WLAN Channel Selection Without Communication

D.J. Leith, P. Clifford, D.W.Malone
Hamilton Institute, National University of Ireland, Maynooth, Ireland

***Abstract*— In this paper we consider how a group of wireless access-points can self-configure their channel choice so as to avoid interference between one another and thereby maximise network capacity. We make the observation that communication between access points is *not* necessary, although it is a feature of almost all published channel allocation algorithms. We argue that this observation is of key practical importance as, except in special circumstances, interfering WLANs need not all lie in the same administrative domain and/or may be beyond wireless communication distance (although within interference distance). We demonstrate the feasibility of the communication-free paradigm via a new class of decentralized algorithms that are simple, robust and provably correct for arbitrary interference graphs. The algorithm requires only standard hardware and we demonstrate its effectiveness via experimental measurements.**

I. Introduction

In this paper we consider how a group of access-points/base-stations[1] 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 problem would go a long way to allowing true plug and play WLAN operation and has recently attracted renewed attention, e.g. see [1], [5], [6], [8], [10], [11], [12], [13].

Historically, common practice in arriving at channel allocations has lain somewhere between (i) a detailed radio survey and careful placement of access points combined with manual spectrum planning, and (ii) placement of access points 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 the use of centralised wireless switch solutions, where channel selection (amongst other things) is delegated to a central automated switch that has a control plane connection to every wireless access point. A key requirement of such a centralised solution is, of course, administrative control of every access point.

Looking ahead, industry roadmaps point towards (i) increasingly dense network deployments (including multi-hop operation) and (ii) multi-channel/multi-radio devices. Dense deployments are already a practical concern, as can be seen from the functionality of centralised wireless switches and also from anecdotal evidence in apartment and office blocks and measurement studies [1]. Multi-hop wireless is already common in community-based WLANs and the subject of at least two IEEE standards-track activities (the 802.11s task group and the recently formed 802.16 MMR study group). Multi-channel devices are already available: most modern 802.11 APs support channel bonding to increase available bandwidth. Multi-radio configurations are already common in community wireless mesh networks and likely to become more prevalent. All of these developments create more pressure on the available spectrum and thus on the channel selection methodology.

Manual channel selection evidently scales poorly with network size. Centralised automated solutions require common administrative control of every interfering access point. Hence, while an excellent niche solution, centralised solutions are clearly problematic in more general contexts where interfering networks may belong to different administrative domains (e.g. interfering wireless networks may be operated by different households or businesses). Over the years, a great many distributed schemes have been proposed in the literature , e.g. see [9], [2], [4], [3], [7], [5], [6], [10], [11], [12], [13] and references therein. However, almost all are distributed in the sense that they require only local communication between access points that directly interfere with one another (perhaps implemented via sniffing of packets rather than by dedicated transmissions, e.g. see [5]). The requirement for message passing means that such schemes suffer from a similar problem to centralised solutions when interfering networks lie in different administrative domains; namely, firewalls and other security devices may hinder explicit communication, while packet sniffing 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 network transmissions interfere (thus interfering access points may well not be able to sniff each others packets). Moreover, most of the proposed distributed schemes are heuristic in nature and come with few performance guarantees (partly due to the NP-hard nature of the channel allocation problem, although NP-hardness only relates to the computational complexity of the problem).

In this paper we suggest that a different approach is possible. Building on the preliminary work reported in [15], we make the observation that to construct *provably correct* decentralized algorithms for channel allocation it is sufficient for each access point to simply be able to sense its own environmental conditions. This requirement is commonly satisfied and creates no need for explicit communication nor for any form of common administrative control. We emphasise that this is very different from recent approaches such as [5], [6], [10], [11], [12], [13] which all require some form of message passing between interfering stations. The closest work to that in the present paper is the unpublished work of Kaufmann et al [8], discussed below.

This work was supported by Science Foundation Ireland grant IN3/03/I346.

[1] In this paper we use the term access point or 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 and the AP here might equally be the access point in an 802.11 infrastructure WLAN, the base station in an 802.16 network, etc. Each AP has associatedwireless client stations and we refer to the collection of clients plus AP as a WLAN.

The principle that communication is not needed for distributed resource allocation is well-known in wireless MAC design, and is for example embodied in the 802.11 CSMA/CA MAC[2]. However, the application of this concept in a soundly-based manner to the channel selection task is new. Direct adoption of a CSMA/CA type scheme is ill-suited to the channel allocation task as it inevitably involves persistent "collisions" and persistent changes in channel, even when the network topology is static.

Our main contribution in this paper is twofold. First we demonstrate the feasibility of the communication-free paradigm for channel allocation by introducing a new class of decentralized algorithms that are simple and robust, provably correct for arbitrary interference graphs, require no communication between interfering WLANs and yet are remarkably efficient under a wide range of network conditions and topologies. These algorithms are suited to implementation on standard equipment, requiring no special hardware support and making only light demands on computational resources. Our second main contribution is the implementation and evaluation of this algorithm in an experimental testbed. We present detailed measurements of the interference environment in a office-based WLAN testbed and highlight a number of features including the time-varying nature of channel quality, the channel dependent nature of interference levels between WLANs and the presence of external interference sources. All of these features have important implications for the development of practically useful channel allocation algorithms. We then implement our decentralized channel allocation algorithm using standard 802.11 hardware and demonstrate, via extensive measurements, that it does indeed offer the potential for effective channel allocation in realistic environments. In particular, this includes environments with complex, spatially varying noise and channel dependent propagation behaviour and with time-varying load.

II. Related Work

The channel allocation task has been the subject of a considerable literature, spanning cellular networks (e.g. see the survey paper [9]), wireless LANs (e.g. see [9], [2], [4], [3], [7], [5], [6], [10], [11], [12], [13] and references therein) and graph theory (our channel allocation task is equivalent to the classical graph-colouring problem). Almost all of this work has, however, been concerned either with centralised schemes or with distributed schemes that employ extensive message-passing. The closest work to that in the present paper is perhaps the unpublished work of Kaufmann et al [8] who study a distributed simulated annealing algorithm for joint channel selection and association control in 802.11 WLANs. Convergence of the annealing algorithm is, however, only achieved asymptotically as temperature tends to zero. Heuristics are therefore needed to stop the algorithm and also to restart it if the network topology changes. Such network-wide stopping/restarting in a distributed context seems problematic without message-passing. Also, for non-zero temperature simulated annealing necessarily has probability of stopping at a non-optimal solution i.e. an interfering channel allocation.

