \Psi(P)$
19: **end if**
20: **until** $(t_{\texttt{ineff}} + t_{\texttt{eff}})\Gamma(\phi) > \tau$, $\phi = \frac{t_{\texttt{eff}}}{t_{\texttt{ineff}}+t_{\texttt{eff}}}$
21: Inform clients of impending channel change
22: **until** true

in the transmission of a packet, that is, it is the time from when the packet is sent for transmission until the receipt of its ACK assuming no congestion or packet losses. By default, the AP keeps incrementing $t_{\texttt{eff}}$ periodically through a timer (Lines 7-10). Note that the AP only increments $t_{\texttt{eff}}$ if there are packets in the queue, as no time is being wasted in the absence of any packets to transmit.

Upon receiving an ACK for a packet P that it transmitted in the past, it subtracts $\Psi(P)$ from $t_{\texttt{ineff}}$, since this amount of time was actually effective transmission time (Lines 11-12) and adds it to $t_{\texttt{eff}}$. It does the same upon receiving a packet since the time spent in receiving a data packet P successfully $\Psi(P)$ is also effective time. Hence, if the AP had been incrementing $t_{\texttt{ineff}}$ since there were packets in the queue, to account for the effective time during reception of P, it subtracts $\Psi(P)$ from $t_{\texttt{ineff}}$ (Lines 13-16).

Thus, in an ideal world with a perfect channel, *i.e.,* without congestion or packet losses, $t_{\texttt{ineff}}$ will become zero after each reception of an ACK or a data packet. This is because the net increment in $t_{\texttt{ineff}}$ since the time the packet was transmitted will be exactly equal to $\Psi(P)$ which will then be subtracted after the reception.

Step 3. [Line 17-18] If there is congestion in the channel or packet losses are occurring, $\phi = \frac{t_{\texttt{eff}}}{t_{\texttt{ineff}}+t_{\texttt{eff}}}$ will keep decreasing. The function $\Gamma(\phi)$, explained below, is a non-linear monotonically decreasing function of ϕ and is designed to increase rapidly as ϕ decreases. Thus, as more and more packets are lost, $\Gamma(\phi)(t_{\texttt{ineff}} + t_{\texttt{eff}})$, increases beyond the parameter τ; the AP then selects another random channel, announces the impending channel hop to its clients and then hops.

3.2 Intuition and choice of $\Gamma(\phi)$

Consider two channels 1 and 2, first with a congestion level of u_1 and the other with $u_2 > u_1$ respectively. An oracle channel switching algorithm will quickly choose channel 1. Suppose, an IQ-Hopping AP is in channel 2, its $t_{\texttt{ineff}}$ will increase quickly in comparison to $t_{\texttt{eff}}$ and so as described above it will hop out of channel 2 quickly. Once it arrives in channel 1, $t_{\texttt{ineff}}$ will increase slowly in comparison to $t_{\texttt{eff}}$ since the congestion is low. Thus, the AP will spend a larger amount of time in channel 1 before it hops back to channel 2. As this process continues, the AP hops between channels 1 and 2 but spends only a small amount of time in 2 each time and thus approximates the oracle.

The ratio of how long an AP spends in a worse channel in comparison to a better channel, depends on the function Γ. Suppose that τ is 1 second. Further, suppose that the AP gets $\phi = 1 - u_1$ in channel 1 and $1 - u_2$ in channel 2. Then the time that AP stays in channel 1 before hopping will be $\Gamma(1-u_1)^{-1}$ and that for channel 2 will be $\Gamma(1-u_2)^{-1}$. Thus, the ratio of times the AP spends in channel 1 and 2 will be $\frac{\Gamma(1-u_2)}{\Gamma(1-u_1)}$. The larger this value, the closer the performance of IQ-Hopping will be to that of the oracle.

Exponential functions have the property that $\frac{\Gamma(\phi+c)}{\Gamma(\phi)}$, is constant (here c is a constant). Thus, irrespective of their actual congestion values, two pairs of channels with the same difference in congestion will see the same ratio of stay times. Thus, we chose Γ as an exponential function to keep IQ-Hopping performance similar over a wide range of congestion values. One drawback of using exponential functions is that they tend to grow very quickly leading to very large stay times, resulting in slow adaptability at lower congestion values. We chose $\Gamma(\phi) = 3^{-10\phi}$. With τ as 1sec this leads to stay times of over 100 minutes in a channel with congestion levels below 20%, but only 9 seconds when the congestion level is 80%. This means that while the AP will move out of highly congested channels quickly, it will take two hours to move from a channel with utilization 20%. However, for most application performance needs, since moving out of highly congested channels is more important, IQ-Hopping's stickiness at low congestion levels is not a significant issue.

3.3 Self-Organizing Properties of Multiple Sharing IQ-Hopping APs

When multiple IQ-Hopping APs co-exist and transmit packets simultaneously, they tend to self-organize and spread out among channels in a manner so as to relieve transient congestion bottlenecks. Specifically, we rigorously prove two important properties of sharing IQ-Hopping APs.

P1 : Multiple IQ-Hopping APs in an interference graphs can self-organize as well as centralized graph coloring algorithms. If there are N APs arranged in an interference graph with Δ as the maximum degree and $N \leq \Delta+1$. Then IQ-Hopping APs will self-organize to find a $\Delta+1$ colouring and avoid congestion completely. This is close to what any centralized algorithm [16] can do, and is the benchmark for all known distributed, synchronous, message-passing coloring algorithms [9]. This is remarkable because the IQ-Hopping protocol is decentralized, asynchronous, and doesn't pass any messages.

