WLAN Channel Selection Without Communication

D.J. Leith, P. Clifford, V. Badarla and D. Malone[1]

Hamilton Institute, National University of Ireland, Maynooth, Ireland.

Abstract

We propose a new class of channel allocation algorithms that are simple, robust and require no communication between interfering WLANs. We show they are provably correct and yet remarkably efficient under a wide range of network conditions and topologies. The algorithms are suited to implementation on standard equipment, requiring no special hardware support and making only light demands on computational resources. We demonstrate this by implementing algorithms on an experimental testbed using commodity hardware. We present detailed measurements of performance in a real office environment. This environment includes complex spatially-varying noise, channel-dependent interference between WLANs, time-varying channel quality and external interference sources. Despite these challenges the algorithm performs well.

Keywords: 802.11; WLAN; Channel allocation

1. Introduction

In this paper we consider how a group of access-points[2] can self-configure their channel choice so as to minimise interference between one another and thereby maximise network capacity. A practical, reliable and resilient solution to this channel selection task is essential for future wireless networks. Industry roadmaps point towards (i) increasingly dense network deployments (including multi-hop operation) and (ii) multi-channel/multi-radio devices. The associated increase in demand on the available spectrum is already creating a real need for a good channel selection methodology.

The common practice for arriving at a channel allocations historically lies somewhere between (i) a detailed radio survey and careful placement of APs combined with manual spectrum planning, and (ii) placement of APs according to current need (leading to organic growth of the wireless network) and use of device default channels. More recently, to reduce the manual administrative burden in larger deployments there has been a move towards centralised solutions, where channel selection (among other things) is delegated to a central management system that has a control plane connection to every AP. However, such schemes assume common administrative control of the APs.

Common control of the APs is impractical in some practical situations; interfering APs may belong to different administrative domains (e.g. interfering wireless networks may be operated by different households or businesses). For this reason, there is literature proposing distributed schemes ([2, 3, 4, 5, 6, 8, 14, 15, 16, 17, 18]) which require only local communication between APs that directly interfere with one another, which can be implemented via explicit message-passing or packet sniffing. However, even message-passing between neighbours may become difficult when interfering APs lie in different administrative domains, and packet sniffing on the radio channel can run into the difficulty as the distance over which packets are readable is

Email address: David.Malone@nuim.ie (D.J. Leith, P. Clifford, V. Badarla and D. Malone)

[1]This work was supported by Science Foundation Ireland grant IN3/03/I346 and 07/SK/I1216a. The authors thank Dave Reid for work on the experimental measurements.

[2]We use the term access point (AP) to denote the co-ordinating station in a WLAN that is responsible for channel selection. There is no intention to restrict consideration to a specific WLAN technology. Each AP has associated client stations and we refer to the collection of clients plus AP as a WLAN.

typically much less than the distance over which network transmissions interfere (thus interfering APs may well not be able to sniff each others packets). Indeed, interference may arise from devices other than APs (e.g. dumb sensors, microwave ovens), which cannot engage in message passing.

In this paper we introduce a class of channel allocation algorithms that aim to operate under such constraints. The algorithms are simple and suited to implementation on standard equipment, requiring no special hardware support. Crucially, they require no communication between interfering WLANs but are still provably correct. We will show that the algorithms are efficient under a range of conditions. To demonstrate the algorithms effectiveness in a real environment we implement the algorithm on an experimental testbed using commodity hardware. We present detailed measurements of performance in a real office environment with which exhibits many complex impairments including spatially-varying noise, channel-dependent interference between WLANs, time-varying channel quality, external interference sources.

The paper is organised as follows. In Section 2 we review some related work in this area. Next, in Section 3 set out the requirements for our algorithm and how we arrived at these requirements. One of the factors in our design is the nature of the interference environment, and in Section 4 we present experimental measurements to illustrate the interference environment. Having introduced our aims and the environment, we describe our communication-free algorithm in Section 5 and assess its basic performance. This algorithm can successfully allocate channels, providing there are sufficient channels to allow a non-interfering channel assignment.

Of course, in practice the number of channels for WiFi is fixed, and it is impossible to know in advance if enough channels are available. In Section 6 we show how the algorithm can deal with the situation where the channel allocation task is infeasible. Another important practical factor is that in practice networks are turned on and off. To understand how our algorithm performs in this situation, Section 7 consider its behaviour when the interference structure is time-varying. Finally, in Section 8 we present the results of implementing the algorithm in an 802.11 experimental testbed.

2. Related Work

The channel allocation task has been the subject of a considerable literature, spanning cellular networks (e.g. [14]), wireless LANs (e.g. [2, 3, 4, 5, 6, 8, 14, 15, 16, 17, 18] and references therein) and graph theory. Almost all previous work has been concerned either with centralised schemes or with distributed schemes that employ extensive message-passing. For example, centralised algorithms for channel allocation have been proposed by Leung et al. [8] and Mishra et al. propose a distributed channel hopping scheme [5]. Centralised and message-passing schemes have many inherent advantages, explaining the large amount of literature on these topics. In certain situations, however, these systems may not be applicable. For example, differing administrative domains may be present as mentioned above.

A notable exception is the work of Kauffmann et al., which proposes a distributed simulated annealing algorithm for joint channel selection and association control in 802.11 WLANs [12, 13]. However, heuristics are used to both terminate the algorithm and to restart it if the network topology changes. Network-wide stopping/restarting in a distributed context can be challenging without some form of message-passing. In addition to requiring communication, the majority of previous work assumes that interference is channel independent. Due to the possibility of channel dependent interference, and other complex features of the environment, experimental validation in a real environment is essential. To this end we include both experimental validation and a detailed study of the environment. [20, 21] present early versions of our work confined to situations where a non-interfering allocation exists; these do not include experimental measurements.

3. Aims and Motivation

As we noted in the introduction, there are situations where channel allocation is necessary, but there is no centralised authority is managing the network. Further, interference may arise from non-network devices or time varying. We will see in Section 4 that interference between networks may even be channel dependent.

By considering the challenges that arise when doing channel allocation in such an unconstrained environment and then attempting a practical implementation, we identified the following key issues that a practically useful channel allocation algorithm should address:

Time-varying interference: The pattern of interference is typically time-varying in nature. Time variations can arise, for example, from new stations connecting to an AP or existing stations becoming inactive. External interference sources are also commonly time-varying. The requirement is for a channel allocation algorithm that supports automatic adaptation to changes in the interference topology. This has implications for algorithm design as stopping/restarting in response to changes may be problematic in a distributed implementation.

Channel dependent nature of interference: Existing work often assumes that the interference between WLANs is similar on every radio channel. A conflict graph is constructed by associating a graph vertex with each WLAN and inserting an edge between WLANs that interfere. The use of a single graph is equivalent to assuming that interference is channel independent. However, in reality frequency dependent differences in path propagation properties mean that the interference between WLANs is generally channel dependent — in this paper we demonstrate via experimental measurements that this effect is significant in practice. This means that a *different* conflict graph is associated with each channel. This potentially has profound implications as the behaviour of many proposed colouring-based algorithms for channel allocation is unclear in this context. Note that this is a different issue to the use of overlapping channels to increase spatial reuse, e.g. [7].

Fully decentralised operation with no message-passing: Centralised solutions have significant advantages but require common administrative control of every interfering AP. Numerous distributed schemes have been proposed e.g. [2, 3, 4, 5, 6, 8, 14, 15, 16, 17, 18], but require message-passing between APs that interfere with one another. The message-passing paradigm allows for many possible benefits. Direct message-passing is however challenging when interfering APs lie in different administrative domains. Indirect message-passing, by sniffing packet headers on the radio channel, runs into the difficulty that the distance over which packets are readable is typically much less than the distance over which transmissions interfere. Thus interfering APs may well not be able to decode each others packets. The requirement is therefore for decentralised channel allocation i.e. channel allocation that does not depend upon message-passing or packet sniffing. We note that the principle of decentralised resource allocation is well-known in wireless MAC design, and is for example embodied in the 802.11 CSMA/CA MAC. However, the application of this principle to the channel selection task is new.

Implementation on standard hardware: We seek algorithms than can be directly implemented on standard hardware, without changes to the MAC or PHY. This implies, for example, that an algorithm cannot require time synchronisation (e.g. use of slotted time) across interfering WLANs.

Performance guarantees: Most existing channel allocation schemes are heuristic and come with few performance guarantees. In particular, there is often no guarantee that the scheme will find a good channel allocation even if one exists.

We argue that these issues will be significant in many situations, and while they do not cover all issues relating to WiFi interference management (e.g. rate control), the do represent significant structural constraints that influence channel allocation.

4. Nature of Interference Environment

We begin by presenting experimental measurements showing the interference behaviour in the real office environment shown in Fig. 1. This is of interest in its own right as interference characteristics of operational networks remain relatively poorly characterised and measurements provide insight into the performance requirements that must be met by any channel allocation algorithm if it is to be practically applicable.

4.1. Testbed Setup

Our testbed consists of five WLANs (denoted WLAN A – WLAN E) located as shown in Fig. 1. The WLANs are constructed using embedded Linux boxes based on the Soekris net4801, with 5 boxes configured

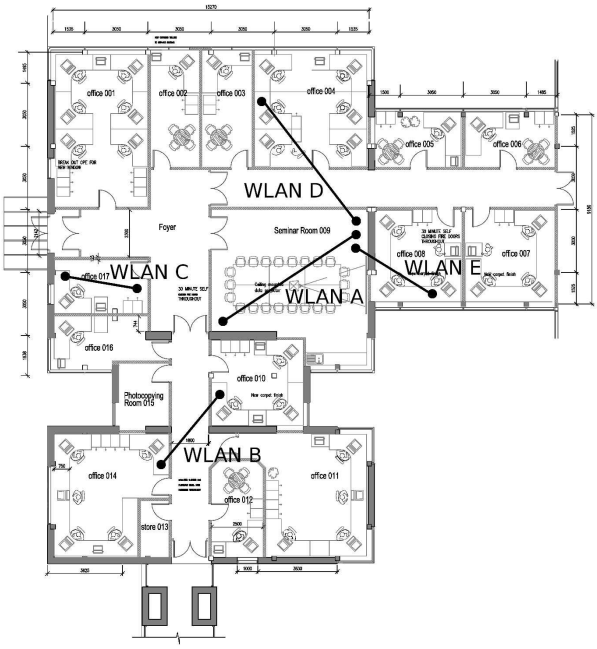


Figure 1: Plan showing wireless node locations (approx. 25m by 22m).