[2]The 802.11 MAC is also able to exploit packet header information, e.g. NAV values, but the basic CSMA/CA operation does not rely upon such information – carrier sense is sufficient.

III. WLAN Channel Allocation Task

We briefly discuss simple examples with the aim of illustrating the channel allocation problem in a WLAN context and of highlighting the very substantial performance gains that can be achieved by a proper choice of channel.

Figure 1 shows an example of interfering 802.11 WLANs. Transmissions within the AP1 and AP2 WLANs can interfere, with the interference range of each WLAN indicated by the dashed circles in Figure 1 (the use of circles is purely indicative). The level of interference between any particular pair of transmissions depends on the physical locations of the communicating stations. This can easily lead to complex hidden/exposed terminal problems. For example, if AP2 transmits data to client 1 at the right-hand edge of the figure at the same time as the client 2 station located at the left-hand edge of the figure sends data to AP1, then reception by AP1 may be blocked by AP2's transmission while AP2's transmission is successfully received at the right-hand station as this is beyond the interference range of AP1. This is, of course, an example of hidden terminal behaviour, known to have the potential to induce gross unfairness and reduced network utilisation. Since AP3 and AP4 are located within communication distance of both AP1 and AP2, their transmissions can similarly interfere creating further potential for four-way hidden/exposed terminal behaviour.

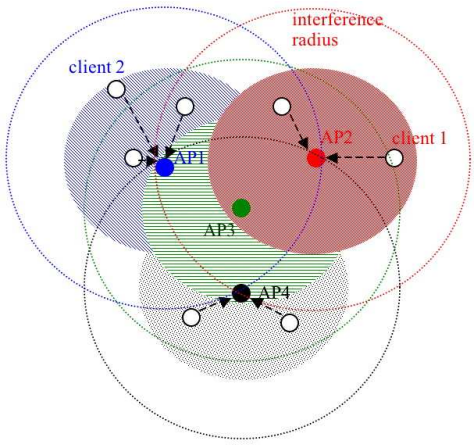


Fig. 1. Example of interfering 802.11 WLANs. Dashed circles indicate interference radius, shaded circles indicate communication radius. The use of circles is purely indicative – we do not assume any constraints on the shape of the interference domains.

Evidently, when all stations use the same channel the interference between stations can lead to complex behaviour that significantly degrades network performance. If, however, each of the APs selects a different channel then the previously described hidden/exposed terminal problems immediately disappear, network behaviour becomes much simpler and overall capacity increased. The challenge is, however, to achieve genuine plug and play operation i.e. for the overall

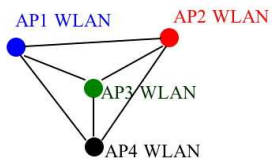


Fig. 2. Interference graph of Figure 1.

network to automatically configure itself so that stations use an optimal (non-interfering) channel selection. We require a simple, robust and reliable solution that can be applied in arbitrary topologies and, since we allow that stations may belong to different administrative domains and/or lie beyond communication distance, our solution cannot rely upon explicit communication to agree a channel selection.

In this simple example the optimal channel selection itself is straightforward to see, but this need not be the case in more complex topologies. Indeed, the underlying channel selection problem considered here is equivalent to graph colouring and thus known to be NP-hard. To see this, define the interference graph by associating a node with each WLAN (e.g. with each BSS in an 802.11 network) and inserting an edge between nodes that interfere. For example, Figure 2 shows the interference graph corresponding to the wireless network in Figure 1. A colouring of the graph assigns colours to each node, and a proper colouring is an assignment of colours to each node such that no adjacent nodes share the same colour. A non-interfering channel allocation is thus equivalent to a proper colouring of the interference graph associated with a wireless network.

It is important to stress here that the use of circles to denote interference regions in Figure 1 is purely illustrative. In general, obstacles, channel variations and so on mean that interference regions can be highly complex. We therefore place no constraints whatsoever on the shape of the interference domains. Importantly, we note that since channel characteristics are dependent on the frequency channel used, we can expect that the shape of the interference regions will be channel dependent. That is, it is in general not sufficient to confine consideration to a single conflict graph as shown for example in Figure 2, but rather a different conflict graph may be associated with each available frequency channel. The occurrence of this type of situation is borne out by our experience in taking experimental measurements – see later. One consequence is that the channel allocation problem is not necessarily equivalent to the standard colouring task on a single graph, but rather may involve a more general multi-graph colouring task. While the decentralised algorithm studied here readily accommodates such conditions, the situation with other classes of algorithm is less clear.

While the example in Figure 1 considers a single-hop single-radio scenario, very similar considerations also apply in multi-hop multi-radio situations. For example, suppose that AP4 is the only access point with a wired backhaul link. Internet traffic to/from clients of access points AP1 and AP2 is passed via intermediate relay station AP3 to the backhaul station AP4 and thereby to the wired internet. Suppose that relay station AP3 is equipped with a 3-channel radio (leaving all other nodes with single channel radios as before). It then becomes possible for the AP1, AP2 and AP4 WLANs all to operate on different channels yet still communicate with the relay station AP3. An appropriate non-interfering channel allocation can then be used to avoid hidden/exposed terminal problems and simplify network administration. Note that provided the relay station has a radio with sufficient channels available then connectivity requirements are satisfied for any non-interfering channel allocation. It is also possible to extend the channel allocation task to include constraints on the number of channels available at each WLAN. The channel allocation task then becomes a list colouring problem, but we do not consider this extension in the present paper. Instead we note that when the relay station is under-provisioned, the available channels can always be partitioned into time-slots and these time-slots allocated to avoid interference. The allocation problem is logically equivalent to the pure channel allocation case. In both cases, the key challenge is to achieve a suitable channel (or time-slot) allocation without relying on message passing between interfering APs.

It is important to note that power control (adjustment of the transmission power employed by each station), while an extremely valuable tool, is only of limited help in examples such as these. In the first example of interfering single-hop WLANs, the WLANs might be located on different floors in an office block and operated by different businesses. The requirement to provide coverage for client stations places a lower limit of the transmit power that makes interference unavoidable. In the second example of a multi-hop relay, our freedom to adjust transmit power is constrained by the requirement to maintain connectivity between client stations and their APs, connectivity between AP1/AP2 and the relay station AP3, and connectivity between the relay and backhaul stations AP3 and AP4. Once again, there is thus a lower limit on the allowable power level which is large enough that significant interference cannot be avoided.