P2 : In a contention domain, when the number of channels is less than the number APs, the APs self-organize to share all the channels in a fair manner. Suppose the number of channels $K < N$, the number of APs. Then a fair channel assignment should be one where each AP transmits roughly K/N fraction of the time. Our second result is that in this scenario, IQ-Hopping converges to a steady distribution where each AP transmits roughly K/N fraction of the time. This is remarkable since no static assignment can guarantee this for simultaneously *all* APs.

These two properties are important in highly congested scenarios since this means that when a transient congestion bottleneck is formed, IQ-Hopping APs will spread out evenly and fairly, thus dissipating the bottleneck. The detailed proofs of these statements are provided in the section below.

3.3.1 Theoretical Proofs

THEOREM 1. *In any N-node graph of max-degree Δ, IQ-Hopping finds a $(\Delta+1)$-coloring in expected $O(N\Delta)$-hops.*

PROOF. Given any coloring (ie channel assignment), call an edge (u, v) "bad" if both the endpoints of the edge have been assigned the same color. Let Φ_t denote the number of bad edges in the graph after t hops have occurred. Note that if $\Phi_t = 0$ for any t, then $\Phi_s = 0$ for all $s > t$. That is, once IQ-Hopping finds a interference-free assignment, it remains in it forever. Also note that $\mathbf{E}[\Phi_0] \leq N/2$ if initially every one AP a random channel – for every edge, the probability both end points choose the same color is $\frac{1}{\Delta+1}$, and the number of edges is at most $N\Delta/2$.

Let $\tau_{\texttt{stop}}$ be the earliest t at which $\Phi_t = 0$; we are interested in upper bounding $\mathbf{E}[\tau_{\texttt{stop}}]$. To do so, let us first evaluate $\mathbf{E}[\Phi_{t+1} \mid \Phi_t]$ when $\Phi_t > 0$. Let v be the node which hops at the $(t+1)$th time step. Without loss of generality we may assume v is assigned channel 1 before hopping. Let n_i denote the number of neighbors of v which have been assigned channel i. Note that $n_1 \geq 1$ since otherwise v won't hop. Also note that

$$\sum_{i=1}^{\Delta+1} n_i = \texttt{deg}(v) \leq \Delta \tag{1}$$

If v hops to channel i, then the number of bad edges will decrease by n_1 (one for each neighbor of v assigned channel 1) and will increase by n_i (one for each neighbor of v assigned channel i). Thus, the drop in the number of bad edges is precisely $n_1 - n_i$. Since v chooses a channel uniformly at random, we get that the expected drop is $\frac{1}{\Delta+1}\sum_{i=1}^{\Delta+1}(n_1 - n_i) = n_1 - \frac{\sum_{i=1}^{\Delta+1} n_i}{\Delta+1} \geq n_1 - \frac{\Delta}{\Delta+1}$, where the inequality follows from (1). Since $n_1 \geq 1$, the drop is at least $\frac{1}{\Delta+1}$. To summarize, we get

$$\mathbf{E}[\Phi_{t+1}|\Phi_t] = \begin{cases} \Phi_t - \frac{1}{\Delta+1} & \text{if } \Phi_t > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

A standard argument then shows that $\mathbf{E}[\tau_{\texttt{stop}}] \leq \mathbf{E}[\Phi_0](\Delta+1) \leq N(\Delta+1)/2$. □

THEOREM 2. *Consider IQ-Hopping on an N-node contention domain with K channels. The number of APs utilizing any channel converges to a bell-shaped distribution with mean N/K and standard deviation $\leq \sqrt{N/K}$.*

PROOF. Throughout the proof we use the notion of time which increases by 1 unit every time some AP hops from one channel to another. Let X_i^t denote the number of APs in channel i at time t. Note that this is a random variable and we are interested in the stationary distribution (if any) of this as t grows.

Note that between time t and $t+1$, in a contention domain we can model $t_{\texttt{ineff}}$ and $t_{\texttt{eff}}$ of any node v as follows: if v is utilizing channel i at time t, $t_{\texttt{ineff}} = (1 - 1/X_i^t)$ and $t_{\texttt{eff}} = 1/X_i^t$. For theoretical convenience, in this idealized model, we prove the theorem for a modification of IQ hopping where $\Gamma(\phi) = 1 - \phi = 1 - 1/X_i^t$, which is a crude approximation to $\exp(-\phi)$. The analysis for this case is simpler for a technical reason (in particular, in Claim 1 below the probability p_i^t decouples and depends only on i). Furthermore, it can be shown (the precise math is beyond the scope of this submission) that this modification only can hurt fairness; the intuition is that $\exp(-\phi)$ is more *aggressive* than $1-\phi$. In fact, for $\Gamma(\phi) = 1-\phi$, we show that the number of APs utilizing any channel converges precisely to the Binomial distribution with mean N/K and std dev $\leq \sqrt{N/K}$.

Let A_t be the channels with $X_i^t \geq 1$, that is, all the channels which have at least one AP utilizing it at time t. Observe that if a channel enters A_t then it never leaves it. This is because an AP hops only more than one AP is in the same channel. Therefore a simple coupon-collector argument [23] shows that in expected $O(K \log K)$ hops, each channel gets utilized. Therefore, we may assume $|A_t| = K$.