Hardware	model	spec
5× AP	Soekris net4801	266Mhz 586
5× client	Soekris net4801	266Mhz 586
5× measurement node	Dell 3100C	2.8Ghz P4
WLAN NIC	Atheros AR5004G	802.11a/b/g Mini PCI

Table 1: Testbed Summary

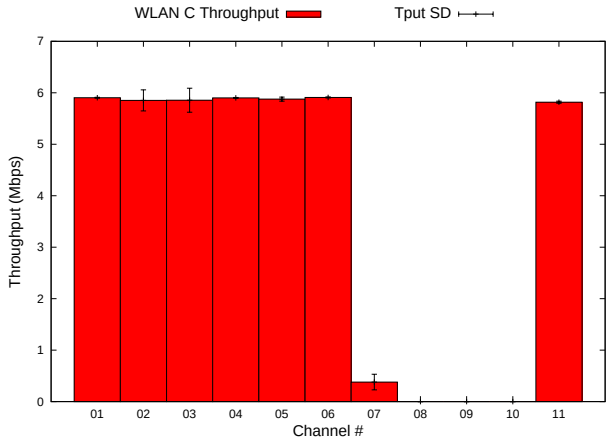


Figure 2: Baseline throughput for WLAN C versus channel number in 2.4GHz band (no other WLANs active, 11Mbps PHY rate).

as APs in infrastructure mode and 5 as client stations. We also use 5 PCs acting as monitoring stations to collect measurements — this is to ensure that there is ample disk space, RAM and CPU resources available so that collection of statistics does not impact on the transmission of packets. All systems are equipped with an Atheros 802.11a/b/g mini-PCI card with a single external antenna. The system hardware configuration is summarised in Table 1. All nodes use a Linux 2.6.16.20 kernel and the MADWiFi wireless driver. Specific vendor features on the wireless card, such as turbo mode, are disabled. Unless otherwise stated, all of the tests are performed using the 802.11a physical transmission rate of 18 Mbps with RTS/CTS enabled and the channel number explicitly set. The histograms shown show mean throughput over approximately 250s using UDP.

4.2. External Interference Sources

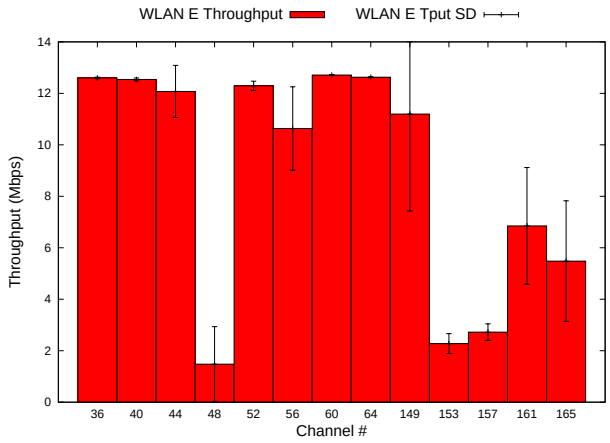
The testbed hardware supports operation both in the 802.11a 5GHz band and in the 802.11b 2.4GHz band. While spectrum analyser measurements revealed little external interference in the 5GHz band (a noise floor of around -80dB being typical), significant external interference was observed in the 2.4GHz band. For example, Fig. 2 shows measured throughput versus channel number in the 802.11b band for WLAN C — none of the other WLANs are active here, so there is no testbed related interference. It can be seen that there exists significant background noise on channels 7–10. Measurements using a spectrum analyser confirmed the presence of noise on these channels, which was traced to bluetooth devices operating in a lab (marked office 001 in Fig. 1) close to WLAN C. We note that the level of external interference is strongly location dependent and is essentially negligible for WLANs B and E which are located approximately 10m further than WLAN C from the interference source.

4.3. Time-varying Channel Quality

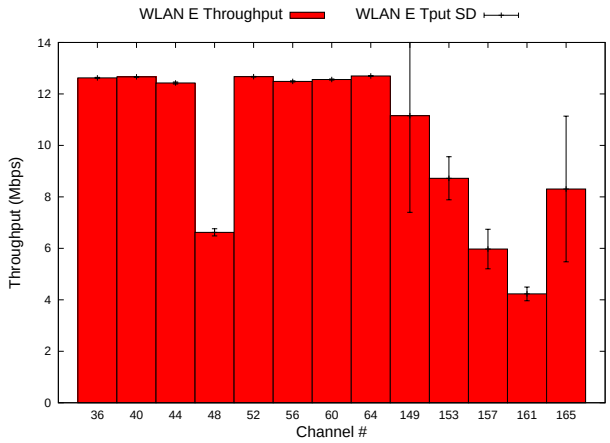
Our measurements indicate that the channel quality can be strongly time-varying. For example, Fig. 3 shows measurements of the mean rate of successful transmissions versus channel number when a single WLAN is active (WLAN E in this case). Measurements are repeated about an hour apart, with identical offered load and network configuration. The time-varying nature of the channel quality is evident — e.g. compare channels 48 and 153.

Also marked on Fig. 3 are error bars that indicate the standard deviation of the error time history measured over a period of 50s from which it is evident that variations in channel quality also occur on shorter time-scales. This is shown in more detail in Fig. 4 which shows an example time history of measured channel quality over a period of approximately 60 minutes[3]. Note that the frame error rate rises to around 15% for about 10 minutes early in this experiment, then falls to around 3% after 30 minutes.

[3]The frame error rate is measured by sending packets with RTS/CTS and counting the fraction of exchanges where RTS/CTS is successful, but no ACK is received for the data.



(a)



(b)

Figure 3: Throughput with a single WLAN active (i.e. no interfering WLANs). Measurements for WLAN E over all 802.11a channels. The measurements in the upper and lower plots are taken about 1 hour apart. Observe variation in throughput with both channel and time.

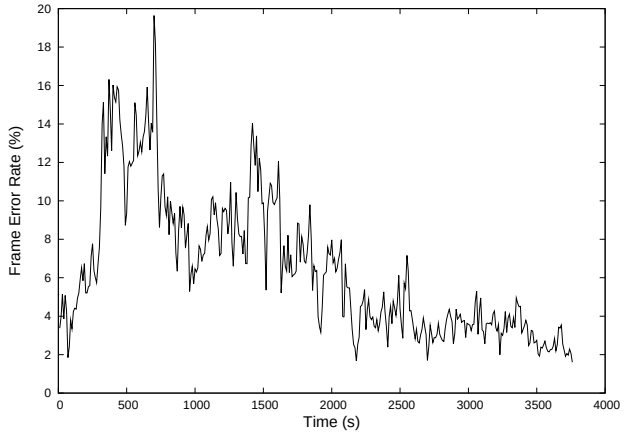


Figure 4: Example of time-varying channel quality.

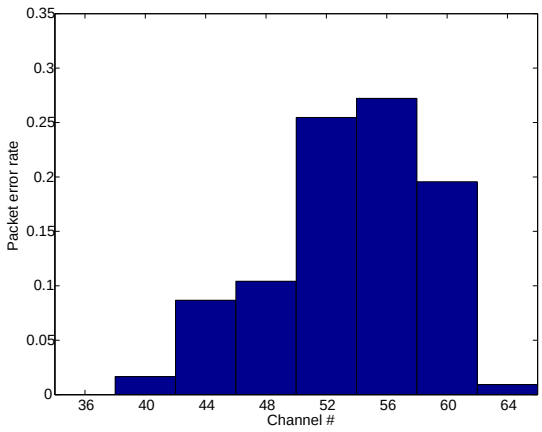


Figure 5: Interference induced error rate versus channel number in 5GHz band. Here WLANs B and C both transmit CBR traffic on the same channel. The plot shows measured packet error rate at WLAN C as the common channel is varied.

In this particular example, measurements using a spectrum analyser indicate that there is little background noise (the noise floor is consistently below -80dB) and thus presumably the measured variations in channel quality are related to radio propagation effects. Radio signal propagation within a building is of course complex and our tests indicate that it can vary as, for example, doors are opened/closed, people move about, etc.

Fortunately, we can also observe in Fig. 3 that certain channels are consistently of good quality, e.g. channels 36–44 and 60–64 and so a channel allocation algorithm should seek to allocate these channels.

4.4. Channel Dependent Interference

Our measurements indicate that the level of interference between WLANs can be strongly channel dependent. For example, Fig. 5 shows the measured interference level between WLANs B and C as the channel number is varied. We found this effect to be particularly pronounced in the 5GHz band, with a significantly lower level of channel dependence measured in the 802.11b 2.4GHz band. This is unsurprising, as we expect path propagation characteristics to be frequency dependent. However, it has profound implications for channel allocation algorithms as it means that the channel allocation task is not equivalent to standard graph colouring, but rather to a multi-graph colouring task.

5. Communication-Free Algorithm

In this section we introduce the class of decentralised algorithms studied in this paper. We begin by considering situations where a non-interfering channel allocation is feasible i.e. where the number of available channels is greater than or equal to the chromatic number of the interference multi-graph. We then extend consideration in Section 6 to the case where a non-interfering channel allocation is infeasible.

In Section 3 we outlined the issues that we want address with our algorithm design. Since communication between interfering APs may be impossible, we design our scheme so that each AP depends on a local measurement of the presence of interference. We have seen that interference may be time varying, so we periodically reassess our channel choice. Also, performance on one channel may not be indicative of performance of other channels, so we must separately learn about the conditions on each channel. One way to achieve this is to work with a probability of choosing each channel, and update it after each channel access. Broadly speaking, if channel access is successful, we would like to continue using the channel and if channel access is unsuccessful we would like to reduce the probability of reusing the channel.

5.1. Algorithm

Let c denote the number of available channels and let each access point with responsibility for channel selection maintain a c element state vector p. Let p_i denote the ith element of p with $\sum_i^c p_i = 1$. Consider

the following class of communication-free decentralised algorithms for updating p.

Communication-Free Learning (CFL) Algorithm

1. Initialise $p \leftarrow [1/c, 1/c, \ldots, 1/c]$
2. Toss a weighted coin to select channel i, with probability p_i. Sense the channel quality. Any interference measure can be used that yields a "success" when interference/channel noise is within acceptable levels and "failure" otherwise[4].
3. On success on channel i, update p as

$$p_i \leftarrow 1, p_j \leftarrow 0 \quad \forall j \neq i \tag{1}$$

   i.e. on a successful choice we use the same channel for the next round. This creates a degree of "stickiness" which ensures that any channel allocation that removes interference between all WLANs is an absorbing state (a state is absorbing when the algorithm cannot leave that state once it enters it).
4. On failure on channel i, update p as

$$p_i \leftarrow (1-b)p_i, \tag{2}$$

$$p_j \leftarrow (1-b)p_j + \frac{b}{c-1} \quad \forall j \neq i \tag{3}$$

   i.e. on a failure multiplicatively decrease the probability of using that channel, redistributing the probability evenly across the other channels. b is a design parameter, $0 < b < 1$; the selection of the value of b is considered in detail below.
5. Return to 2.