IV. Communication-Free Algorithm

In this section we introduce the class of decentralized algorithms studied in this paper. 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 decentralized algorithms for updating p.

Communication-Free Learning (CFL) Algorithm

1) Initialise $p = [1/c, 1/c, \ldots, 1/c]$
2) Toss a weighted coin to select a channel, with p_i the probability of selecting channel 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. Thus, we might, for example, use an aggregate measure derived from multiple packet transmissions or from direct measurement of the channel SINR.

3) On a successful choice of channel i, update p as

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

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

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

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

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.

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 in this paper is to answer both of these questions in the affirmative. More formally, let G denote the interference graph associated with a wireless network. A non-interfering channel allocation is one where each WLAN uses a different channel from all of its neighbours in G.

Theorem 1 *Suppose each node in a graph G operates the CFL algorithm. Assume that the channel allocation problem is feasible (i.e. the number of available channels c is greater than or equal to the chromatic number χ of G). Then the CFL algorithm converges, with probability one, to a non-interfering channel allocation.*

The proof of this result is given the the Appendix. Our 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. We have the following property.

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

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 more detail in later sections.

Before proceeding, however, we make the following brief observations.

(i) Multiple Radios. WLANs where stations are capable of making simultaneous use of multiple channels can be accommodated by running a copy of the CFL algorithm for each channel required.

(ii) Clock Synchronisation/Slotted time. Of key practical importance, we note that Theorem 1 carries over 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.

(iii) 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.

(iv) Multi-graph colouring. Although not discussed in detail here, it is straightforward to show that Theorem 1 can be generalised to encompass the case where, as discussed in the previous section, the interference conflict graph is channel dependent.

(v) 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.

(vi) No need for stopping/restarting. The CFL algorithm is a genuinely convergent one, 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 the network topology. This is of importance in practice as stopping/restarting in a distributed context seems problematic without message-passing.

V. Convergence Rate

A. Impact of Learning

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

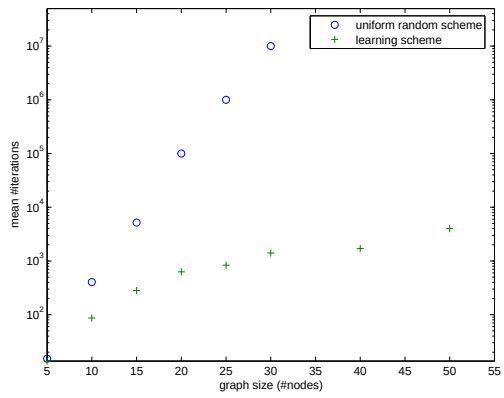


Fig. 3. Mean number of iterations to converge to an optimal channel allocation vs number of nodes in interference graph (random disk graphs with radius R=0.5, mean is taken over 1000 graphs, #channels $c = \chi$, $b = 0.1$)

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 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 $[1/c, 1/c, ..., 1/c]$"). Figure 3 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 modeled as a random disk graph; that is, APs are uniformly randomly located in a unit square and the WLANs associated with two APs interfere when the APs are located within a radius R of each other. A "failure" or "collision" occurs when neighbouring nodes select the same channel at a given iteration of the channel allocation algorithm, and a "success" when a node selects a different channel from all of its neighbours. For each interference graph the number of channels is set equal to the chromatic number χ (calculated using the DSATUR algorithm); 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. Note the log scale used in this graph, and in many others in this paper.

B. 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. Figure 4 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.

VI. Time-varying Topologies

A channel allocation scheme must 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 on WLANs, from mobility, changes in environmental conditions, and so on.

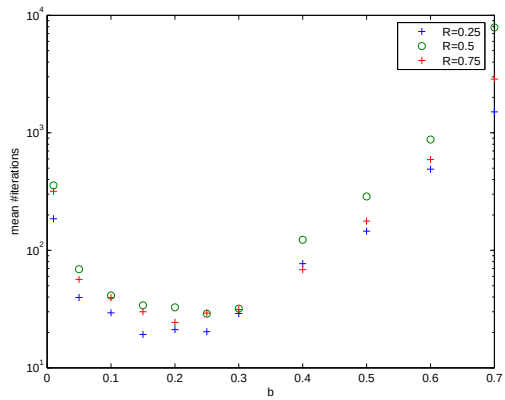


Fig. 4. Mean number of iterations to converge to an optimal channel allocation vs learning parameter b (random disk graphs, random numbers of nodes, #channels $c = 1.25\chi$, each point is mean taken over 1000 graphs)

A. 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 so that each node in the interference graph is connected to every other node) 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 our distributed channel allocation 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 yield 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. Figure 5 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. Figure 5 shows results as b is varied and it can be seen that choosing values of b in the range 0.1-0.2 yield the fastest convergence, which is in good agreement with the previous results in Figure 4.

Also shown in Figure 5 are the predictions corresponding to the following simple 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 $E[S_{N+1}(k)] = 1 - E[F_{N+1}(k)]$.

It can be seen from Figure 5 that the predictions of this analysis are remarkably accurate for the case when $b = 0.1$,

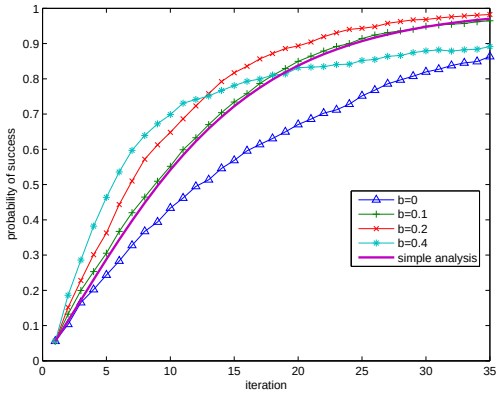


Fig. 5. Probability of success (convergence) vs iteration and choice of parameter b following addition of a new WLAN. (10 node complete interference graph, simulation probability is the mean from 1000 runs).

indicating that the channel allocations of the WLANs in the original network possess sufficient "inertia" that they do indeed effectively remain unchanged (this is also confirmed by direct measurement 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 simple 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 WLANs effectively retain their original channel allocations) is violated.

While the foregoing analysis is for networks with complete interference graphs, the arguments carry over directly to general situations. For example, Figure 6 shows simulation results for a network with a random disk interference graph together with the corresponding analytic predictions. In this example we take a network where WLANs are located randomly in the plane and WLANs interfere if they are within a distance R of one another. In this case the interference graph is a random disk graph G. We then randomly[3] 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 it may be that there exists more than one channel that the new WLAN can select that will yield 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 Figure 6. The foregoing analysis can be applied provided there exists at least one local solution, and it can be seen from Figure 6 that in such situations it yields remarkably accurate predictions.