Let v_t be the AP which hops at time t. Let p_i^t be the probability that v_t was in channel i. Since the idle times are initialized to exponential random variables, we can precisely figure out p_i^t as stated in the following claim.

CLAIM 1. $p_i^t = \frac{(X_i^t - 1)}{\sum_{i \in A_t}(X_i^t - 1)} = \frac{X_i^t - 1}{N-K}$

PROOF. This proof crucially uses the following properties of exponential random variables; below $X_i \sim \exp(\lambda_i),\ i = 1, \ldots, k$.

1. For any $c > 0$, $c \cdot X_i \sim \exp(\lambda_i / c)$.
2. $\min(X_1, \ldots, X_k) \sim \exp(\lambda_1 + \cdots + \lambda_k)$.
3. $\Pr[X_i = \min(X_1, \ldots, X_k)] = \lambda_i / \sum_{j=1}^{k} \lambda_j$.

Consider the situation after the $(t-1)$th hop. Some vertex v has hopped to a channel and has set it's initial ineffective time τ to an exponential random variable with $\lambda = 1$. The τ's of all other nodes have been decremented by some amount. Since these are exponential random variables which are non-zero, the distribution of the residual τ's are still rvs drawn from the *same* exponential distribution. Furthermore, all these random variables are independent. Now the vertex v_t lies in channel i if its τ at $(t-1)$ divided by $\Gamma(\phi)$ is the smallest. As discussed above, for the contention domain, $\Gamma(\phi) = 1 - 1/X_i^t$, and so for an AP in channel i, the τ is an exponential rv with parameter $\lambda = 1 - 1/X_i^t$ (Property 1 of exponential rvs). The smallest τ divided by rate among all APs in this channel i is another exponential random variable with parameter $X_i^t \cdot \frac{X_i^t - 1}{X_i^t} = (X_i^t - 1)$ (Property 2 of exponential rvs). Therefore, the probability the minimum is from i, is precisely p_i^t (Property 3 of exponential rvs). □

Fix a channel i. Note that the random variable X_i^t evolves as a Markovian process whose transition probabilities are as follows.

$$X_i^{t+1} = \begin{cases} X_i^t - 1 & \text{w.p. } p_i^t \left(1 - \frac{1}{K}\right) \\ X_i^t + 1 & \text{w.p. } \frac{1}{K}\left(1 - p_i^t\right). \\ X_i^t & \text{otherwise.} \end{cases} \quad (3)$$

The first case is the probability that v_t lies in channel i and then hops to a channel which is not i. The second is the probability that v_t lies in some channel $j \neq i$ and then hops to channel i. Since it is Markovian, we have a stationary distribution, which we now calculate. Let $Y_t := X_i^t - 1$ and let $M := N - K$. We now claim that the stationary distribution of Y_t is $\mathsf{Bin}(M, 1/K)$. To see this, let π_i be $\Pr[Y_t = i]$ in the stationary distribution. We get the following recurrence using (3).

$$\pi_i = \pi_{i+1}\frac{i+1}{M}\left(1 - \frac{1}{K}\right) + \pi_{i-1}\frac{1}{K}\left(1 - \frac{i+1}{M}\right) + \pi_i\left(\frac{i}{MK} + \left(1 - \frac{i}{M}\right)\left(1 - \frac{1}{K}\right)\right)$$

A calculation shows that $\pi_i = \binom{M}{i}\left(\frac{1}{K}\right)^i\left(1 - \frac{1}{K}\right)^{M-i}$. That is, $Y_t \sim \mathsf{Bin}\left(M, \frac{1}{K}\right)$. Therefore, X_i^t converges to a Binomial distribution of mean $M/K + 1 = N/K$ and standard deviation $= \sqrt{M/K} \leq \sqrt{N/K}$.

3.4 Illustrative Examples

We now illustrate the functioning of IQ-Hopping through some simple examples to help provide the reader a feel for how IQ-Hopping works. We first show simulation results with 10 contending APs and 10 channels available (numbered 0 through 9). All APs start by selecting channel 0 and then run IQ-Hopping. In this simulation, packet losses only occur due to collisions, τ is chosen to be 1 sec and there is saturated downlink traffic. Figure 7 depicts the sequence of channel hops for each AP with time. The remarkable observation is that, at the end of 10 seconds, all the AP have self-organized and settled into a separate channel each!

When the number of available channels is less than number of APs. Figure 8 depicts the fraction of air-time obtained by each AP over a minute where there are only 3 available channels for 10 APs. As seen from Figure 8, all the APs keep hopping, however each AP obtains an average share of 0.3 (the Jain's Fairness Index was 0.99974). *This example demonstrates how IQ-Hopping converges quickly give each AP its fair share.*

Figure 9 depicts the average fraction of time that each AP (total 10) had access to a channel as the number of channels is increased from 1-10. Figure 9 also depicts the Jain's Fairness Index for all the APs. As seen from Figure 9, the average airtime for each AP for k channels is almost equal to $\frac{1}{k}$ and the Jain's fairness index is close to 1 for all k. Thus, IQ-Hopping provides a fair share to every AP, while providing full aggregate utilization.