5.2. Convergence to non-interfering allocation

This CFL algorithm is evidently straightforward, but two immediate questions that arise are whether it will indeed always converge to a channel allocation and whether or not this allocation is non-interfering. Our main analytic result is to answer both of these questions in the affirmative, providing it is possible with the available channels. The situation where insufficient channels are available is addressed in Section 6.

Let $G(i) = (V, E(i))$ denote the interference graph associated with use of channel i in a wireless network. That is, the vertices V of $G(i)$ are the WLANs and the edge set $E(i)$ contains an edge between vertices (u, v) when WLAN u and v interfere on channel i. The interference environment is then characterised by the family of graphs $\{G(i), i \in [1, 2, .., c]\}$. A non-interfering channel allocation is one where each WLAN uses a channel i that is different from all of its neighbours in $G(i)$. Note that in the special case where $G(i) = G \forall i$ then the interference graph is the same on every channel and we recover a standard single graph colouring problem.

Theorem 1 *Suppose each vertex in V operates the CFL algorithm. Assume that the channel allocation problem is feasible (i.e. a non-interfering channel allocation does indeed exist). Then the CFL algorithm converges, with probability one, to a non-interfering channel allocation.*

We omit the proof of this result [10] and a more detailed analysis [11]. The proof actually provides a partial answer to a further question, namely how quickly the algorithm converges to a non-interfering allocation. The stopping time is the time taken for the algorithm to converge.

Corollary 1 *Let τ denote the stopping time of the CFL algorithm. Then prob$[\tau > k] < \alpha e^{-\gamma k}$, for positive α, γ.*

[4] We might, for example, use an aggregate measure derived from multiple packet transmissions or from direct measurement of the channel SINR. See Section 8 for details of the measure used in our testbed.

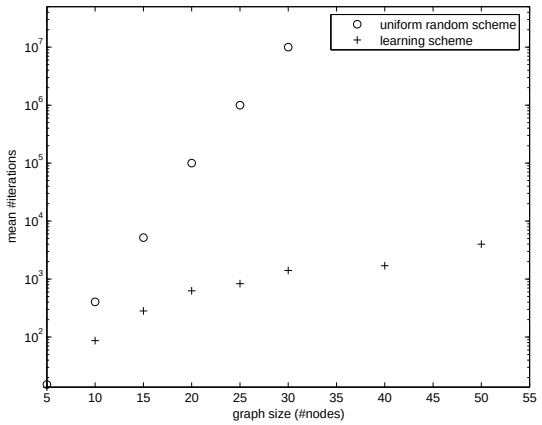


Figure 6: Mean number of iterations to converge to an optimal channel allocation vs. number of nodes in interference graph (random disk graphs, R=0.5, mean over 1000 graphs, #channels = χ, $b = 0.1$).

That is, the stopping time probability decays exponentially. Our argument does not yield a tight estimate of the exponent γ, which determines the precise convergence rate of the algorithm, but given that the underlying colouring problem is NP-hard this is unsurprising. Characterising the convergence rate is discussed in detail later in this section.

Before proceeding we make the following observations.
(i) Multiple Radios. WLANs where stations are capable of simultaneous use of multiple channels can be accommodated by running multiple copies of the CFL algorithm.
(ii) CSMA/CA. Although both are stochastic algorithms, the proposed CFL algorithm differs from CSMA/CA type algorithms in many fundamental respects. For example, for a given network of WLANs the CFL algorithm converges to a static allocation with no collisions, whereas the CSMA/CA algorithm incurs a persistent collision overhead.
(iii) No need for stopping/restarting. The CFL algorithm is naturally convergent, with no need for heuristic stopping criteria. One consequence is that the CFL algorithm can safely be left running at all times, supporting automatic adaptation to changes in topology. This is important in practice as distributed stopping/restarting is problematic without message-passing.
(iv) Clock Synchronisation/Slotted time. Of key practical importance, we note that Theorem 1 applies without change to the situation where channel updates at nodes are not synchronised. That is, there is no requirement for global synchronisation of clocks across interfering WLANs.
(v) Hidden nodes and Uncooperative nodes. The algorithm converges to a proper channel allocation in the presence of hidden nodes or legacy/uncooperative nodes, although a proper allocation may require a larger number of channels than when such nodes are not present.
(vi) Channels used. The algorithm will usually not identify the channel allocation using the smallest possible number of channels, but rather will find, if possible, a non-interfering channel assignment using the given number of channels. In practice, the 802.11a/b channels are known, and the algorithm can be run using those. Note that interference measures can be chosen so that all non-interfering channel assignments are (approximately) equally efficient.

5.3. Convergence Rate

5.3.1. Impact of Learning

We begin by studying the impact on convergence rate of the learning elements of the CFL algorithm, Steps 3 and 4.

We can remove these steps to yield a crude algorithm which assigns a constant probability to each channel and thus evolves as a uniform random walk over every possible combination of channel allocations. More interesting is a modification of this crude algorithm to add the "stickiness" step 3 whereby an AP

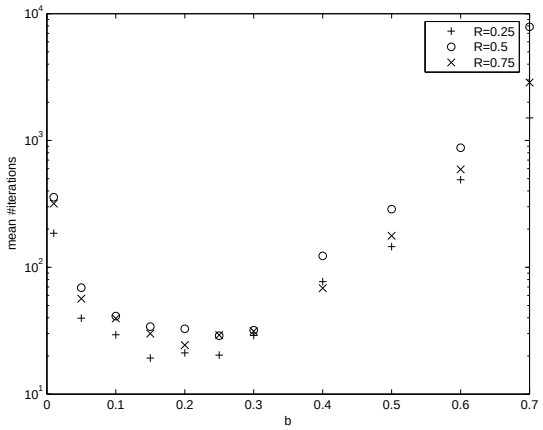


Figure 7: Mean number of iterations to converge to an optimal channel allocation vs. learning parameter b (random disk graphs, random numbers of nodes, #channels $c = 1.25\chi$, mean taken over 1000 graphs).

settles on a successful channel, but which upon failure still assigns uniform probability to every channel (i.e. in the CFL algorithm step 4 is replaced by "On failure update p to $[1/c, 1/c, ..., 1/c]$"). Fig. 6 plots the mean number of iterations to converge versus the number of wireless nodes for this strategy and for the full CFL algorithm. In this example the network interference graph is modelled as a random disk graph (i.e., APs uniformly randomly placed in a unit square and two WLANs interfere when within a radius R of each other). A failure/collision occurs when neighbouring nodes select the same channel at a given iteration of the channel allocation algorithm. For each interference graph the number of channels is set equal to the chromatic number χ (calculated using the DSATUR algorithm [19]); that is, we use the minimum possible number of channels for a feasible solution. The impact on the convergence rate of using larger numbers of channels is discussed in detail later. The convergence time values plotted are the average over 1000 randomly chosen disk graphs. The impact of the learning step 4 is evident: e.g. for a 30 node graph the learning step yields an improvement of four orders of magnitude in mean convergence time.

5.3.2. Choice of Learning Parameter b

The CFL algorithm contains a parameter b that needs to be specified. If rapid convergence required tuning of the parameter to each specific graph, then obviously this would diminish the utility of the algorithm. Instead, we would like there to exist a "universal" choice of b that yields a sweet spot with good performance on a wide range of graphs.

The parameter b determines how quickly an AP discounts previous successes on a channel (or failures on other channels) on experiencing transmission failures on that channel. As b is made larger, failures are penalised more and the "inertia" or "stickiness" of the system decreases. Small inertia allows the system to escape from poor choices of channel allocation but if the inertia is too small then convergence is slowed. Fig. 7 plots the mean number of iterations to converge to a proper channel allocation versus the learning parameter b used. It can be seen that as b approaches 0 the algorithm does not learn from failures and the convergence time rapidly increases. As b approaches 1 the convergence time also rises as a consequence of the small inertia in the system. We can see that values of b in the range 0.1–0.3 yield the fastest convergence times for a range of interference levels (results are shown for interference radius R=0.25, 0.5, 0.75), with the convergence rate largely insensitive to the value used within this range. That is, we can indeed choose a universal value of b that performs well in a wide range of circumstances and does not require case-by-case tuning. In the remainder of this paper we use the value $b = 0.1$ in all examples.

5.4. Impact of Channel Over-provisioning

We now study the impact on convergence of over-provisioning available channels. As the number of available channels increases we expect that the channel allocation problem becomes easier. Fig. 8 plots

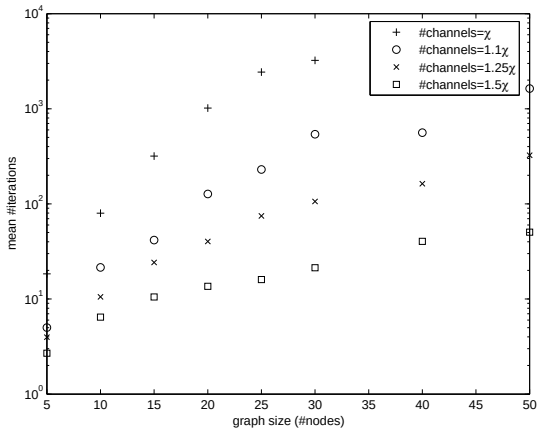


Figure 8: Mean number of iterations to converge to optimal channel allocation vs. number of nodes in interference graph for various levels of channel over-provisioning (random disk graphs, R=0.5, mean over 1000 graphs, $b = 0.1$).

the mean number of iterations for the CFL algorithm to converge versus the channel provisioning (as a percentage of the chromatic number). As expected, we see that convergence time decreases as the level of over-provisioning increases. However, what is most significant is that the impact of even a relatively small amount of over-provisioning can be considerable. For example, 25% additional channels above the minimum required for a feasible solution yields more than an order of magnitude reduction in convergence time, while 50% yields nearly two orders of magnitude reduction. We have found that these reductions are largely insensitive to the interference graph parameters.

We can gain some analytic insight into the great impact of small amounts of over-provisioning as follows. Suppose that there are N_c possible optimal channel assignments when we have χ available channels. How many colourings are there when $\chi + \delta$ channels are available ? We find that the number of optimal channel allocations increases at least exponentially with the number of extra channels provided. In more detail, suppose we have $\chi + \delta$ channels. First we observe that if exactly χ of the $\chi + \delta$ channels are used, there are N_c possible optimal allocations. Hence there are exactly $\binom{\chi+\delta}{\chi} N_c$ optimal allocations using exactly χ channels. If exactly $\chi + 1$ channels are used however, there will be greater than N_c optimal allocations and we can lower bound the total number of optimal allocations $T(\delta)$:

$$T(\delta) \geqslant N_c \left[\binom{\chi+\delta}{\chi} + \binom{\chi+\delta}{\chi+1} + \cdots + \binom{\chi+\delta}{\chi+\delta} \right].$$

Applying Stirling's approximation to the first summand we immediately have $T(\delta) \geqslant N_c \left(\frac{\chi}{\delta} + 1\right)^{\delta}$. Alternatively we can apply the equality $\binom{n+1}{k+1} = \binom{n}{k} + \binom{n}{k+1}$ to get $T(\delta) \geqslant N_c 2^{\delta}$.