[3]The network is located in the plane, making it straightforward to add a new WLAN. Specifically, we select uniformly random x and y co-ordinates for the new WLAN and then determine its neighbours using the interference radius R.

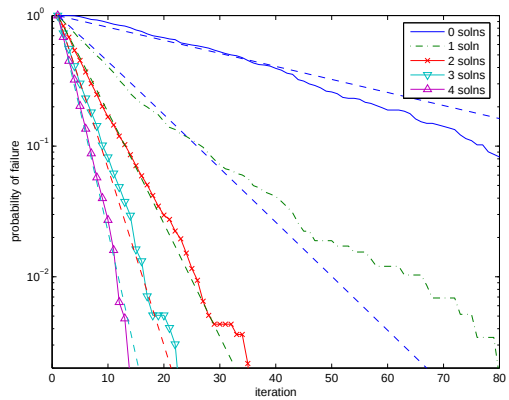


Fig. 6. Probability of failure (1- prob of success) vs iteration following addition of a new WLAN. Dashed lines are analytic predictions. "Solutions" refer to the number of possible channels that the new WLAN may select to achieve a proper channel allocation without disturbing the allocations of the original WLANs. (20 node random disk interference graph with R=0.5, #channels c=12 (1.25χ), $b = 0.1$.)

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 Figure 6. 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 NN. We know that these WLANs make use of all available channels. Our measurements on many hundreds of thousands 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 NN 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. This possibility is neglected in our model because simulations show it is a rare occurrence. Hence our model predicts that independently at each timestep, the system will reconverge approximately with probability at least $b/(c-1)$. The accuracy of this approximate analysis is illustrated in Figure 6.

The foregoing analysis can be used to make quantitative predictions of convergence rate, provided we have knowledge of the number of local solutions in a neighbourhood of interest

(or perhaps the probability distribution of these). We argue, however, that its real importance lies in the qualitative insight that the addition of a new WLAN induces only minimal changes in the original network channel allocation. This property is key to achieving high performance in time-varying environments.

B. Example

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

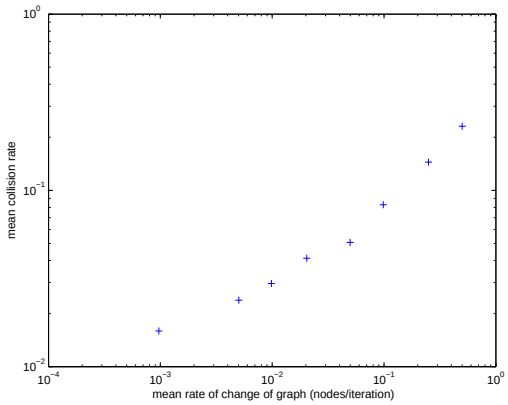


Fig. 7. Overhead (failure rate) induced by topology changes vs rate of interference graph change. (Mean number of nodes 20, nodes are added/deleted at exponentially distributed intervals with mean rate given by the x-axis to random disk graph with radius R=0.25, 5 channels are available, results are average over 100 graphs, $b = 0.1$).

Figure 7 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. to the time required to sense the channel state and so might be on the order of a few tens or hundreds of milliseconds depending upon the metric used.

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 I
TESTBED SUMMARY

VII. EXPERIMENTAL RESULTS

.

The CFL algorithm requires no special hardware support and, in addition to avoiding message passing, does not require clock/slot synchronisation between interfering WLANs. We have implemented a prototype version of the CFL algorithm on a standard Linux platform and present results demonstrating performance under real interference conditions. Before proceeding, however, we first describe our testbed setup.

A. Testbed Setup

The testbed consists of 10 PC-based embedded Linux boxes based on the Soekris net4801, 5 boxes configured as APs in infrastructure mode and 5 as client stations, see Table I. 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. These machines are setup as five WLANs (denoted WLAN A - WLAN E) located in a university office space as shown in Figure 8. All systems are equipped with an Atheros 802.11a/b/g mini-PCI card with an external antenna. The system hardware configuration is summarised in Table I. All nodes use a Linux 2.6.16.20 kernel and the MADWiFi wireless driver. All of the systems are also equipped with a 100Mbps wired Ethernet port, which is used for control of the testbed from a PC. Specific vendor features on the wireless card, such as turbo mode, are disabled. Channel scanning is also disabled as we use the CFL algorithm for channel selection. 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. With this PHY rate and using 1500 byte packets, the achieved throughput in a single isolated WLAN is measured to be approximately 13 Mbps.

B. 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 user-space 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.

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

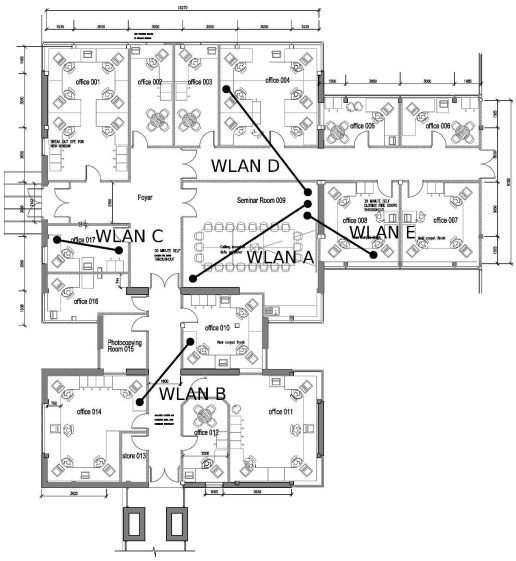


Fig. 8. Plan showing wireless node locations.

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 VII-D for details. We note that the use of RTS/CTS creates an overhead that can reduce network capacity. While RTS/CTS is sufficient for proof of concept, we are therefore also investigating alternative measures [16] and will report experimental results on these at a later date.

C. Nature of Interference Environment

Prior to evaluating the performance of the CFL channel allocation algorithm, we attempted to characterise the interference environment in our testbed. This is, of course, of interest in its own right as interference characteristics in the "wild" remain relatively poorly characterised. Moreover, such measurements provides insight into the sort of performance requirements that must be met by any channel allocation algorithm if it is to be practically applicable.