IQ-Hopping vs Colouring algorithms on graphs In order to compare how well IQ-hopping finds channel assignments in general graphs, we generated 10 random graphs and 10 disc graphs with 100 nodes each with degrees 2, 5 and 10. On each graph we ran three popular centralized graph-colouring algorithms – *Greedy* [31], *Max Degree First* (MDF) [31] and *Recursive Largest First* (RLF) [19] to determine the number of colours. We then ran IQ-Hopping on each of these graphs with varying number of channels to find the minimum number of channel that results in an interference-free assignment. In Table 1 we depict the average number of minimum channels required by each of the schemes and the average maximum degree.

Deg	Disc				Random			
ree	Greedy	MDF	RLF	IQ	Greedy	MDF	RLSF	IQ
10	10.7	0.7	9.4	9.5	7.5	6.5	6	6
5	7.8	7.0	6.9	7.0	5.2	4.7	4	4.3
3	6.9	6.6	6.6	6.6	4.5	3.8	3.3	3.8

Table 1: Comparison of IQ-Hopping and Centralized Colouring Algorithms

As seen from Table 1, IQ-Hopping outperforms Greedy and MDF centralized colouring schemes and is very close in performance to RLSF which is considered as one of the best heuristics for colouring on both disc graphs as well as random graphs. This is a remarkable result since IQ-Hopping has no knowledge of the graph itself and operates in a completely distributed manner.

3.5 IQ-Hopping with Channel Bonding

In order to extend IQ-Hopping to work for channel bonding scenarios in 802.11n/ac, we make a simple modification to scale the

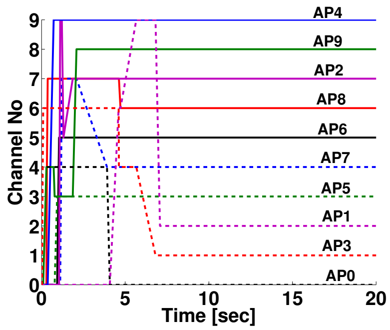

Figure 7: Self-Organization of IQ-Hopping in a single contention domain

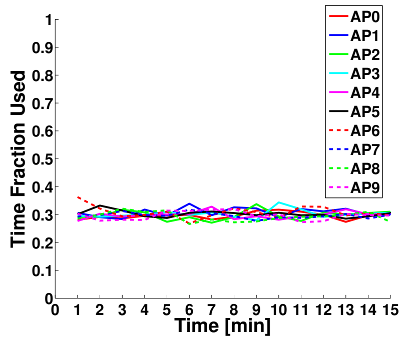

Figure 8: Convergence of IQ-Hopping

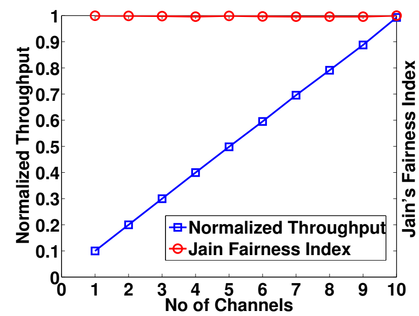

Figure 9: IQ-Hopping in a single contention domain with few channels

ineffective or wasted time according to the channel width. Suppose that $maxWidth$ is the width of widest possible channel that can be used (*e.g.*,80MHz) and further suppose that the currently used bandwidth is $CurrentWidth$. Line 10 in Algorithm 1 is modified as

$$t_{\texttt{ineff}} = t_{\texttt{ineff}} + \delta \times MaxWidth/CurrentWidth \qquad (4)$$

In 802.11ac, devices may use wide bonded channels only if there is no other device operating with a narrower channel overlapping with the wide channel. Thus, the set of channels $Channels$ in Line 2 must be the set of widest channels that an 802.11 device can possibly use, not only governed by the devices own limitations but also the limitations of co-existing devices.

4. IMPLEMENTATION

We have implemented IQ-Hopping on OpenWRT (Chaos Calmer 15.05) running on off-the-shelf routers (TP-LINK Archer C7 and Netgear WNDR3800) and MadWiFi0.9.4 running on Dell PCs. In this section, we briefly describe our OpenWRT implementation.

The IQ-Hopping implementation is a couple of hundred lines of code in mac80211, a module in the Linux kernel that performs all 802.11 frame management functions. The timer, as described in Algorithm 1, is expensive since it needs to fire at the granularity of packet transmission times (e.g., 100μs or even lower for ac) to get an accurate estimate of ineffective time. We avoid the timer in our OpenWRT implementation. Instead, we timestamp each packet just before it is enqueued to the hardware for transmission. When the hardware interrupts and provides the status of transmission (e.g., the PHY rate and if ACK was received), we obtain another timestamp. From these timestamps, we compute the time spent by a packet at the head of the hardware queue; subtracting the actual effective time for successful transmission (computed using packet size and PHY rate) from the queue head time gives us the ineffective time for each packet transmission.

Finally, when the function of effective and ineffective time exceeds the deadline, we notify the clients of an upcoming channel change. We include the 802.11h Channel Switch Announcement (CSA) Information Element (IE) to the beacon. The CSA has a channel number field which indicates the new channel and a count field which notifies when the switch will happen. For example, a count of five indicates that at the sixth upcoming beacon, the AP will switch to the new channel, with each subsequent beacon decrementing the count by one. This ensures that the switching process is robust to beacon losses. In the rare case that clients lose all five CSA beacons, the client will simply initiate re-association.