6. Too Few Channels

The previous section considers situations where a non-interfering channel allocation exists, i.e. sufficient channels are available to make such an assignment. In this section we extend consideration to situations where there are too few channels to allow a non-interfering allocation. In this situation we consider the channels to be too crowded, and make extra space by having some APs randomly defer any accesses. We present a simple extension of the the CFL algorithm to achieve this via probing and backoff. We show through analysis and simulations that the extended scheme is effective even in situations with few channels.

6.1. Extended Algorithm

When there are too few channels, a key drawback of many channel allocation algorithms is that each WLAN continuously probes and searches for a channel to use. When there are many WLANs and few

channels there is a high probability of multiple stations probing the same channel simultaneously and so low probability of a WLAN successfully winning access to a channel. If the cost of channel collisions is high, this can lead to highly unsatisfactory performance. To address this, we extend the CFL algorithm to include a probability p_{probe} of probing for a channel. The basic idea is to temporarily deny channel access to some WLANs, thus reducing the channel contention. To ensure fairness we adjust p_{probe} using an additive-increase multiplicative-decrease (AIMD) algorithm. That is, we additively increase p_{probe} at rate α until there is "failure", at which point p_{probe} is decreased multiplicatively by factor β. AIMD is used since it is a decentralised scheme (no message-passing) for allocating probing opportunities fairly between contending WLANs. To maintain high efficiency when contention is low we create a degree of "stickiness" whereby when a WLAN successfully wins a channel it continues to use that channel, irrespective of p_{probe}. This yields the following extended algorithm:

Extended CFL algorithm

1. Initialise $p_{probe} \leftarrow 1$, $sticky \leftarrow 0$.
2. $p_{probe} \leftarrow \min[1, p_{probe} + \alpha]$ // additive increase.
3. Draw a random number u uniformly from [0,1]. If $u < p_{probe}$ OR $sticky = 1$, select a channel using CFL algorithm and probe. Otherwise return to step 2.
4. On success, $sticky \leftarrow 1$.
5. On failure, $p_{probe} \leftarrow p_{probe} \times \beta$, $sticky \leftarrow 0$.
6. Update channel probabilities using CFL algorithm.
7. Return to 2.

6.2. Choice of α and β

The extended CFL algorithm inherits the convergence property of the CFL algorithm when a non-interfering channel allocation exists (Theorem 1 can be modified to cover the extended algorithm). When there are too few channels for a non-interfering allocation to exist, the values of the AIMD parameters α and β determine the efficiency of the algorithm.

Roughly speaking, a trade-off exists between efficiency and delay. For small values of α, p_{probe} increases slowly. This reduces contention and, due to the stickiness feature of the algorithm, once a WLAN wins access to a channel it is likely to retain the channel for a long period of time. Hence, efficiency (channel utilisation) is high. However, delay is also high as other WLANs are denied access for long periods because of their low p_{probe} value. In contrast, for large values of α, p_{probe} increases quickly, contention is high and efficiency falls due multiple WLANs probing a channel at the same time. This behaviour is illustrated in Fig. 9, which plots the *success probability* and *mean idle time between successes (MIBS)* vs. α. Success probability is the probability that one of the WLANs has successfully won access to the channel and so is a direct measure of efficiency. MIBS is the mean time interval between a WLAN winning access to the channel and so provides a measure of delay — a large value of MIBS means that WLANs can be denied access for long periods of time. From Fig. 9 we find that $\alpha = 0.01$ offers a good tradeoff between efficiency and delay, and we observe similar results for other interference scenarios and values of β.

With regard to the value of β, when β is large WLANs do not reduce p_{probe} by much on failure. Thus p_{probe} tends to be large, contention is high and efficiency low. This is illustrated in Fig. 10 which plots success probability and MIBS vs. β (this plot is for $\alpha = 0.01$ but similar results are obtained for other values of α). We select $\beta = 0.15$ as providing high efficiency and low delay. From now on we fix $\alpha = 0.01$ and $\beta = 0.15$.

6.3. Performance

Fig. 11 plots the network capacity vs. number of channels for a complete interference graph (where each WLAN interferes with every other WLAN). The capacity is the number of successes achieved by all WLANs divided by the simulation time. It can be seen that the capacity increases with the number of channels, as

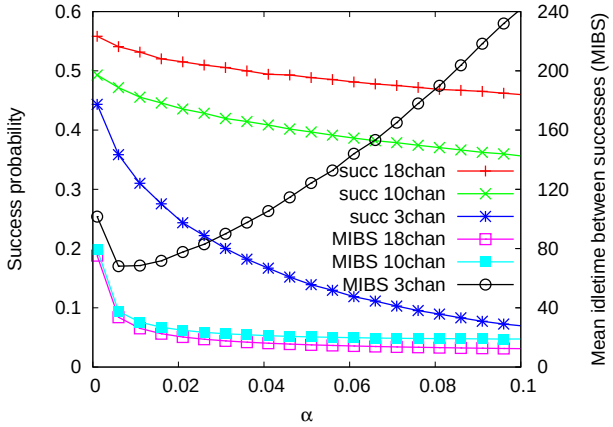


Figure 9: Success probability and Mean idle time between successes vs. α and number of channels. Results are shown for a 50 node complete interference graph with 3, 10 and 18 channels, $\beta = 0.15$.

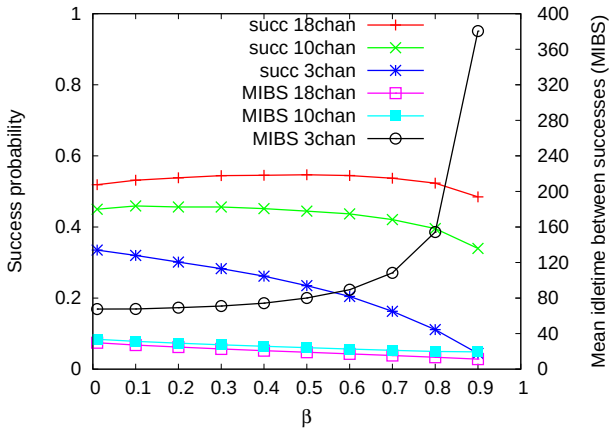


Figure 10: Success probability and Mean idle time between successes vs. β and number of channels. Results are shown for a 50 node complete interference graph with 3, 10 and 18 channels, $\alpha = 0.01$.

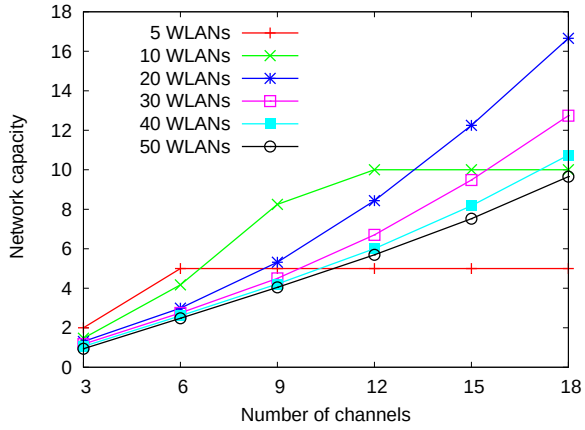


Figure 11: Network capacity vs. number of channels (complete graph, $\alpha = 0.01$ and $\beta = 0.15$).

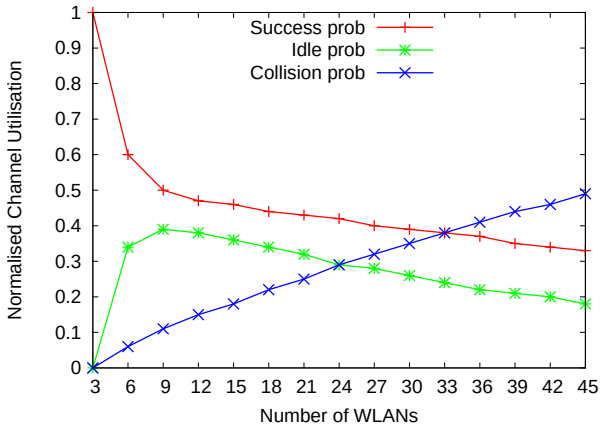


Figure 12: Channel utilisation vs. number of WLANs, 3 channels ($\alpha = 0.01$ and $\beta = 0.15$).

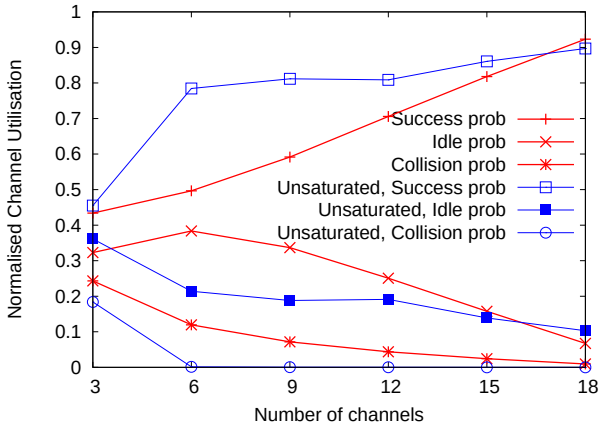


Figure 13: Channel utilisation vs. number of channels for 20 WLANs with saturated/unsaturated demand ($\alpha = 0.01$ and $\beta = 0.15$).

is to be expected. When the number of channels is large enough that a non-interfering channel allocation exists (e.g. for 5 WLANs when the number of channels ≥ 5) a maximum capacity equal to the number of WLANs is achieved. For a smaller number of channels the capacity degrades gracefully. The scheme achieves a high level of fairness among the competing WLANs: Jain's fairness index is greater than 98% for all scenarios in Fig. 11. Fig. 12 plots the success probability, channel collision and idle probability for 3 channels. When there are 3 WLANs, each WLAN tunes to a different channel and so the success probability is one and both channel collision and idle probabilities are zero. As we increase the number of WLANs, and so the level of contention, there is an approximately linear increase in the channel collision probability. The success probability initially falls quite quickly up to 9 WLANs and thereafter decreases more slowly. The selection of β has an impact on this behaviour. Smaller values of β cause a larger back-off of p_{probe}, thus initially resulting in an increase in idle time as the number of WLANs is increased. However, as the level of contention increases further, the larger back-off helps to reduce the collision probability, which is essential for the WLANs to win successful access to the channels when the numbers of interfering WLANs is large.