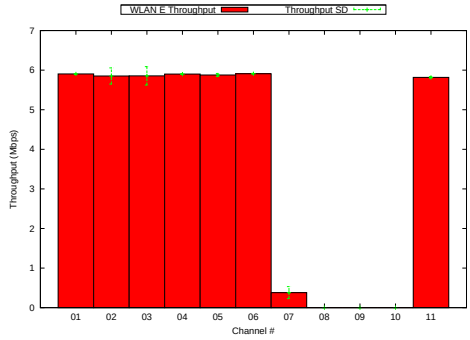


Fig. 9. Baseline throughput for WLAN C versus channel number in 2.4GHz band (no other WLANs active).

1) 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 analyzer 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, Figure 9 shows measured throughput versus channel number in the 802.11b band for WLAN C – none of the other WLANs 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 is attributed to bluetooth devices operating in a lab (marked office 001 in Figure 8) 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.

2) Time-varying Channel Quality: Our measurements indicate that the channel quality can be strongly time-varying. For example, Figure 10 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. The time-varying nature of the channel quality is evident – e.g. compare channels 48 and 153.

Also marked on Figure 10 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 Figure 11 which shows an example time history of measured channel quality over a period of approximately 60 minutes. It can be seen, for example, that the error rate rises to around 15% for a period of about 10 minutes early in this experiment, then falls to around 3% after approximately 30 minutes.

In this particular example, measurements using a spectrum analyzer 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 Figure 10 that certain

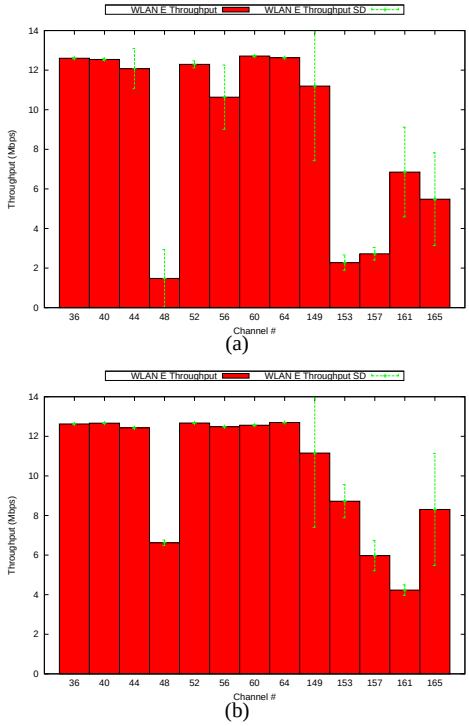


Fig. 10. Measured throughput with a single WLAN active (i.e. no interfering WLANs). Measurements are shown for WLAN E over the range of 802.11a channels. The measurements in the upper and lower plots are taken about 1 hour apart. Observe the substantial variation in throughput both with channel number and time.

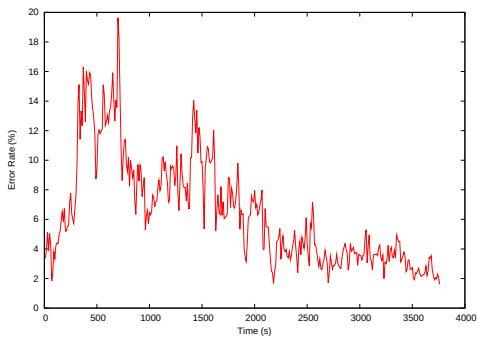


Fig. 11. Example of time-varying channel quality.

channels are consistently of good quality, e.g. channels 36-44 and 60-64 and so a channel allocation algorithm should seek to allocate on these channels.

3) Channel Dependent Interference: Our measurements indicate that the level of interference between WLANs can be strongly channel dependent. For example, Figure 12 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 perhaps unsurprising as we can 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-

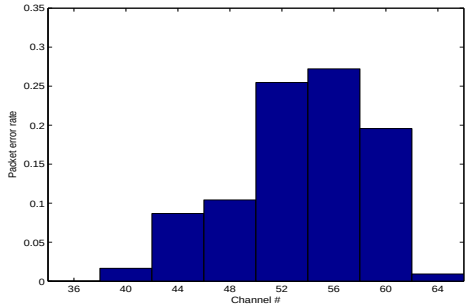


Fig. 12. Measured interference induced error rate versus channel number in 5GHz band. Here WLANs B and C both transmit CBR traffic on the same channel. Plot shows measured packet error rate at WLAN C as the channel number used for transmission is varied (with WLANs B and C always sharing the same channel)..

graph colouring task.

D. Communication-Free Channel Allocation Algorithm Experimental Results

We now study the practical performance of the implemented CFL algorithm for decentralized channel allocation.

1) 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. To create a relatively demanding channel allocation task, the channel allocation algorithm was restricted (via scripting) to the use of four 802.11a channels. Initially, all WLANs are started on the same channel and a copy of the CFL algorithm is run 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 as discussed previously.

Figure 13 shows traces of the channel selection time histories for each of 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 network throughput from 50-60 seconds is approximately 51 Mbps compared with an aggregate throughput of 11.31 Mbps when the WLANs all use the same channel. That is, we obtain approximately a factor of four increase in network capacity through appropriate channel selection. Table II gives detailed measurements. Similar results were obtained across many runs, confirming that this level of capacity improvement is consistently achieved.

2) Convergence Rate: It can be seen in Figure 13 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 (this is discussed in more detail below) yielding an overall convergence time of 200s.

We note firstly that during this convergence period the network continues to achieve a significant level of throughput.

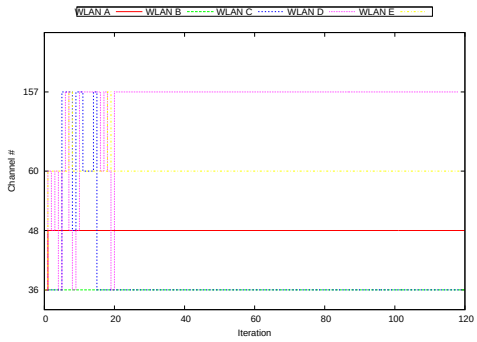


Fig. 13. WLAN channel time histories. Five WLANs, four available channels. Note that in this example the network settles on only three channels.

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 II

MEASURED THROUGHPUTS, 5 WLANS AND 4 AVAILABLE CHANNELS.

This is illustrated n Figure 14 , 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.

Secondly, the simulation analysis in [15] 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.

