Channel Dependent Interference and Decentralized Colouring

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

***Abstract*— We consider channel allocation to mitigate interference between wireless LANs. The channel allocation task is often formulated in the literature as finding a proper colouring of a single graph. In this paper we demonstrate that this formulation may be unrealistic. Specifically, we show that the interference between WLANs can be channel dependent in which case a different conflict graph is associated with each channel. Channel allocation then corresponds to a multi-graph colouring problem. This potentially has profound implications as the behaviour of many proposed colouring-based algorithms for channel allocation is unclear in a multi-graph context. We are, however, able to show that a recently proposed decentralized colouring algorithm does generalise to the multi-graph setting.**

I. Introduction

In this paper we consider how a group of access-points/base-stations [1] can configure their channel choice so as to minimise interference between one another. This problem has recently been the subject of an upsurge of interest in the WLAN literature, e.g. see [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14].

The channel allocation task is often formulated in the literature as finding a proper colouring of a single graph. That is, a conflict graph is constructed by associating a graph vertex with each WLAN and inserting edges between WLANs that interfere. A non-interfering channel allocation then corresponds to a proper colouring of this conflict graph. In this paper we demonstrate that this formulation may be unrealistic. Specifically, we show that the interference between WLANs can be channel dependent in which case a different conflict graph is associated with each channel. Channel allocation then corresponds to a *multi-graph* colouring problem. This potentially has profound implications as the behaviour of many proposed colouring-based algorithms for channel allocation is unclear in a multi-graph context.

Our second main contribution in this paper is to establish that a recently proposed decentralized colouring algorithm does indeed generalise to the multi-graph setting. We also present a new, extended version of this algorithm suited to a wide range of multi-radio architectures.

[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.

II. Channel allocation and graph colouring

The channel allocation task is usually formulated as a standard graph colouring problem. For example, Figure 1 shows four interfering 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 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.

The underlying channel selection problem in this example is equivalent to graph colouring. 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

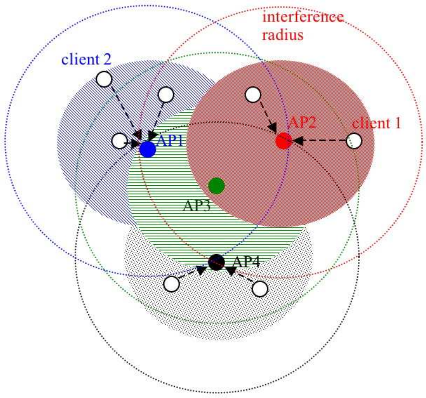

Fig. 1. Example of interfering 802.11 WLANs. Dashed circles indicate interference radius, shaded circles indicate communication radius.

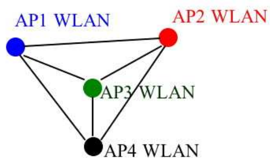

Fig. 2. Interference graph of Figure 1.

node such that no adjacent nodes share the same colour. A non-interfering channel allocation is then equivalent to a proper colouring of the interference graph associated with a wireless network.

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.

III. Channel dependent interference

It is important to stress here that the use of circles to denote interference regions in Figure 1 is an idealisation. In general, obstacles, channel variations and so on mean that interference regions can be highly complex. Importantly, we note that since channel characteristics are dependent on the frequency used, we can expect that the shape of the interference regions will be *channel dependent*.

To investigate this question, we took measurements on an experimental testbed. 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. 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 3. All systems are equipped with an Atheros 802.11a/b/g mini-PCI card with an external antenna. 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 and channel scanning, are disabled.

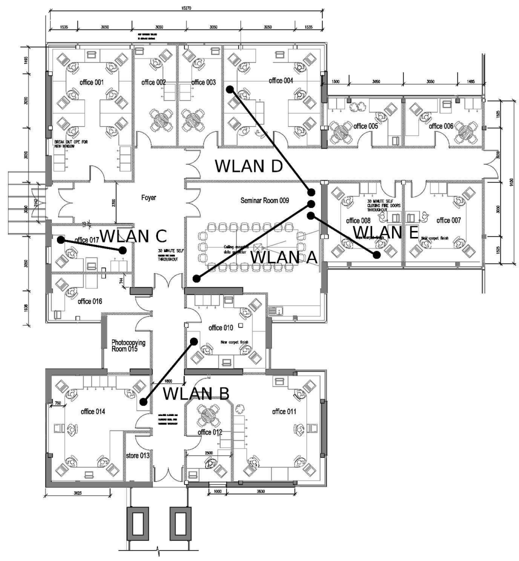

Fig. 3. Plan showing wireless node locations.