In all of the above simulations we consider saturated demand at each WLAN. Therefore, every WLAN always contends for a channel. However, in reality we do not expect demand to be saturated. Fig. 13 illustrates the performance of the extended CFL algorithm with exponentially distributed load. The mean arrival rate μ is selected as $\frac{e}{3}$ where e is the measured success probability with saturated traffic. Arrivals accumulate and a WLAN probes for channel access once it has enough demand to utilise the channel at the next timestep. Fig. 13 plots the channel success, collision, and idle probabilities versus number of channels, under saturated and unsaturated conditions. It can be seen that under unsaturated conditions, the success

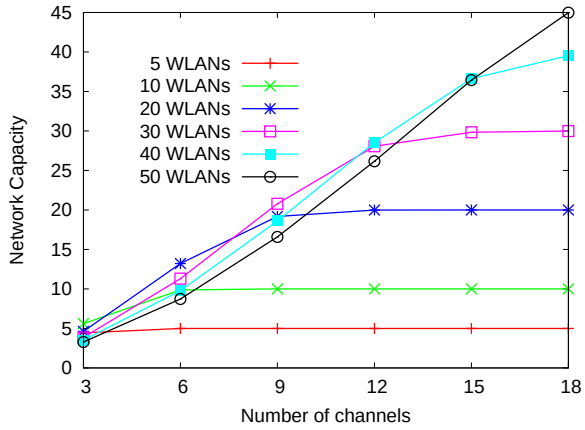


Figure 14: Network capacity vs. number of channels (mean over 1000 random disk graphs, R=0.5, $\alpha = 0.01$ and $\beta = 0.15$).

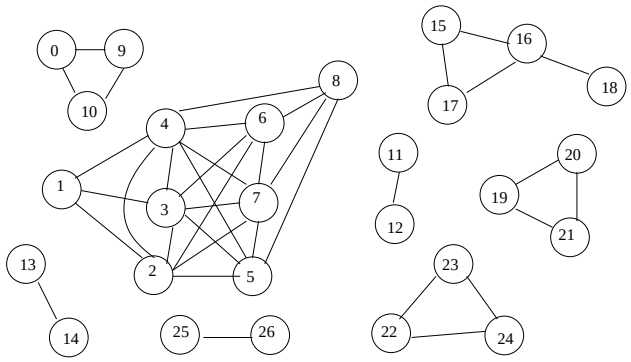


Figure 15: Topology used for MAXchop comparison, drawn from Wigle topologies [5].

probability is increased while the collision probability is reduced.

The foregoing results are for a complete interference graph, corresponding to the case where each WLAN interferes with every other. Fig. 14 illustrates the performance vs. the number of channels for random disk graphs. Again, it can be seen from Fig. 14 that as we provide more channels, the capacity increases until it reaches a maximum value which is just the number of WLANs (at this point we have a non-interfering channel allocation). However, we need fewer channels compared to Fig. 11 to reach this maximum because of spatial reuse. The efficiency of the extended CFL algorithm is reflected in the fact that, subject to this maximum, the capacity is almost invariant with the number of WLANs. For example, with 6 channels the capacity is approximately 10, regardless of the number of contending WLANs.

6.4. Comparison of CFL, Extended CFL and MAXchop

In this section we give a brief comparison of the performance of the CFL scheme, the Extended CFL scheme and the MAXchop scheme [5]. MAXchop is a scheme that requires interfering WLANs to exchange a "hopping sequence", which is a list of channels that the WLAN plans to use. MAXchop adjusts the hopping sequence for each WLAN reducing the number of conflicts. MAXchop is shown to be competitive with a centralised scheme [9].

We use a Wigle-based topology used for the original evaluation of MAXchop (from Figure 3, [5]). The topology is shown in Figure 15. The topology of 26 WLANs requires 5 channels to operate without conflicts, though some components of the graph require only 2 channels. We run each algorithm for 10,000 steps and calculate the throughput over these steps. We consider two simplistic formulas for the throughput: (*sharing*) a WLAN gets $1/(n+1)$ throughput in a step where n WLANs interfere with it; (*zeroing*) a WLAN gets no throughput in a step with interfering WLANs. These represent two extremes of WLAN interference: *sharing* represents WLANs close enough so that the MAC protocol shares the medium and

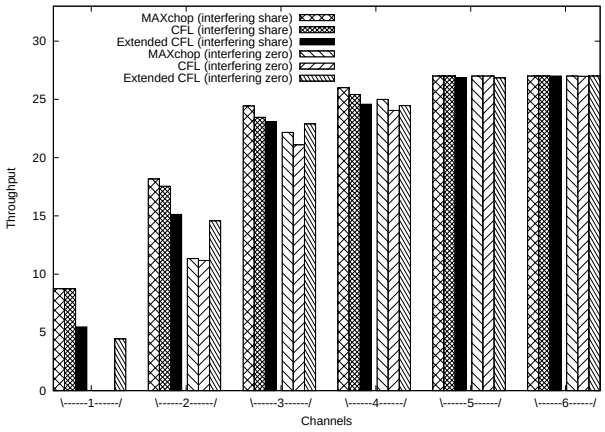


Figure 16: Throughput performance of CFL, Extended CFL and MAXchop as the number of available channels is varied.

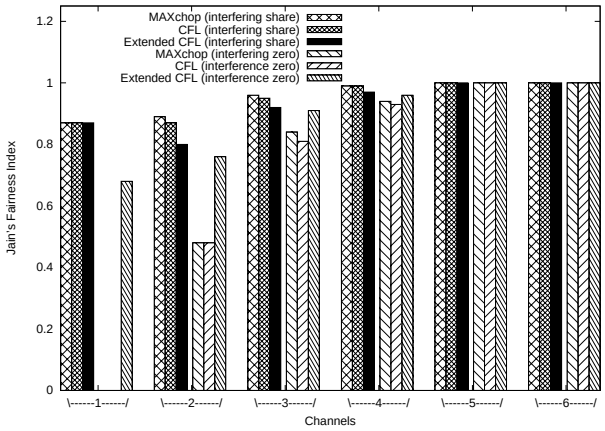


Figure 17: Achieved fairness of CFL, Extended CFL and MAXchop as the number of available channels is varied.

zeroing represents WLANs that cause serious hidden node problems for one another. In Section 8.7 we will see that both these extremes and combinations thereof are possible in practice. MAXchop is implicitly designed for the possibility of sharing.

Fig. 16 shows the throughput performance of each scheme as the number of available channels is varied. Results are shown for the three schemes (CFL, Extended CFL, MAXchop) for both sharing and zeroing throughputs. In all cases the throughput for MAXchop and CFL are close. The throughput of MAXchop can be slightly higher than CFL. This is to be expected because it works with more information.

Note that when 5 or 6 channels are available, all the schemes achieve the maximum throughput of 27. For 3 or 4 channels, the performance of sharing vs. zeroing results in a small drop in throughput for CFL and MAXchop. The Extended CFL algorithm produces slightly lower throughput than CFL/MAXchop for sharing. However, it does not see a drop in throughput for zeroing. When zeroing, CFL and MAXchop begin to struggle when short of channels. Indeed, they both get zero throughput when presented with a single channel.

Fig. 17 shows the corresponding value of Jain's fairness index. MAXchop is designed with fairness as a design goal. Again note that the CFL scheme is close to MAXchop, and that the Extended CFL scheme offers improved fairness over MAXchop in zeroing scenarios.

These results suggest that the performance of CFL is comparable with MAXchop, while working with less information. MAXchop typically changes channel multiple times per step, and thus usually requires more channel hops than CFL. The Extended CFL scheme offers improved performance over both CFL

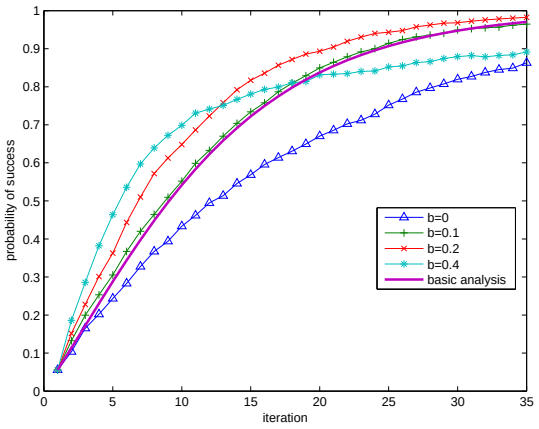


Figure 18: Probability of convergence vs. iteration and choice of parameter b following addition of a new WLAN. (10 node complete interference graph, simulation probability is the mean from 1000 runs).

and MAXchop with fewer channels and zeroing. This is unsurprising, as we have tuned the parameters of Extended CFL to deal with this type of situation. It seems likely that in many practical situations, CFL will be sufficient. However, given a particular deployment, a designer could re-tune Extended CFL to the expected interference behaviour.

7. Time-varying Topologies

We want the CFL scheme to be parsimonious in reacting to changes in network conditions (avoiding unnecessary channel switches) yet adapt rapidly when needed. We consider the impact of variations in the interference graph over time on the performance of the CFL scheme. Time-variations might arise from many factors including changes in traffic load, mobility, changes in environment, and so on. Our experience suggests that good adaptive behaviour of an allocation algorithm is important if it is to be practically effective, because unforeseen circumstances, such as time-varying channel-dependent noise/interference, may stymie schemes that make strong assumptions about the environment.

7.1. Perturbation Analysis: Adding a New Node

We can gain insight into the impact of changes in the network interference graph by considering a network with an optimal channel allocation and adding a new WLAN.

Consider, for the moment, a network where the interference graph is complete (every WLAN interferes with every other WLAN) with N WLANs and $c = N + 1$ channels. We will extend consideration to more general situations later. Suppose that the WLANs in this original network are using the CFL algorithm and have converged to an optimal non-interfering channel allocation using the N channels, $\{1, 2, .., N\}$. We now add a new WLAN to now get an $N + 1$ node complete interference graph. Letting $F_{N+1}(k)$ denote the probability that the new WLAN experiences a failure at iteration k, then $S_{N+1}(k) = 1 - F_{N+1}(k)$ is the probability of success. Fig. 18 plots $E[S_{N+1}(k)]$, the mean probability of success (i.e. convergence to a non-interfering channel allocation), obtained by averaging over 1000 simulation runs for the new WLAN following its addition to the network. Fig. 18 shows results as b is varied. It can be seen that choosing values of b in the range 0.1–0.2 yields the fastest convergence, which is in good agreement with the previous results in Fig. 7.