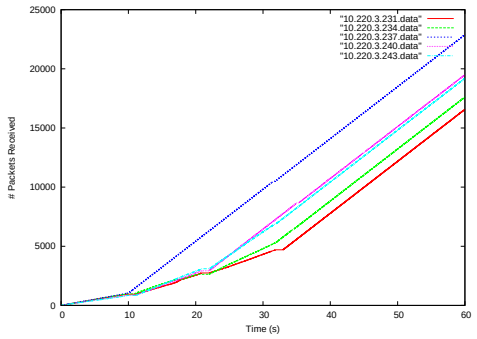


Fig. 14. Cumulative packets received versus time for each of the five WLANs. Five WLANs, four available channels.

3) Estimating Channel Quality: Channel quality is estimated from the average frame error rate measured over a 10 second interval. The CSMA/CA MAC scheduling process in 802.11 means that frame errors due to interfering transmissions are random in nature and the choice of measurement interval used therefore affects the accuracy of measurements of frame error rate. For example, use of a very short measurement interval tends to yield measurements with high variability, and conversely a long measurement interval yields smoother measurements.

This behaviour is illustrated in Figure 15 – here, the same packet trace is processed using different measurement intervals to allow direct comparison of the resulting time histories of estimated frame error rate. While use of a long measurement interval yields smoother channel estimates, if the channel quality is in fact time-varying (as often seems to be the case, see below) then use of a long interval will introduce measurement errors. Further, use of a long interval means that it takes longer for the network to sense channel conditions and adapt to changes.

We therefore want to choose the shortest measurement interval that yields consistent channel quality estimates. Figure 16 plots the standard deviation of the frame error rate measurements as a function of the measurement interval used. In our tests, a 10 second interval corresponds to between 2000 and 10000 packet transmissions (depending on the level of interference), yielding a standard deviation of less than 1% which we found to be a good compromise between accuracy and responsiveness.

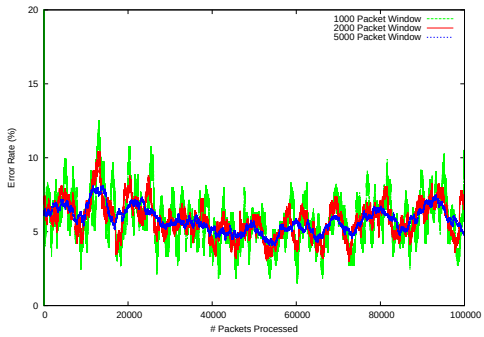


Fig. 15. Illustrating impact of choice of measurement interval on accuracy of frame error rate measurements. Results shown are for the same packet trace but with measurement intervals of 1000, 2000 and 5000 packets. Packet trace covers 200,000 transmission attempts.

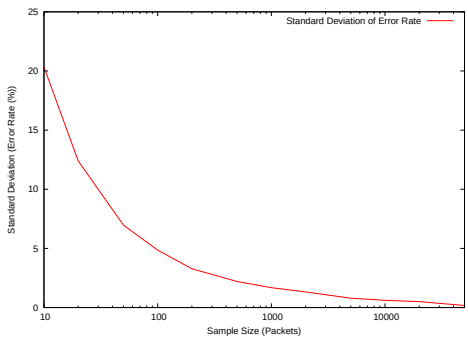


Fig. 16. Standard deviation of measured frame error rate vs length of measurement interval. Results shown are for the same packet trace but with measurement intervals varied. Packet trace covers 200,000 transmission attempts.

4) Controlling local channel reuse: Observe in Table II that WLAN B and WLAN D settle on the same channel. It can be seen from Figure 8 that these WLANs are located near to each other and on closer inspection of packet traces we find that the nodes in these WLANs are visible to each other

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 III

MEASURED THROUGHPUTS, 5 WLANS AND 4 AVAILABLE CHANNELS. DETECTION OF FOREIGN BEACONS USED TO FORCE CO-LOCATED WLANS ONTO DISTINCT CHANNELS.

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 IV

MEASURED THROUGHPUTS. FOUR WLANS ACTIVE INITIALLY, WITH FIFTH WLAN BEGINNING TRANSMISSIONS ARE TIME 60

(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 seems desirable in dense deployments as it increases the level of channel reuse. That is, 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, however, 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 otherwise unchanged.

Table II 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.

5) Impact of external and channel dependent interference: Our measurements of the testbed interference environment highlight 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 Figure 12, we recorded statistics on the channels selected by these WLANs over a series of 10 tests. In line with Figure 12 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 it can be seen from Figure 10 that the quality of certain channels can be strongly time-varying. We can also observe in Figure 10 that certain channels are consistently of good quality, e.g. channels 36-44 and 60-64. Our measurements confirm that the CFL algorithm automatically avoids the low quality channels and settles on the good quality channels.

Our experience suggests that this adaptive behaviour of the CFL algorithm is a key feature that any channel allocation algorithm must provide if it is to be practically effective, although the issue of channel dependent noise/propagation and the strong spatial variation in channel quality does not seem to have been widely considered in the WLAN channel allocation literature.

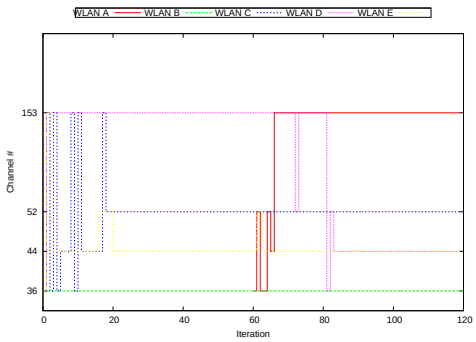


Fig. 17. Example of a new WLAN becoming active. Four WLANs active initially, with fifth WLAN beginning transmissions are time 60.

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 and therefore generates essentially no interference. Importantly, when a WLAN that has been inactive becomes active, we require to allocate a channel 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, such as a previously dormant WLAN becoming active, 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 Figure 17. 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 IV gives the corresponding WLAN throughputs.

7) Spatial Reuse: 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. Figure 18(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. Figure 18(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 Figure 8 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.