5. TESTBED RESULTS

In this section, we first present a testbed evaluation of our IQ-Hopping implementation on OpenWRT running on a commodity AP (TP-Link Archer C7). For clients, we use unmodified windows laptops with netgear A6200 WiFi cards using 802.11n with two spatial streams (MIMO). In the final subsection, we show experimental results using our MadWiFi implementation. Unless otherwise mentioned, experiments were run late in the night in a basement to ensure that outside interference was minimal.

5.1 IQ-Hopping Switching Impact on Real-Time Traffic

When an AP switches its channel, it first announces its intention of switching to all its clients and then the clients switch channels. This process could be result in an overhead in terms of transient packet losses. Since these are transient losses, they will most adversely effect high bandwidth real-time traffic such as video.

In this section we ask the question *How much will the rapid channel switching in IQ-Hopping effect real-time UDP and TCP traffic?* In order to evaluate this, we use one AP switching periodically to a randomly chosen channel in 2.4 GHz every 5 seconds. We use iperf to generate different types of traffic from a server connected through ethernet to the AP and further to a wireless Windows client. We then evaluate the impact of 100 such channel changes on various bit rates of UDP Constant Bit Rate (CBR) traffic, and TCP fixed rate traffic corresponding to streaming video.

Since we used the IEEE standards-compliant Channel Switching Announcement (CSA) mechanism, clients received channel switching message over multiple beacons (we tried counts of 3-5) and never lost synchronization in these experiments.

UDP CBR Traffic. The results for the UDP CBR experiments are shown in table 2 and also in Figure 10. The inter-packet interval was varied between 120μs at 80Mbps to $1ms$ at 10Mbps. The maximum data rate was chosen as 80Mbps after determining it was slightly higher than the maximum data rate the wireless channel could support without losses. We ensured that there was adequate queueing in software to avoid buffer overflow losses.

Figure 10 shows the second-by-seconds loss rates seen over a trace of 100 channel switches at different data rates – each spike depicts losses occurring in that second. The results in table 2 capture the average loss rates due to switching and otherwise. Between 10Mbps to 60MBps the only losses only occur during channel switching, however at 80Mbps losses also occur at other times. Table 2 indicates that *switching losses for UDP CBR traffic are quite minimal, with a loss rate of at most 1.3%.*

The reason for the losses during channel switch occurs most

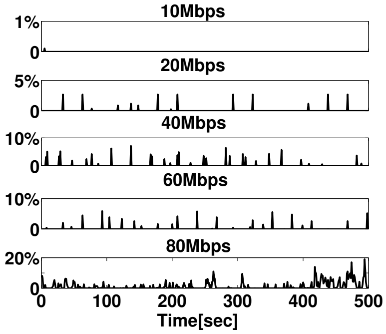

Figure 10: IQ-Hopping channel switching loss rate for UDP traffic

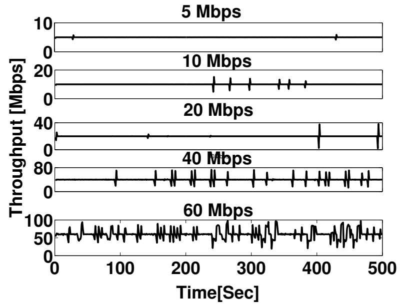

Figure 11: IQ-Hopping channel switching effects on TCP traffic

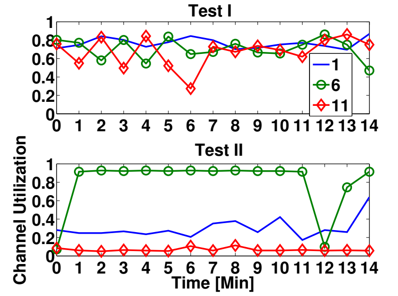

Figure 12: Background utilization for Trace I and II

Data Rate Mbps	5	10	20	40	60	80
Switching Loss(%)	0	0.001	0.26	1.03	0.7	1.34
Other Loss(%)	0	0	0	0	0	1.3

Table 2: Overhead of IQ-Hopping: UDP CBR, 802.11n

Data Rate Mbps	5	10	20	40	60
Mean Throughput Dip(%)	10	25	45	40	55
Mean Dip Duration(sec)	1	1	1	1	2.5
Fraction of Dip(%)	2	6	8	20	48

Table 3: Overhead of IQ-Hopping: Effect on TCP CBR (HTTP video streaming), 802.11n

likely when the AP's radio switches channel before the client's radio and starts to transmit packets. While MAC level retransmissions take care of most of these losses, at higher rates some losses never recover. We estimate that the radios take $1 - 60ms$ for the channel switches and is the cause of these packet losses.

TCP CBR Traffic (Video over TCP). The results for the TCP CBR experiments are shown in table 3 and also in Figure 11. The TCP traffic was generated with various CBR values to mimic HTTP streaming video transmission at different data rates. Figure 11 depicts the second-by-second throughput at the client. As seen in Figure 11 that out of the 100 channel switches, many have no impact on TCP throughput. However, when there is an impact on TCP, unlike UDP, the impact of losses is amplified due to TCP's congestion control. Even so, TCP throughput suffers for at most a second or two (when a dip occurs, on average the dip in throughput for 1-2s is about 20% at 40Mbps and 48% at 60Mbps) and then immediately bounces back to above the CBR rate to ensure that client buffers are refilled and there is no over all loss in average rate.