Also shown in Fig. 18 are predictions corresponding to the following basic analysis. Let $p_{N+1}(k)$ denote the probability of the new WLAN choosing channel $N+1$ at iteration k. Assume that the channel allocation of WLANs in the original network remains unchanged. Then on a collision p_{N+1} is updated according to $E[S_{N+1}(k)] = 1 - E[F_{N+1}(k)]$.

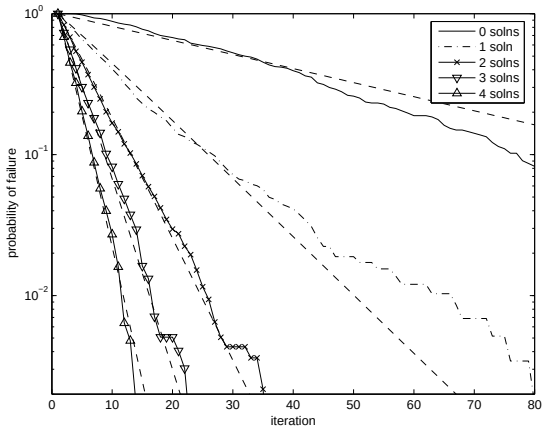


Figure 19: Probability of failure vs. iteration following addition of new WLAN. Dashed lines are analytic predictions. "Solutions" refers to the number of possible channels that the new WLAN may select to achieve a proper channel allocation without disturbing the original allocation. (20 node random disk graph, R=0.5, #channels c=12 (1.25χ), $b = 0.1$).

It can be seen from Fig. 18 that the predictions of this analysis are accurate for the case when $b = 0.1$, indicating that the channel allocation in the original network possesses sufficient inertia that it effectively remains unchanged (this was also confirmed by direct inspection of the network channel allocations before and after the addition of a new WLAN).

Recall, of course, that the "stickiness" or "inertia" of the channel allocation to the original WLANs depends upon the value of the algorithm learning parameter b: $b = 0$ prevents changes in an allocation once it has been successful, but changes become more likely as b is increased. For larger values of b, the new WLAN generates collisions with the original WLANs that result in them choosing new channels and the algorithm must find a new allocation for the whole network rather than just the new WLAN. While the predictions of our basic analysis are accurate for small values of b, they become more inaccurate for large values of b as the key assumption of the analysis, that the original network effectively retains its original channel allocation, is violated.

While the foregoing analysis is for networks with complete interference graphs, the arguments carry over directly to general situations. For example, Fig. 19 shows simulation results for a network with a random disk interference graph together with the corresponding analytic predictions. In this case the interference graph is a random disk graph G. We then randomly[5] add a single new WLAN and record the probability of success and the number of collisions that occur. We do this repeatedly (always starting from the same network and randomly adding one new WLAN) to sample the distribution. A little care has to be taken in applying our analysis as there may be more than one channel that the new WLAN can select to find a proper channel allocation without disturbing the allocations of the original WLANs. We therefore condition upon the number of local solutions and bin the simulation data according to the number of local solutions when comparing against the analytic predictions, see Fig. 19. The foregoing analysis can be applied provided there exists at least one local solution, and it can be seen from Fig. 19 that in such situations it yields remarkably accurate predictions.

It can of course happen that there exist *no* local solutions, i.e. all of the available channels are already used by the neighbours of the new WLAN. This situation is marked as the "zero solutions" curve in Fig. 19. In this case a non-local re-allocation of channels is necessary in order to achieve a non-interfering channel allocation and the previous analysis cannot be applied. We can nevertheless carry out an approximate analysis of this case as follows. Denote the set of WLANs neighbouring the new WLAN by $\mathcal{N}$. We know that these WLANs make use of all available channels. Our measurements on many hundreds of thousands

[5]The network is located in the plane, making it straightforward to add a new WLAN: we select uniformly random coordinates for the new WLAN and determine its neighbours using the interference radius.

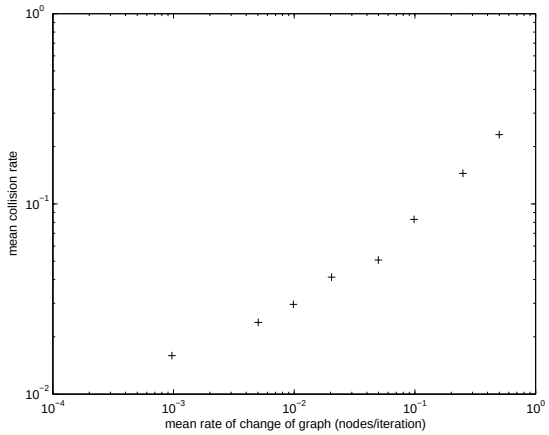


Figure 20: Overhead (failure rate) induced by topology changes vs. rate of interference graph change. (Mean number of nodes 20, nodes added/deleted with mean rate given by the x-axis to random disk graph, radius R=0.25, 5 channels available, average over 100 graphs, $b = 0.1$).

of disk graphs indicate that we almost never see adjacent nodes such that the neighbourhoods of *both* nodes make use of all available channels. We therefore assume that the neighbours $\mathcal{N}$ do themselves have the freedom to change channel. Consider the behaviour of the new WLAN: because there is no local solution it must choose the same channel as one of its neighbours. By assumption, a neighbour will change channel with probability at least $b/(c-1)$ and otherwise stay on the same channel. Note that it can occur that more than one neighbour shares the same channel, in which case we need all such neighbours to change channel in order to free up that colour. We neglect this possibility because simulations show it is rare. Hence our model predicts that independently at each timestep, the system will re-converge approximately with probability at least $b/(c-1)$. The accuracy of this approximation is illustrated in Fig. 19.

This analysis can be used to make quantitative predictions of convergence rate, provided we know the local structure in a neighbourhood of interest. However, the real value lies in the qualitative insight that the addition of a new WLAN is accommodated parsimoniously and results in minimal changes in the original channel allocation. This property is the key to achieving high performance in time-varying environments.

7.2. Persistent Perturbations

We now illustrate the impact of persistent, multiple changes to the interference graph, rather than the one-off addition of a singe new node. Based on the above previous section, we expect that when changes in the interference graph occur slowly (compared with the convergence time of the channel selection), then they will induce only minimal channel reallocation and the level of failures will be small relative to the number of successful outcomes.

Fig. 20 presents simulation results showing the mean channel selection overhead ("failures" as a proportion of all channel selection outcomes) as a function of the rate of change of the interference graph. The results are the average over 100 tests each with a mean of 20 nodes. In each test we start with a random disk graph. Nodes are randomly added/deleted at time intervals which are exponentially distributed, with the rate of change of the interference graph given by the reciprocal of the mean number of iterations between node addition/deletion. A maximum of only 5 channels are assumed available (so that at some instants the channel allocation problem may in fact not be feasible). It can be seen that, as expected, the overhead increases with the rate of change of the network interference graph. Observe, however, that the absolute overhead remains low even in rapidly changing conditions; for example, the overhead is only 10% even when a node is added/deleted from the network on average every 5 iterations.

8. Experimental Results

We have implemented a prototype version of the CFL algorithm on a Linux. This required no hardware modifications and a simple software implementation. We test its performance under real interference conditions in an office environment. Verifying performance in a real environment is essential, as complex simulation environments may not exhibit the variety of features seen in Section 4.

8.1. Implementation of CFL Algorithm

The CFL algorithm is implemented as a user-space perl script that runs on each WLAN AP. WLAN-wide channel switching is achieved by a broadcast instruction from the AP that is received by a script running on each WLAN client station, which then uses the *iwconfig* command to change channel. Ultimately, the 802.11s standard could be used to request channel changes. UDP Traffic is generated using *mgen* and traffic is directed from the AP to the clents. Packets have a size of 1400 bytes and are generated at a rate of 1000pps.

The CFL algorithm requires a measure of channel quality. We initially investigated using the RSSI value returned by the AP wireless NIC. However, we found this value to be unreliable — when channel quality is degraded due to interfering WLANs it is quite possible for the background noise level to be low yet for the frame error rate to be high due to colliding transmissions. We therefore use a direct measure of frame error rate as our channel quality metric.

For our prototype implementation, to allow scripting entirely within user-space we took advantage of the RTS/CTS functionality. Using *tcpdump* to monitor packets transmitted, over 10 second intervals we collected statistics on (i) RTS transmissions for which no corresponding CTS handshake was received, (ii) transmissions for which the RTS/CTS handshake was successful but the data packet transmission was not paired with a MAC ACK, and (iii) transmissions with successful RTS/CTS and data/ACK handshakes. We label (i) as CSMA/CA collisions, (ii) as frames lost due to interference and (iii) as successful transmissions. The first of these labels is only approximate as RTS/CTS handshakes may be lost due to interfering transmissions or noise. However, the CFL algorithm only requires a coarse good/bad measure of channel quality and we find that measuring channel quality by the percentage of type (ii) events and thresholding at 10% is quite effective. In addition, for some tests we augmented this metric to include a test for the presence of beacon packets from alien WLANs — see section 8.4 for details.

We note that the use of RTS/CTS creates an overhead that can reduce network capacity. However, it means our criterion for successful transmissions depends on the successful decoding of the CTS/ACK, and so it accounts for the the bidirectional nature 802.11 links. While RTS/CTS is sufficient for proof of concept, we have also been investigating more sophisticated measures of link-quality [22, 23], which indicate it is possible to estimate bidirectional link quality from the AP. In a larger network an AP could combine multiple link quality measurements, either acquired locally or using 802.11k, using some fairness policy to arrive at a final decision regarding whether the channel is of good/bad quality.

8.2. Convergence to non-interfering channel allocation

To demonstrate the operation of the CFL algorithm for channel selection, we simultaneously generated traffic between the nodes on each of the five WLANs in Fig. 1. To create a relatively demanding channel allocation task, the channel allocation algorithm was restricted (via scripting) to use four 802.11a channels. Initially, all WLANs are started on the same channel and a copy of the CFL algorithm runs on each WLAN to learn a non-interfering channel selection.

We emphasise that there is no message passing whatsoever between the WLANs — the only information available to each WLAN is its local measure of channel quality. Local channel quality is measured based on packet trace statistics over a 10 second sampling interval.