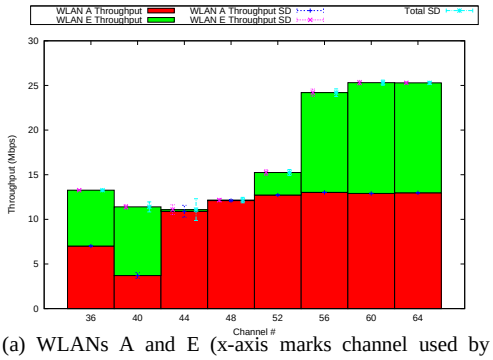
It can be seen that when WLAN A is located on channel 56 and above, the aggregate network throughput is 26Mbps which is approximately the maximum combined capacity that can be achieved by two independent WLANs for the 802.11a settings used here. Observe also that both WLANs achieve throughputs i.e. network 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 Figure 18(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.

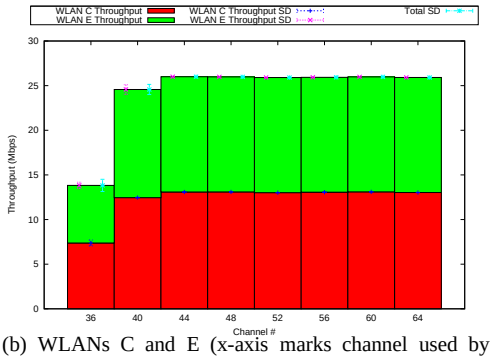
To investigate this behaviour further, for comparison Figure 19 shows the corresponding measurements when the 802.11b 2.4GHz channel is used. Note that in these 802.11b measurements there exists significant background noise on channels 7-10 for WLAN C – see earlier discussion. The impact of this noise can be seen in Figure 19(a), 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.

Focussing 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 Figure 19(b) 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 always seek an orthogonal channel allocation, automatically avoiding channel configurations that create interference.



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

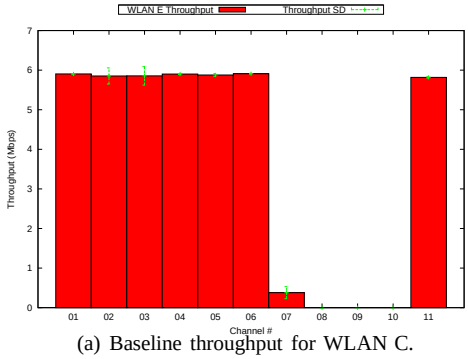


(b) WLANs C and E (x-axis marks channel used by WLAN C).

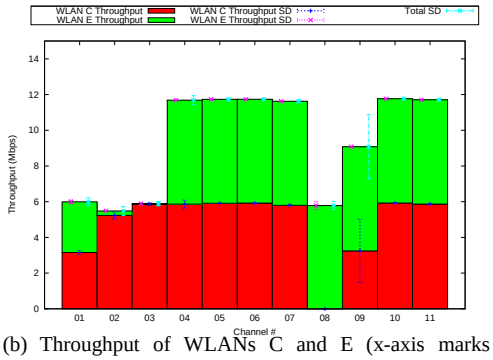
Fig. 18. Measuring potential for channel reuse. Using the 802.11a 5GHz band, WLAN E is held fixed on channel 36 while the channel used by second WLAN is varied. Measurements are shown for WLANs A and E and for WLANs C and E. Height of histogram indicates aggregate throughput of both active WLANs. Light 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.

VIII. Scope of the Paper

In this paper we focus on the channel allocation task itself, rather than joint channel allocation, routing (and the related issue of association control) and power control. Certainly these network aspects are coupled, but also individually challenging and joint optimisation is left as future work. In this paper we also confine consideration to situations where sufficient



(a) Baseline throughput for WLAN C.



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

Fig. 19. Channel reuse with 802.11b. WLAN E is held fixed on channel 1 while the channel used by WLAN C is varied

channels are available to allow an non-interfering allocation to exist. This is not a fundamental constraint as frequency channels can always be subdivided into (frequency, time slot) pairs to create sufficient logical "channels" and the CFL algorithm can then be applied directly. Time slotting might be implemented for example via the CFB functionality in 802.11. That said, the case of insufficient channels remains of interest. The CFL algorithm can be extended to encompass this case, but this introduces issues of fairness and efficiency that are outwith the scope (and space constraints) of the present paper.

IX. Conclusions

In this paper we introduce a new class of decentralized channel allocation algorithms that are simple and robust, provably correct for arbitrary interference graphs, require no communication between interfering WLANs and remarkably fast under a wide range of network conditions and topologies. These algorithms are suited to implementation on standard equipment and we investigate the implementation of our channel allocation algorithm in an experimental testbed located in an office environment. Initially, we use measurements to investigate the interference environment present in the testbed. We observe that the environment includes (i) external non-cooperating interference sources in the 2.4GHz band, (ii) strongly time-varying quality on some channels, (iii) channel dependent levels of interference between transmissions on different WLANs. These effects are all of importance when developing a practically useful channel allocation algorithm. We then implement the channel algorithm on standard 802.11 hardware. We demonstrate, via extensive measurements, that this algorithm does indeed offer the potential for effective channel allocation in realistic environments. In particular, this includes environments with complex, spatially varying noise and propagation behaviour and with time-varying load.

References

[1] A. Akella, G. Judd, P. Steenkiste, and S. Seshan. "Self management in chaotic wireless deployments". In *MobiCom*, 2005

[2] L. Tassiulas, A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks", *IEEE Trans Automatic Control*, 37 (12), 1992.

[3] H. Luo, P. Medvedev, J. Cheng, S. Lu, "A self coordinating approach to distributed fair queuing in ad hoc wireless networks", *Proc. of IEEE INFOCOM '01*, 2001.

[4] A. Raniwala, T. Chiueh. "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network". In *Proc IEEE International Conference on Computer Communications*, 2005.

[5] A.Mishra, V.Shrivastava, D.Agarwal, S.Banerjee, S.Ganguly, Distributed Channel Management in Uncoordinated Wireless Environments. In *MobiCom*, 2006.

[6] A.Mishra, V.Brik, S.Banerjee, A.Srinivasan, W.Arbaugh, "A Client-Driven Approach for Channel Management". *Proc. IEEE INFOCOM*, 2006

[7] B.J. Leung, K.K. Kim, "Frequency assignment for IEEE 802.11 wireless networks". *Proc. 58th IEEE Vehicular Technology Conference*, 2003.

[8] B. Kauffmann, F. Baccelli, A. Chaintreau, K. Papagiannaki, C. Diot, "Self Organization of Interfering 802.11 Wireless Access Networks,", *INRIA Technical Report*, August 2005.

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

[10] A. Subramanian, H. Gupta and S. R. Das, "Minimum interference channel assignment in multi-radio wireless mesh networks" *Proc. Mobicom*, 2006.