5.2 Evaluation using real congestion traces

In order to demonstrate how IQ-Hopping can perform better than a congestion measurement based scheme in real world scenarios, in this experiment, we emulate congestion using two traces from our measurement data. We use the traces to generate background traffic at the appropriate utilization each minute in each of the three channels in 2.4 GHz simultaneously using three APs associated to clients. We ensured that the background traffic was high priority (DSCP code 46, expedited forwarding) so that it suffered no losses and was able to occupy anywhere from 0.0 to 0.9 of airtime utilization to accurately mimic the utilization levels as seen in the traces. The measurement based scheme chooses the least congested channel once every 15 minutes.

Figure 12 shows the utilization for a 15 minute interval for the two traces. Trace I is a highly congested environment where instantaneous congestion is high and varying in all channels. In Trace II, channel 6 is highly congested while channels 1 and 11 are relatively free. However, congestion is lowest in channel 6 at the time of measurement in both traces. This leads to the measurement-based scheme incorrectly choosing channel 6 in trace II, (the best channel is 11 in Trace II) and staying there throughout the interval.

Figure 13 depicts the TCP throughput time series for IQ-Hopping and measurement-based scheme for Trace I. IQ-Hopping is able to use variations in congestion levels in the channels to achieve a higher throughput than the least congested scheme. Figure 14 shows the TCP throughputs obtained by each of the schemes in the two traces. We see that IQ-Hopping provides throughput gains of about 75% in Trace I and over 2X in Trace II.

5.3 IQ-Hopping and Microwave Interference

It is well-known that microwaves may interfere with WiFi communication in the 2.4 GHz band. Figure 15 shows the spectrum analyzer output when the microwave was on. It shows that while channels 1 and 6 are affected a little, channel 11 is significantly affected (higher energy in the figure). However, any static channel assignment scheme cannot recover from such an effect, especially since microwaves are typically on for only a few minutes and the interference pattern varies across models.

In this experiment, we place an IQ-Hopping AP about 10 feet from the microwave and a client about 30 feet from the microwave and perform a TCP download. The TCP throughput variation over time and the various channel sojourn durations are shown in Figure 16. With the microwave off, the average throughput on channels 1, 6, and 11 are 55.1, 46.3, and 52.6 Mbps, respectively while with the microwave on, the same drop to 51, 39, and 25 Mbps, respectively. Thus, channel 11 seems to suffer the most which is as expected. While the IQ-Hopping AP hops across all channels 1, 6, and 11, note that the durations it spends in channel 11 are the lowest as ineffective time in this channel accumulates the fastest. The average throughput achieved by IQ-Hopping is 41.2 Mbps when the microwave is on, which is higher than average throughputs across channels 1, 6 and 11 with the microwave on and also higher than both channels 6 and 11 individually. This demonstrates the benefit of IQ-Hopping in the presence of non-WiFi sources of interference.

5.4 Evaluation on MadWiFi

In this section, we use varying background traffic to demonstrate the robust adaptability of IQ-Hopping on a MadWiFi testbed.

MadWiFi Experiment 1. In this experiment we used six nodes. Four of these were used to create two AP-Client pairs running stan-

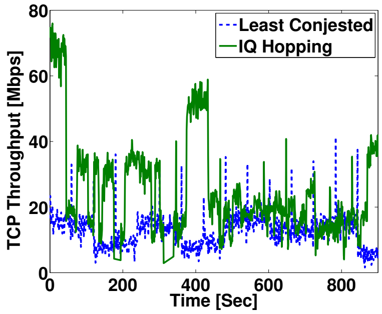

Figure 13: Throughput observed during Trace I

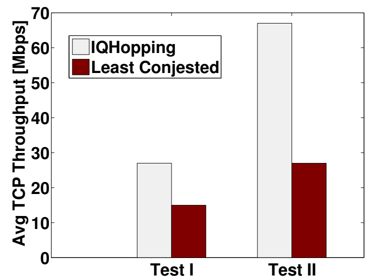

Figure 14: Avg TCP Throughput for of Trace I and II

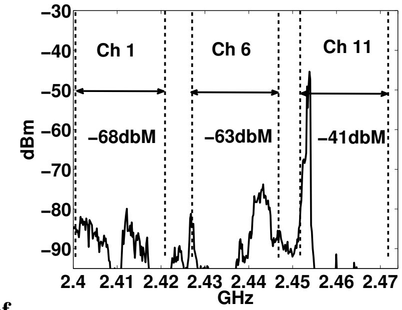

Figure 15: Spectrum analyzer output of microwave interference

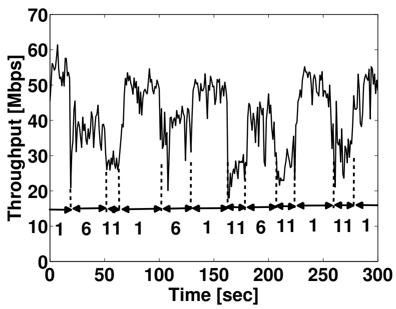

Figure 16: IQ-Hopping in the presence of microwave interference

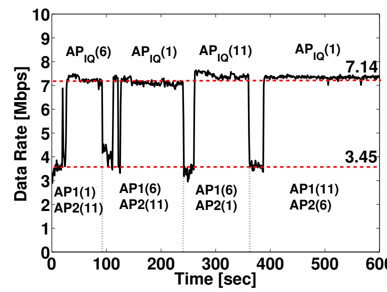

Figure 17: How IQ-Hopping helps APs avoid saturated channels.