Fig. 21 shows traces of the channel selection time histories for the five WLANs as we run the CFL algorithm. Throughput significantly increases once a non-interfering channel allocation is selected, yielding a substantial increase in network capacity: the aggregate throughput from 50–60 seconds is approximately 51 Mbps compared with an aggregate throughput of approximately 12 Mbps when the WLANs all use the same channel. That is, we obtain approximately a factor of four increase in network capacity through

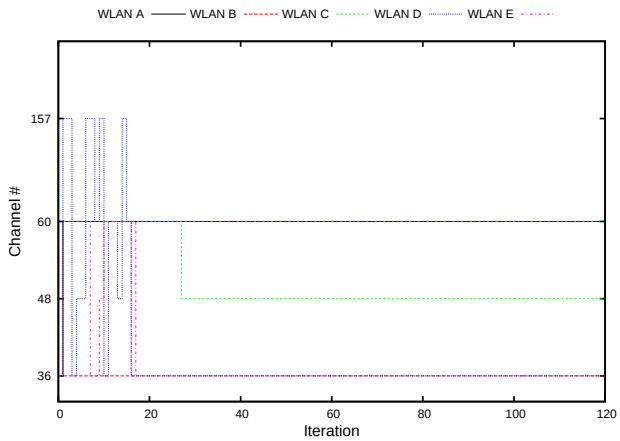


Figure 21: WLAN channel time histories. Five WLANs, four available channels. Note that in this example the network settles on only three channels.

WLAN	Default Channel Selection (Mbps)	CFL Channel Selection (Mbps)
WLAN A	2.56	12.90
WLAN B	3.86	8.08
WLAN C	2.58	12.69
WLAN D	2.39	5.84
WLAN E	1.51	12.02
Totals	12.93	51.55

Table 2: Throughputs, 5 WLANs and 4 available channels.

appropriate channel selection. Table 2 gives detailed measurements. Similar results were obtained across many runs, confirming that this level of capacity improvement is consistently achieved.

In practice, we expect that this will result in substantial improvements for users. In our example, Table 2 shows that each network sees an improvement in throughput by a factor of at least two, which should decrease the times for file/attachment downloads by a factor of two. For interactive traffic, the reduced number of retransmissions due to interference will result in lower latencies.

8.3. Convergence Rate

It can be seen in Fig. 21 that the network converges to a non-interfering channel allocation in approximately 20 iterations. The duration of an iteration is determined by the time required to sense channel quality and is set to 10s in our tests yielding an overall convergence time of 200s.

First note that during this convergence period the network continues to achieve a significant level of throughput. This is illustrated in Fig. 22, which plots cumulative packets number received versus time for each of the five WLANs for a second example. Hence, the cost of the convergence period in terms of throughput is limited.

Second, the foregoing simulation analysis indicates that the CFL algorithm converges rapidly under a wide range of conditions and this is confirmed in our experimental tests. For example, the mean convergence time measured over 10 tests is five iterations with five WLANs and four available channels.

8.4. Controlling local channel reuse

Observe in Table 2 that WLAN B and WLAN D settle on the same channel. It can be seen from Fig. 1 that these WLANs are close together. Inspection of packet traces shows that the nodes in these WLANs are visible to each other with no hidden nodes. That is, both nodes involved in a collision are able to detect that the collision occurred, thus the 802.11 CSMA/CA MAC is able to schedule transmissions properly and the frame error rate (i.e. packet losses not associated with CSMA/CA collisions) is low. Since our objective here in allocating channels is to avoid hidden node and interference related problems, this behaviour is as expected. Indeed, it is desirable in dense deployments as it increases the level of channel reuse. That is,

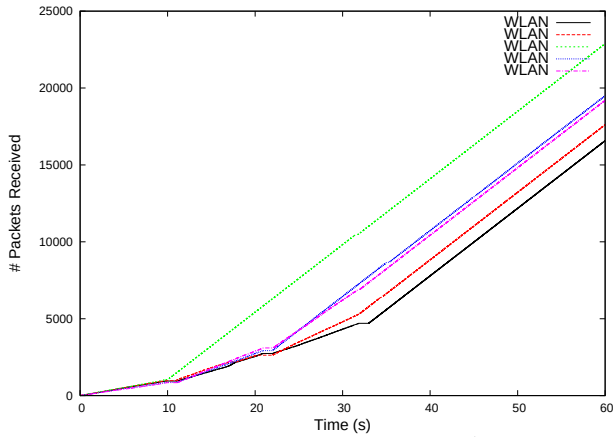


Figure 22: Cumulative packets received versus time for each of the five WLANs. Five WLANs, four available channels.

WLAN	Default Channel Selection (Mbps)	CFL Channel Selection (Mbps)
WLAN A	1.52	13.05
WLAN B	4.54	12.99
WLAN C	3.41	12.97
WLAN D	1.41	12.57
WLAN E	0.85	12.45
Totals	11.73	64.03

Table 3: Throughput, 5 WLANs and 4 channels. Foreign beacons used to allocate co-located WLANs on distinct channels.

channel reuse is possible not only between WLANs located so far apart that their transmissions do not interfere, but also between WLANs located close together so that CSMA/CA operates correctly.

If desired it is straightforward to force nearby WLANs to use different channels (channel reuse is then confined to WLANs located sufficiently far apart). To illustrate this, we augmented our channel quality metric to include not only frame error rate but also detection of beacon frames from foreign WLANs. Channel quality is judged unacceptable if the frame error rate exceeds 10% or if foreign beacons are detected. The CFL algorithm itself is unchanged.

Table 3 gives an example of measured performance when this change is made — it can be seen that the network capacity increases from 11.73 Mbps to 64.03 Mbps, with each station achieving a throughput of close to 13Mbps which the achieved throughput measured in a single isolated WLAN (no interference). Note that in this example the five WLANs make use of only four channels, and the CFL algorithm successfully exploits the potential for spatial reuse in our testbed.

8.5. Impact of external and channel dependent interference

Our measurements of the testbed interference environment, in Section 4, highlighted the presence of external interference sources in the 2.4GHz band, and the channel dependent nature of the level of interference between WLANs.

Returning to the channel dependent interference between WLANs B and C noted in Fig. 5, we recorded statistics on the channels selected by these WLANs over a series of 10 tests. In line with Fig. 5 we find that, as expected, the CFL algorithm settles on either channel 36, 40 or 64 and avoids the lower quality channels. Similarly, in the case of WLAN E, Fig. 3 shows that the quality of certain channels can be strongly time-varying. We can also observe in Fig. 3 that certain channels are consistently of good quality, e.g. channels 36–44 and 60–64. Measurements confirm that the CFL algorithm automatically avoids poor quality channels and settles on good quality ones. This also indicates that the CFL scheme will also work around WLANs operating on fixed channels, that are not using CFL.

Our experience suggests that this adaptive behaviour of the CFL algorithm is a key feature that a channel allocation algorithm must provide if it is to be practical, although the issue of channel dependent

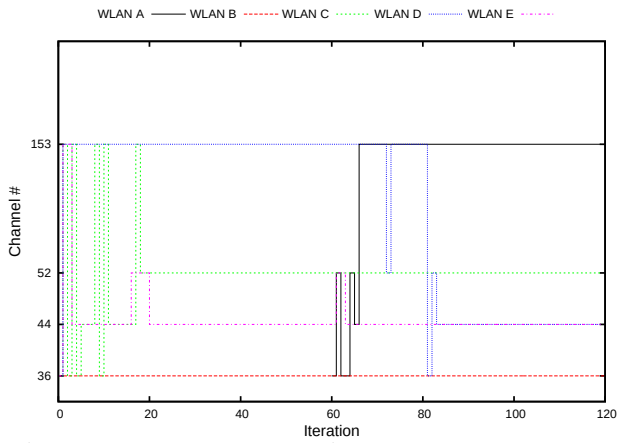


Figure 23: Example of a new WLAN becoming active. Four WLANs active initially, with fifth WLAN beginning transmissions at time 60.

WLAN	Throughput for Default Channel Selection (Mbps)	Throughput for CFL Channel Selection (Mbps)	
		at time 50	at time 120
WLAN A	0.45	12.45	12.12
WLAN B	4.77	12.90	11.71
WLAN C	0.00 (inactive)	0.00 (inactive)	11.76
WLAN D	2.27	12.21	12.57
WLAN E	0.74	12.56	12.15
Totals	11.31	50.12	60.31

Table 4: Throughputs, 4 WLANs active initially, with fifth WLAN beginning transmissions at time 60.

noise/propagation and the strong spatial variation in channel quality does not seem to have been widely considered in the WLAN channel allocation literature.

8.6. Time-varying network conditions

The level of interference between WLANs is dependent on the traffic load on each WLAN. In particular, when a WLAN carries no traffic it generates essentially no interference. Importantly, when a WLAN that has been inactive becomes active, a channel must be allocated to that WLAN and this may require reconfiguration of the channel allocations used by other nodes. Since the CFL algorithm is convergent (i.e. stays settled on a non-interfering channel allocation once it has found one), it can be left running at all times. Changes in the network that create new interference will then automatically activate the CFL algorithm to adapt the channel allocation to restore a non-interfering allocation. This is illustrated in Fig. 23. Here, we start with four WLANs which quickly settle on a non-interfering channel allocation. At iteration 60 of the CFL algorithm, a fifth WLAN is activated (i.e. begins transmitting traffic). It can be seen that the network automatically reconfigures its channel allocation to accommodate this new WLAN and quickly settles on a new non-interfering configuration. Table 4 gives the corresponding WLAN throughputs.

8.7. Spatial Reuse

In 802.11b/g there are three so-called "orthogonal" channels (channels 1,6,11). However, this is a simplified view as there is not true orthogonality between channels but rather a degree of attenuation of transmit power. For example, the 802.11 standard specifies a spectral mask for each channel than requires 30dB attenuation between approximately every other channel (e.g. between channels 1 and 3). When combined with attenuation due to the physical separation of transmitters, this means that nearby channels can in fact effectively be "orthogonal". If we allow the CFL algorithm to use all 11 channels in 802.11b/g, nearby channels with insufficient attenuation between transmissions will automatically be avoided.

To investigate the level of spatial reuse feasible in our testbed, we measured the frame error rate between pairs of WLANs as the channel used by one WLAN was varied. Initially we consider the behaviour when the 802.11a 5GHz band is used. Fig. 24(a) shows the measured throughputs of WLANs A and E when WLAN E is held fixed on channel 36 while the channel used by WLAN A is varied between channel 36 and channel 64. Fig. 24(b) shows the corresponding measurements for WLANs C and E. Note that unlike in 802.11b/g, 802.11a channels are not numbered consecutively i.e. channels 36 and 40 are in fact adjacent. Observe from Fig. 1 that WLANs A and E are located adjacent to each other whereas WLANs C and E are located approximately 10m apart. We therefore expect that a larger separation in channels is needed between WLANs A and E than between WLANs C and E and indeed our measurements support this prediction.