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

[12] 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". *Proc. IEEE Globecom*, 2006

[13] K. Ramachandran, E. Belding, K. Almeroth, M. Buddhikot, "Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks". *Proc. IEEE INFOCOM*, 2006.

[14] D. Aguayo, J. Bicket, S. Biswa, G. Judd, R. Morris. "Link-level measurements from an 802.11b mesh network". *Proc. ACM SIGCOMM*, Boston, 2004.

[15] Leith,D.J., Clifford, P., "A Self-Managed Distributed Channel Selection Algorithm for WLANs". *Proc. ACM/IEEE RAWNET*, Boston, 2006.

[16] Malone, D.W., Clifford,P., Leith,D.J., 2006, MAC Layer Channel Quality Measurement in 802.11. *IEEE Communcations Letters*, to appear.

Appendix: Proof of Theorem 1

We show that in a determined finite amount of steps the system has some minimum positive probability of convergence. We show that starting from any configuration the system can reach some standard state after two steps. From this standard state we show that the system can then potentially reach a state where every node experiences a failure simultaneously, allowing convergence without issues of dependence between nodes. Hence the network always has positive probability of global success and will almost surely converge.

In the sequel we refer to two nodes choosing the same channel as a "collision". We say that the state $\mathbb{S}$ consists of all possible non globally converged configurations with all colour selection probabilities strictly greater than $\frac{b}{c-1}$. Denote the maximum node degree by md and the diameter of the graph (length of the longest shortest path between two nodes) by D.

The system may avoid state $\mathbb{S}$ by some node undergoing repeated same channel collisions. We show in Lemma 1 that

if the system has reached a configuration with some colour selection probabilities very small there is a positive lower bounded probability that it will return in two steps to our standard state $\mathbb{S}$. Thus (with some probability) the initial colour selection probabilities will have no effect on the probability of a given evolution.

Lemma 1. From any configuration of the system, if after two steps the system has not converged, the system is in state $\mathbb{S}$ with some probability $pr_5 > 0$.

Proof of Lemma 1. After any step T_0 there was either global success (and convergence) or at least two nodes suffered a collision. Starting at time T_0 we allow the system to evolve for 2 more steps and lower bound the probability of the system not being in state $\mathbb{S}$. We ignore nodes who succeed and then collide as their colour selection probabilities are clearly at least $b/(c-1)$. Consider any just collided node; after the first collision by choosing colour i_1, it has probability $pr_1 > b/(c-1)$ of choosing some specific other colour i_2 and probability $pr_2 > (c-1)pr_1$ of choosing any colour other than i_1. So the probability of two repeated collisions on the same colour at a specific node is $pr_3 < 1 - pr_2$. In the whole system the probability of some node having two consecutive same colour collisions is $pr_4 < npr_3 - \binom{n}{2}(pr_3)^2 + \cdots < 1$. Hence with some probability $pr_5 > 1 - pr_4 > 0$ the system has no node with consecutive same colour collisions. Thus after these two steps with probability pr_5 all colour selection probabilities of nodes which have just collided are strictly greater than $b/(c-1)$ and the system is in state $\mathbb{S}$. ■

Lemma 2. Suppose that the system is in state $\mathbb{S}$. Assume without loss of generality that the graph is connected. There exists a specific evolution $\mathbb{E}$ of the system which results in all nodes transmitting succesfully.

Proof of Lemma 2. In state $\mathbb{S}$ by definition we have not converged and (at least) two nodes k_1 and k_2, say, have just experienced a collision. By way of notational convenience we say these two nodes were *visited* at step 2. Suppose now that k_1 collides with its first non visited neighbour k_3 (if any) at step 3. Suppose also that k_2 collides with its first non visited neighbour (if any, potentially k_3 also) at step 3 also. We say that such nodes are *visited* at step 3. Inductively suppose now that a node once visited collides with all its nonvisited neighbours in consecutive steps. This is possible because a visited node having just collided can potentially choose any colour. Note that a node being visited simultaneously (along two different equal length paths from k_1 and k_2 say) is also possible.

Suppose that once a node has collided with all its nonvisited neighbours it then repeatedly chooses colour 1 until step $T_1 = T_0 + 3 + md \times D$. We note that as a node k_4 is colliding with its nonvisited neighbours some of them may become visited from other nodes before they collide with k_4; we suppose then that k_4 does not visit such nodes. Note also that $md \times D$ upper bounds the time needed for this visiting procedure to visit every node.

By connectedness, at time $T_1 - 1$ it is possible for every node to have been visited and to be choosing colour 1. Hence every node is colliding. Since every node is colliding, every node can choose every colour, so we can finally suppose that at step T_1 every node selects a colour so that no collisions occur. ■

Lemma 3. There is a strictly positive lower bound pr_8 on the probability of the evolution $\mathbb{E}$ occurring from any configuration in state $\mathbb{S}$.

Proof of Lemma 3. Given the initial colour selection probabilities and which nodes collided initially, the evolution $\mathbb{E}$ is well defined and has some positive (computable) probability pr_6 of occurring since the system is finite.

By assumption the system begins in state $\mathbb{S}$ and so the initial colour selection probabilites of just collided nodes are lower bounded; therefore there is some probability $pr_7 > 0$ such that $pr_6 > pr_7$ irrespective of the initial selection probabilities. The subset of nodes which collided initially is one of finitely many possibilities and so again there is some probability $pr_8 > 0$ such that $pr_7 > pr_8$ irrespective of which nodes collided initially. ■

Proof of Theorem 1. Defining $pr_9 = pr_8 pr_5$ gives the probability that the system is in state $\mathbb{S}$ after the first two steps and then follows evolution $\mathbb{E}$. So every $2 + md \times D$ steps the system will converge with probability at least pr_9. Hence after $j(2 + md \times D)$ steps we have converged with probability at least $1 - (1 - pr_9)^j$ which converges to 1 as $j \to \infty$. ■

PLACE PHOTO HERE

Doug Leith graduated from the University of Glasgow in 1986 and was awarded his PhD, also from the University of Glasgow, in 1989. In 2001, Prof. Leith moved to the National University of Ireland, Maynooth to assume the position of SFI Principal Investigator and to establish the Hamilton Institute (www.hamilton.ie) of which he is Director. His current research interests include the analysis and design of network congestion control and distributed resource allocation in wireless networks.

PLACE PHOTO HERE

Peter Clifford

PLACE PHOTO HERE

David Malone