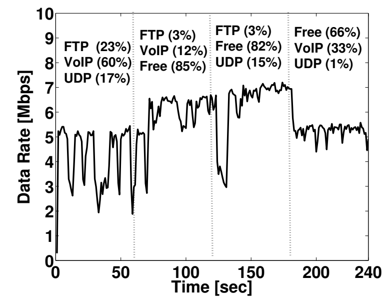

Figure 18: How IQ-Hopping functions across real traffic

dard WiFi – AP1-C1 and AP2-C2. While the other two formed an AP-Client pair, $AP_{IQ} - C_{IQ}$ that ran IQ-Hopping. AP1-C1 and AP2-C2 ran saturated UDP traffic. Through the course of the experiment AP1 and AP2 changed their channels. We first ran iperf on regular WiFi in a free channel and an occupied channel to find the baseline achieved throughputs to be 7.14 and 3.45 Mbps respectively (indicated as dotted horizontal lines in Figure 17). The channels they were on are indicated in Figure 17, for example, in the interval 90-240 sec, AP1(6)AP2(11) indicates that AP1 and AP2 were on channels 6 and 11 respectively. As seen from Figure 17 AP_{IQ} correctly moved to the empty channel after each channel change, usually within about 20s. The average throughput of AP_{IQ} was 6.6Mbps – a loss of about 7% from the maximum possible.

MadWiFi Experiment 2. In this experiment we used eight nodes with three AP-Client pairs serving as background traffic and the last pair running IQ-Hopping. On channels 1, 6 and 11 we had FTP traffic, VoIP traffic and saturated UDP traffic respectively. Initially all three traffic sources were on but once every 60s one of them was switched off as depicted in Figure 18. In Figure 18 we also provide the fraction of time spent in each of the channels. In 0-60 seconds, IQ-Hopping spent most time (60%) in VoIP channel and the least in the UDP channel (17%) – this is as desired since VoIP is the least congested background traffic while UDP is the most. When UDP was turned off between 60-120s, IQ-Hopping spent 85% on the free channel and only 3% on the FTP channel. Similarly, in other sections as well the fractions of time IQ-Hopping spent in the channels is in decreasing order of the amount of congestion they cause, demonstrating IQ-Hopping's robust adaptability to background traffic.

6. RELATED WORK

Significant prior work has been done in the general context of channel assignment. An excellent survey on some of the proposed approaches is in [12].

Centralized Approaches. Traditional channel assignment mechanism have relied on careful AP placement [15] and vertex coloring [17], where the objective is to minimize interference during channel assignment. Various other approaches consider jointly optimizing the channel assignment and AP placements [11, 30]. However, these approaches do not consider the actual channel usage during assignment process. PIE proposes online interference estimation [28] which uses collective information from the various nodes to deduce the interfering patterns. There are other dynamic assignment schemes that either try to mitigate interference [24], optimize channel assignment based on global information [26], or based on end-to-end QoS [8] where the objective is to maximize the demand acceptance rate. Further, other optimization approaches that attempt to jointly optimize both channel assignment and routing have also been proposed in the literature[5]. However, the centralized nature of these approaches render them unsuitable for uncoordinated deployments.

Distributed Approaches. [18] proposes an approach based on a requirement of a common channel between all nodes to ensure reliable connectivity. This requirement is lifted in [25], where a node selects a channel which is least used by its neighbors. ROMA [14] further builds on this approach by considering both link losses and loss fluctuations. Some approaches [10] rely on local load measurements at the node and periodically update the channels based on this information. Significant number of schemes rely on decoding beacons to determine the number of competing devices and then choose the channel which is the least congested [4, 21]. A key limitation of these approaches is that often the available share in a channel is hard to estimate, since it depends not only on the number and traffic from neighboring devices but also on the interference relationships between them and their own neighbors.

Channel Hopping Approaches. Channel hopping schemes [6, 13, 20] have been proposed in the literature, where APs hop between different channels based on a hopping sequence. This helps them harness frequency diversity of the channels and avoids getting stuck in low throughput configurations. For example, in MaxChop [20], the pre-decided hopping sequence is used to ensure that the clients and the APs hop in a controlled manner. [13] employs a common rendezvous channel for coordination between different nodes. Distinction between the various hopping approaches can also be made based on whether they are initiated by the sender [32] or receiver [29]. However, construction of the hopping sequence requires accurate knowledge of interfering APs and their traffic dynamics to be effective.

Flexible Channel Widths Approaches. With newer 802.11 standards and White spaces supporting channel bonding, recent work [7, 27, 22] support flexible channel widths, which is lacking in the earlier schemes. In mCham [7], a device uses a dedicated secondary radio to measure the interferers and the utilization on each channel, and uses this information for channel selection. SampleWidth [22] on the other hand, adjusts both the center frequency and the channel width based on the subjected traffic load. Fluid [27] on the other hand, requires a central controller to perform per-packet channelization decisions like choice of central frequency and channel width. However, given the centralized approach and fine computational granularity, it can result in significant overheads.

7. CONCLUSION

Through a large scale measurement study we show that there can be significant congestion in WiFi and channel selection can provide gains during congestion times. In order for a channel selection scheme to obtain gains, it must switch to a low congested channel very dynamically – in the order of few minutes. We propose, IQ-Hopping, a channel selection scheme that causes APs to move away from congestion bottlenecks and spread out across uncongested channels in a matter of seconds. Further IQ-Hopping does not require any channel measurements. We show the effectiveness of IQ-Hopping through formal theoretic guarantees, implementation on off-the-shelf routers and through simulations.