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 which is attributed to bluetooth devices .

Focussing on the 5GHz band, our measurements indicate that the level of interference between WLANs can

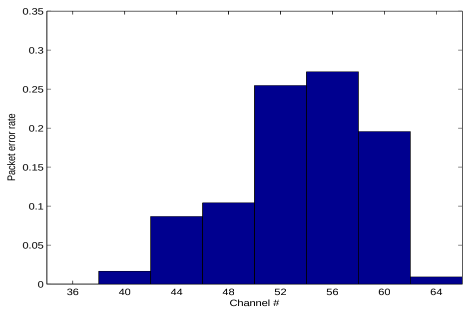

Fig. 4. Measured interference induced error rate versus channel number in 5GHz band. Here WLANs B and C both transmit CBR traffic on the same channel. Plot shows measured packet error rate at WLAN C as the channel number used for transmission is varied (with WLANs B and C always sharing the same channel).

be strongly channel dependent. For example, Figure 4 shows the measured interference level between WLANs B and C as the channel number is varied (with WLANs B and C always sharing the same channel).

This behaviour is perhaps unsurprising as we can expect path propagation characteristics to be frequency dependent. Nevertheless, it has profound implications for channel allocation algorithms. In particular, 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. An immediate 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.

IV. Implications for Channel Allocation Algorithms

The multi-graph colouring problem is fundamentally different from the single graph colouring problem. For example, the chromatic number (minimum number of colours for a proper colouring) of the multi-graph problem is only weakly related to the constituent individual graphs. We illustrate this by example.

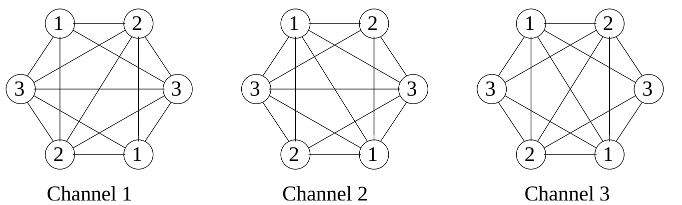

Fig. 5. Multi-graph example 1. Individual channel conflict graphs shown with proper colouring requiring only 3 colours.

Figure 5 shows the conflict graphs associated with channels 1,2 and 3 in a network of 6 interfering WLANs and also shows a successful channel allocation using 3 channels. Although for each channel there is only one pair of nodes which do not interfere, the arrangement is such that only three channels are necessary to avoid interference, rather than the six which would be required if every node interfered with every other on every channel. This example demonstrates that the problem of multi-graph colouring is dramatically different to normal graph colouring.

To our knowledge, no analytic results are available on the performance of colouring algorithms on multi-graphs. Existing convergence proofs for distributed algorithms such as those in [9], [11], [14] relate to colouring of a single graph. Centralised channel allocation algorithms based on single graph colouring may exhibit unexpected behaviour in a multi-graph context.

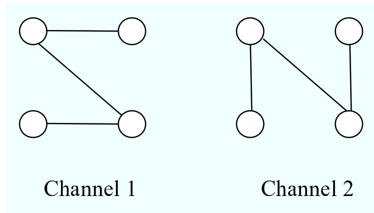

Fig. 6. Multi-graph example 2. Individual channel conflict graphs have chromatic number 2 yet multi-graph proper colouring requires at least 3 colours.

For example, standard centralised algorithms for colouring bipartite graphs can fail to yield a non-interfering channel allocation, and indeed may fail to converge. To see this consider the example in Figure 6 which shows the conflict graphs associated with channel 1 and channel 2 in a network of 4 interfering WLANs. The individual channel conflict graphs are bipartite and so each can be coloured using at most 2 colours. However, the multi-graph colouring problem requires at least 3 colours for a proper solution.

Channel dependent interference also has direct implications for frequency hopping approaches to channel allocation such as that in [6] and elsewhere. The performance of heuristic algorithms is unclear.

V. Communication-Free Colouring Algorithm

In this section we briefly review the decentralized channel allocation algorithm introduced in [14], before presenting our main analytic result.

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 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 — see [15], [16].
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.

A notable feature of this CFL algorithm, in addition to its evident simplicity, is that it does not require any message passing (either direct or by sniffing packet headers) between interfering stations. We note that this communication free property is also shared by the algorithm recently proposed in the unpublished work of Kaufmann et al [9], although the latter makes use of a simulated annealing approach whereas the CFL algorithm is based on a learning strategy.