It can be seen that when WLAN A is located on channel 56 and above, the aggregate network throughput is 26Mbps which is approximately the maximum combined capacity that can be achieved by two independent WLANs for the 802.11a settings used here. Observe also that both WLANs achieve similar throughputs i.e. network capacity is allocated equally. However, when WLAN A is on a channel that is closer to that of WLAN E we have that (i) the aggregate network throughput falls substantially and (ii) the WLANs can experience dramatically different throughputs (e.g. when WLAN A uses channels 44 or 48 it achieves a throughput close to zero, while WLAN E achieves throughput close to 12Mbps). The latter unfairness is associated with hidden node type effects that occur when the WLANs operate on channels that are sufficiently close for their transmissions to interfere yet not so close that they can successfully decode each others transmissions. When the WLANs operate on the same channel, they can decode each others transmissions since the WLANs are located near to each other and thus the 802.11 CSMA/CA operation fairly allocates the available bandwidth. However, the aggregate network throughput is half that achieved when the WLANs operate on orthogonal channels.

This behaviour can be contrasted with that of WLANs C and E. From Fig. 24(b) we see that even when WLANs C and E use adjacent channels the aggregate network throughput is nevertheless close to 26Mbps. WLANs C and E are located only 10m apart, yet the attenuation due to walls etc. when combined with the attenuation between adjacent channels is sufficient to effectively yield orthogonality of transmissions.

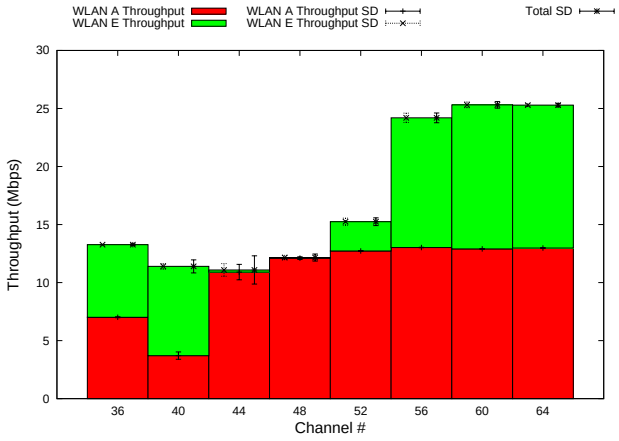
To further investigate this, Fig. 25 shows the corresponding measurements when the 802.11b channels are used. Note that in these 802.11b measurements there exists significant background noise on channels 7–10 for WLAN C — (see Section 4). The impact of this noise can be seen in Fig. 2, which plots the measured frame error rate on different channels when WLAN C alone is active — the high frame error rate on channels 7–10 is evident.

Focusing for the moment on channels 1–6 where the background noise level is low, it can be expected that the level of transmission path attenuation is reduced when lower frequency transmissions are used and indeed this is the case. Nevertheless, it can be seen from Fig. 25 that a spacing of 3 channels is sufficient for effective orthogonality, compared to the 5–6 channels usually quoted based on co-located radios.

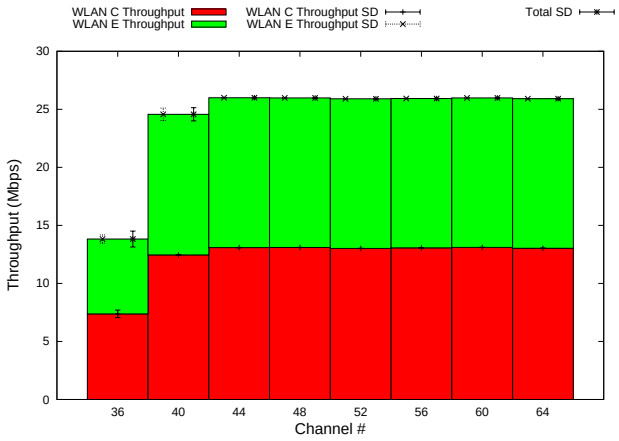
This behaviour is encouraging as it suggests that in practice we do not need to confine allocations to the usual orthogonal channels. Instead, attenuation within a building means that dense deployments can potentially take advantage of aggressive spatial reuse. It is important to emphasise that the CFL algorithm can be used without modification to achieve orthogonal channel allocations even in such complex settings. If we allow the algorithm to choose any of the available channels (e.g. to select from all 11 channels in 802.11b), the CFL algorithm will seek an orthogonal channel allocation, avoiding channel configurations that create interference.

9. Conclusions

In this paper we propose a new class of channel allocation algorithms. The algorithms are simple and robust, require no communication between interfering WLANs, are provably correct when a graph colouring exists and yet remarkably efficient under a wide range of network conditions and topologies, including when a non-interfering channel allocation is infeasible. The algorithms are suited to implementation on standard equipment, requiring no special hardware support and making only light demands on computational resources. We demonstrate this by implementing algorithms on an experimental testbed using commodity



(a) WLANs A and E (x-axis marks channel used by WLAN A).



(b) WLANs C and E (x-axis marks channel used by WLAN C).

Figure 24: Measuring potential for channel reuse. Using the 802.11a 5GHz band, WLAN E is held fixed on channel 36 while the channel used by second WLAN is varied. Measurements are shown for WLANs A and E and for WLANs C and E. Height of histogram indicates aggregate throughput of both active WLANs. Light shaded area marks throughput of WLAN E and dark shaded area marks throughput of second WLAN. Also marked on the histogram are the standard deviations of the throughput, which give a measure of throughput variability — it can be seen that the standard deviations are consistently low. WLANs A and E are located adjacent to each other whereas WLANs C and E are located approximately 10m apart.

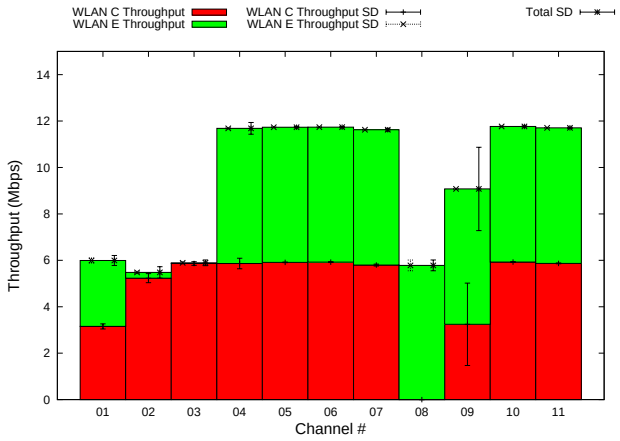


Figure 25: Channel reuse (802.11b). Throughput of WLANs C and E. WLAN E is held fixed on channel 1 while WLAN C's channel is varied.

hardware. We present detailed measurements of performance in a real office environment with complex, spatially varying noise and exhibiting channel dependent interference between WLANs, time-varying channel quality and the presence of external interference sources. Our measurements indicate that the proposed approach offers the potential for effective channel allocation in real environments.

In this work we focus on the channel allocation task, rather than joint channel allocation, routing (and related issues of association control) and power control. Certainly these problems can be coupled, however channel allocation is individually challenging and joint optimisation is left as future work.

References

[1] A. Akella, G. Judd, P. Steenkiste, and S. Seshan. "Self management in chaotic wireless deployments", *in Proc. ACM Mobicom*, pp. 185–199, Sept 2005.

[2] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks", *IEEE Trans Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[3] H. Luo, P. Medvedev, J. Cheng, and S. Lu, "A self coordinating approach to distributed fair queuing in ad hoc wireless networks", *in Proc. IEEE Infocom*, pp. 1370–1379, Apr 2001.

[4] A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network", *in Proc. IEEE Infocom*, vol. 3, pp. 2223–2234, Mar 2005.

[5] A. Mishra, V. Shrivastava, D. Agarwal, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments", *in Proc. ACM Mobicom*, pp. 170–181, Sept 2006.

[6] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, "A client-driven approach for channel management", *in Proc. IEEE Infocom*, pp. 1–12, Apr 2006.

[7] A. Mishra, E. Rozner, S. Banerjee and W. Arbaugh, "Exploiting Partially Overlapping Channels in Wireless Networks: Turning a Peril into an Advantage", *in Proc. Internet Measurement Conference*, Apr 2005.

[8] B.J. Leung and K.K. Kim, "Frequency assignment for IEEE 802.11 wireless networks", *in Proc. IEEE Vehicular Technology Conference*, vol. 3, pp. 1422–26, Oct 2003.

[9] Y. Bejerano, S.-J. Han and L. (Erran) Li, "Fairness and load balancing in wireless LANs using association control", *MobiCom '04*, 2004.

[10] P. Clifford and D.J. Leith, "Channel Dependent Interference and Decentralized Colouring", *NET-COOP 2007*, Jun 2007.

[11] K. Duffy, N. O'Connell and A. Sapozhnikov, "Complexity analysis of a decentralised graph colouring algorithm", *Information Processing Letters*, Volume 107, Issue 2, July, 2008 .

[12] B. Kauffmann, F. Baccelli, A. Chaintreau, K. Papagiannaki, and C. Diot, "Self organization of interfering 802.11 wireless access networks", *INRIA Technical Report*, Aug 2005.

[13] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot, "Measurement-Based Self Organization of Interfering 802.11 Wireless Access Networks ", *Proc. INFOCOM*, May 2007.

[14] L. Narayanan, "Channel assignment and graph multicoloring," *Handbook of wireless networks and mobile computing*, Wiley series on parallel and distributed computing, 2002.

[15] A. Subramanian, H. Gupta, and S. R. Das, "Minimum interference channel assignment in multi-radio wireless mesh networks", *in Proc. IEEE Secon*, pp. 481–490, Jun 2006.

[16] B.J.Ko, V.Mishra, J.Padhye, and D.Rubenstein, "Distributed channel assignment in multi-radio 802.11 mesh networks", `http://www1.cs.columbia.edu/~danr/publish/2006/jun-tr06.pdf`

[17] A. K. Das, S. Roy, and R. Vijaykumar, "Static channel assignment in multi-radio multi-channel 802.11 wireless mesh networks: issues, metrics and algorithms", *in Proc. IEEE Globecom*, pp. 1–6, Nov 2006.

[18] K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks", *in Proc. IEEE Infocom*, pp. 1–12, Apr 2006.
[19] M. Trick, "Network Resources for Coloring a Graph", `http://mat.gsia.cmu.edu/COLOR/color.html`, Oct 1994.
[20] D.J. Leith, and P. Clifford, "A self-managed distributed channel selection algorithm for WLANs", *in Proc. ACM/IEEE Rawnet*, pp. 1–6, Apr 2006.
[21] Clifford.P., Leith,D.J, "Channel Dependent Interference and Decentralized Colouring". *Proc Net-Coop*, Avignon, 2007.
[22] D. Malone, P. Clifford, and D.J. Leith, "MAC Layer channel quality measurement in 802.11", *IEEE Communcations Letters*, vol. 11, no. 2, pp. 143–145, 2007.
[23] D. Giustiniano, D. Malone, D.J. Leith, K. Papagiannaki, "Experimental Assessment of 802.11 MAC Layer Channel Estimators.", *IEEE Communications Letters.* Volume 11, Issue 12, 2007.