8. REFERENCES

[1] iPass Wi-Fi Growth Map Shows 1 Public Hotspot for Every 20 people on Earth by 2018. http://www.ipass.com/press-releases/ipass-wi-fi-growth-map-shows-one-public-hotspot-for-every-20-people-on-earth-by-2018/.

[2] Ruckus brings big WiFi to small businesses. http://www.prnewswire.com/news-releases/ruckus-brings-big-wi-fi-to-small-business-with-xclaim-280745642.html.

[3] Wi-Fi Wireless Trends 2015: The New Network. http://www.pipelinepub.com/2014_Telecom_Trends/WiFi_wireless/2.

[4] M. Achanta. Method and apparatus for least congested channel scan for wireless access points, 2006. US Patent No. 20060072602.

[5] M. Alicherry, R. Bhatia, and L. E. Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *ACM MobiCom*, pages 58–72. ACM, 2005.

[6] P. Bahl, R. Chandra, and J. Dunagan. Ssch: Slotted seeded channelhopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *ACM MobiCom*, 2004.

[7] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh. White Space Networking with Wi-Fi like Connectivity. In *ACM SIGCOMM*, August 2009.

[8] B. Bakhshi, S. Khorsandi, and A. Capone. On-line joint qos routing and channel assignment in multi-channel multi-radio wireless mesh networks. *Computer Communications*, 34(11):1342–1360, 2011.

[9] L. Barenboim and M. Elkin. Distributed $o(\delta + 1)$-coloring in linear (in δ) time. In *ACM STOC*, 2009.

[10] V. Bhandari and N. H. Vaidya. Channel and interface management in a heterogeneous multi-channel multi-radio wireless network. Technical report, DTIC Document, 2009.

[11] G. Bianchi. Performance analysis of the IEEE 802.11 distributed corrdination function. *IEEE Journal on Selected Areas in Communications*, March 2000.

[12] S. Chieochan, E. Hossain, and J. Diamond. Channel Assignment Schemes for Infrastructure-Based 802.11 WLANs: A Survey. *IEEE Communications Surveys and Tutorials*, 12(1):124–136, 2010.

[13] C. Cordeiro and K. Challapali. C-mac: A cognitive mac protocol for multi-channel wireless networks. In *DySPAN*. Ieee, 2007.

[14] A. Dhananjay, H. Zhang, J. Li, and L. Subramanian. Practical, distributed channel assignment and routing in dual-radio mesh networks. In *ACM CCR*, volume 39, pages 99–110, 2009.

[15] M. Gast. *802.11 Wireless Networks: The Definitive Guide*. OReilly, 2005.

[16] M. M. Halldórsson. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, pages 19 – 23, 1993.

[17] A. Hills. Large-scale wireless lan design. *IEEE Communication Magazine*, Nov 2001.

[18] B.-J. Ko, V. Misra, J. Padhye, and D. Rubenstein. Distributed channel assignment in multi-radio 802.11 mesh networks. In *IEEE WCNC*, 2007.

[19] F. Leighton. A graph coloring algorithm for large scheduling problems. 84(6):489–506, 1979.

[20] A. Mishra, D. Agrawal, V. Shrivastava, S. Banerjee, and S. Ganguly. Distributed Channel Management in Uncoordinated Wireless Environments. In *ACM MobiCom*, 2006.

[21] A. Mishra, S. Banerjee, and W. Arbaugh. Weighted coloring based channel assignment for WLans. *SIGMOBILE Mobile Computing Communication Review*, 2005.

[22] T. Moscibroda, R. Chandra, Y. Wu, S. Sengupta, P. Bahl, and Y. Yuan. Load-aware spectrum distribution in wireless lans. In *Network Protocols, 2008. ICNP 2008. IEEE International Conference on*, pages 137–146. IEEE, 2008.

[23] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge, 1993.

[24] K. N. Ramachandran, E. M. Belding-Royer, K. C. Almeroth, and M. M. Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *Infocom*, 2006.

[25] A. Raniwala and T.-c. Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *IEEE INFOCOM*, volume 3, pages 2223–2234, 2005.

[26] A. Raniwala, K. Gopalan, and T.-c. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(2):50–65, 2004.

[27] S. Rayanchu, V. Shrivastava, S. Baneree, and R. Chandra. FLUID: Improving Throughputs in Entreprise Wireless LANs through Flexible Channelization. In *ACM MobiCom*, 2011.

[28] V. Shrivastava, S. Rayanchu, S. Banerjee, and D. Papagiannaki. PIE in the Sky: Online Passive Interference Estimation for Enterprise WLANs. In *NSDI*, 2011.

[29] A. Tzamaloukas and J. Garcia-Luna-Aceves. A receiver-initiated collision-avoidance protocol for multi-channel networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 189–198. IEEE, 2001.

[30] P. Wertz, M. Sauter, F. Landstorfer, G. Wolfle, and R. Hoppe. Automatic optimization algorithms for the planning of wireless local area networks. In *IEEE Vehicular Technology Conference*, 2004.

[31] R. A. Wilson. *"Graph, Colourings, and Four-Colour Theorem"*. Oxford Science Publications.

[32] Z. Yang and J. Garcia-Luna-Aceves. Hop-reservation multiple access (hrma) for ad-hoc networks. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 194–201. IEEE, 1999.