It is important to emphasise here that while a great many distributed schemes have been proposed in the literature (e.g. see [1], [2], [5], [4], [8], [6], [7], [10], [11], [12], [13] and references therein), almost all are distributed in the sense that they require only local communication between access points that directly interfere with one another albeit perhaps implemented via sniffing of packets rather than by dedicated transmissions, e.g. see [6]. 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 in the literature 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 [14] we establish that the CFL algorithm converges with probability one to a non-interfering channel allocation provided one exists and in [17] a bound on the convergence rate is established. These results are derived, however, by formulating the channel allocation task as being equivalent to finding a proper colouring of a single graph.

Recently, we have also carried out experimental testing to evaluate the CFL algorithm for channel allocation, see [16]. It is this testing which highlighted the potential multi-graph nature of the channel allocation task in real networks and which motivates the main analytic result in the present paper. Namely, in this paper we establish that the CFL algorithm is not only convergent for colouring single graphs but also possesses the much stronger property that the algorithm converges with probability one to a non-interfering channel allocation when interference may be channel dependent i.e. that the CFL algorithm solves the multi-graph colouring problem, provided of course that a solution exists.

VI. Main Result

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 network 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.*

The proof of this result is given the next section. Our proof also 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. Obtaining a tight bound on γ is the subject of current work – following a similar approach to that used in [17] to obtain an analytic bound for a single (channel independent) conflict graph. In the present paper, we employ numerical simulations to gain insight into the convergence rate behaviour for multi-graph problems. We note that [14] presents extensive numerical simulations illustrating the rapid convergence, on average, of the CFL algorithm for colouring of single graphs and so we focus here on multi-graph colouring performance.

VII. Convergence Rate

Figure 7 illustrates the exponential convergence nature of the CFL algorithm. In this example network interference on each channel 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 on channel i when the APs are located within a radius R_i of each other. The interference radius R_i for channel i is randomly selected in the interval [0.25, 0.75]. In the channel independent case the chromatic number of the graph is defined to be the minimum number of channels in any successful channel allocation. In the channel dependent case we must first assume that the interference graphs are specified. Now we can define the chromatic number to be the smallest number of channels χ such that there is a successful channel allocation using the first χ channels[2]. In Figure 7 the

[2]in general the chromatic number is defined as the smallest number of channels χ such there is a successful channel allocation using any χ channels. However this χ is exceptionally hard to compute

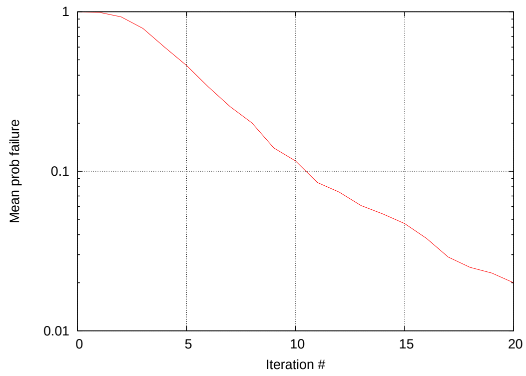

Fig. 7. Mean failure probability vs number of iterations (randomly selected 15 node disk graph, number of channels $c = \chi$, mean is over 1000 runs, $b = 0.1$)

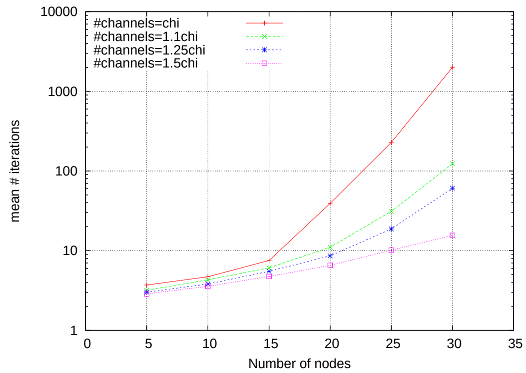

Fig. 8. Mean number of iterations to converge to an optimal channel allocation vs number of nodes in interference graph and channel provisioning relative to chromatic number χ

number of channels provided is always χ, which is found by numerical search.

We investigate the mean convergence time in more detail by varying the number of WLANs, see Figure 8. Also in this figure we show the impact of the number of available channels. We plot the mean convergence time versus the level of channel over-provisioning over and above χ (which is again found numerically). As expected, we can see that the convergence time decreases as the level of over-provisioning is increased. However, what is perhaps most significant is that it can also be seen that the impact of even a relatively small amount of over-provisioning can be very considerable. For example, 25% additional channels over and 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.

There is a related algorithm which assigns a constant uniform probability to each channel after a failure and remains settled after a success. Simulations (not presented here) have verified that this non learning algorithm performs significantly worse than the learning algorithm.

We can gain some analytic insight into the very great impact of even small amounts of over-provisioning as follows. 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. Suppose we have $\chi + \delta$ channels. We must assume the worst case, i.e. that the δ additional channels interfere with everybody. We can thus choose δ nodes in $\binom{n}{\delta}$ ways and give each chosen node a new channel in $\delta!$ ways. Applying Stirling's approximation and assuming δ is small relative to n gives the expected exponential lower bound

$$T(\delta) \geq \left(\frac{n}{e}\right)^{\delta}.$$

PROOF OF THEOREM 1

We will 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 so 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 the set of all possible configurations where (i) the channel selections of at least two nodes interfere and (ii) at all colliding nodes the selection probability for every channel is bounded away from zero (in fact, we will consider the case where they are strictly greater than $\frac{b(1-b)}{c-1}$). We define the master graph: an edge is in the master graph if it is in any of the individual channel graphs $G(i), i \in [1, 2, .., c]$. Denote the maximum node degree of the master graph by md and the diameter of the master graph (length of the longest shortest path between two nodes) by D.

Consider a colliding node. Observe from (2) and (3) that a node colliding on one colour and then on a different colour ensures that its selection probabilities for all channels are strictly greater than $b(1-b)/(c-1)$. Similarly if a node succeeds and then collides. However, it can be seen from (2) that repeated collisions on the same channel can result in the channel selection probability becoming arbitrarily small. Thus, 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 channel selection probabilities lower than $b(1-b)/(c-1)$ at one or more colliding nodes, then there is a positive lower-bounded probability that it will return in two steps to our standard state $\mathbb{S}$.

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 being in state $\mathbb{S}$. We ignore nodes who succeed and then collide as their channel selection probabilities are clearly at least $b/(c-1) > b(1-b)/(c-1)$. We ignore nodes which collide on one channel and then another as their channel selection probabilities are also at least $b(1-b)/(c-1)$. Consider any just collided node and say the collision has occurred by choosing channel i_1. From (3), the node now has probability $pr_1 > b/(c-1)$ of choosing some specific other channel i_2 and so probability $pr_2 > (c-1)pr_1 = b$ of choosing any channel other than i_1. So the probability of two repeated collisions on the same channel at a specific node is $pr_3 < 1 - pr_2$. In the whole system the probability of some node having two consecutive same channel 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 channel collisions. Thus after these two steps with probability pr_5 all channel selection probabilities of nodes which have just collided are strictly greater than $b(1-b)/(c-1)$ and the system is in state $\mathbb{S}$. ■

A key issue is that transitioning to a non-interfering channel allocation can require non-local changes in the node channel allocations. That is, it may happen that a non-interfering channel allocation cannot be found by changing only the channels used by currently colliding nodes while leaving the converged nodes unchanged. We are therefore interested in the ability of a colliding node to force its neighbours to change colour, and for these neighbours in turn to propagate changes throughput the network if necessary. That is, we need to consider the appropriate connectedness.

We proceed by defining the directed graph DG which is dependent on the current channel selection by the network nodes. There is an edge in DG from node u to node v if an edge exists between u and v in the graph $G(i_v)$ where i_v is the channel currently chosen by node v. We say v is a DG-neighbour of u. The edges directed *into* a node v are determined by the channel selection of that node together with the conflict graphs G, but are unaffected by the channel selections of other nodes.

Existence of a directed path in graph DG from node u to node v indicates that the node u can potentially force a collision at node v (by first generating a collision with its immediate neighbour, which in turn can generate a collision with its neighbour, and so on until node v is reached).

DG-graphs associated with an example network are illustrated in Figure VII. Consider the lower left node. Edges involving this node only exist on channel R. Hence, this node can potentially create collisions with its neighbours by selecting channel R. However, by selecting channel B, the lower left node can always avoid interference from any of the other nodes regardless of their channel selection. This asymmetric nature of the relationship between the lower left node and its neighbours is indicated by the directional arrows on the DG-graph links.

Note also that once it chooses channel B, the lower left node in Figure VII is unreachable from the other nodes. Since it is unreachable, no collisions can occur, choice of channel B will yield a "success" in the CFL algorithm and the node will remain on channel B thereafter i.e. the node will be converged and permanently unreachable. That is, the CFL algorithm therefore ensures that unreachable nodes remain permanently unreachable. A second example illustrating this behaviour is also given in the right-hand graphs in Figure VII. These examples illustrate the general point that as the CFL algorithm proceeds connectivity can change and, in particular, certain nodes may become permanently unreachable and we need to take account of this when analyzing convergence.

We define the set of nodes CN to be all nodes which are unreachable from any node which just collided. We note that any node $w \in CN$ must have just been successful. In addition, no matter what colour choices other nodes make in the future, w will never subsequently undergo a collision (since w is unreachable). Hence any nodes in CN are converged and can be ignored for the remainder of the proof. Note that the graph DG changes as the algorithm proceeds, and nodes can join CN but will never leave. In Figure VII we see two stages of the algorithm, the corresponding DG, and the set CN illustrated by nodes in bold.

Lemma 2. Suppose that the system is in state $\mathbb{S}$. There exists a specific evolution $\mathbb{E}$ of the system which results in all nodes not in CN colliding.

Proof of Lemma 2. Consider one of the collisions. 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 DG-neighbour k_3 (if any) at step 3. Suppose also that k_2 collides with its first non visited DG-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 DG-neighbours in consecutive steps. This is possible because a visited node having just collided can potentially choose any channel. Note that a node being visited simultaneously (along two different equal length paths from k_1 and k_2 say) is also possible.

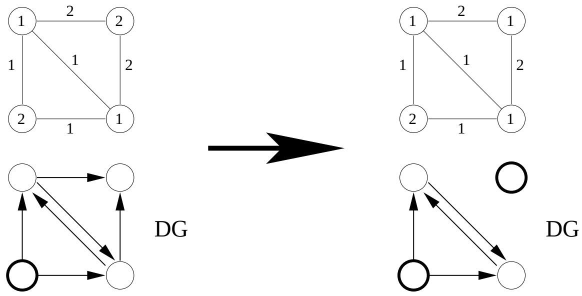

Fig. 9. Illustrating definition of DG-graphs. Upper graphs show node channel selections and the channel-dependence of edges is indicated by labels. Lower graphs show corresponding DG-graphs. The set CN indicated by nodes in bold.

Suppose that once a node has collided with all its nonvisited neighbours it then repeatedly chooses channel 1 until step $T_1 = T_0+3+md\times D$. We note that as a node k_4 is colliding with its nonvisited DG-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.

Concurrently with this visiting procedure starting at the nodes k_1 and k_2, we can suppose that the same visiting procedure starts at all nodes in JC, and traverses the graph as before. Again as a node k_5 is colliding with its nonvisited DG-neighbours some of them may become visited from other nodes, and we again suppose that k_5 does not visit such nodes.

When all the visited nodes have visited all their neighbours, every node not in CN has been visited and is choosing channel 1. Some nodes which are now choosing channel 1 may of course have entered the set CN and are ignored. Hence every node not in CN is colliding. At the next time step we suppose that every node chooses a colour so that no collisions occur. ■

Lemma 3. Suppose that there exists a choice of channels that yields a non-interfering allocation. 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 the set JC, the evolution $\mathbb{E}$ is well defined. The duration of $\mathbb{E}$ is at most $md \times D$ timesteps. Hence $\mathbb{E}$ 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 colour selection probabilities.

The set JC is one of finitely many possibilities and so again there is some probability $pr_8 > 0$ such that $pr_7 > pr_8$ irrespective of the initial choice of JC. ■

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 $\mathbb{E}$. Hence 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 $j \to \infty$. ■

VIII. Comments

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. We consider multiple radios in more detail below.

(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.

converges to a proper channel allocation in the presence of hidden nodes or legacy/uncooperative nodes, although a non-interfering allocation may require a larger number of channels than when such nodes are not present.

(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.

IX. Multiple radios

The use of wireless access points equipped with multiple radios has been the subject of much recent interest. As noted previously, the CFL algorithm can be applied without change to multi-radio access points by running a separate copy of the CFL algorithm for each radio. This will yield a non-interfering channel allocation for every radio. In this section we illustrate that the CFL algorithm can be further generalised to take explicit account of bit rate requirements in a multi-radio setting.

Specifically, we consider the following task. Suppose we have a set of interfering WLANs (possibly with channel-dependent interference) and a set $\mathcal{C}$ of available channels. Let b_i denote the bit rate associated with channel i. At access point j we require to select a non-interfering set of channels $C \subseteq \mathcal{C}$ such that $\sum_{i \in C} b_i \geq B$ and with cardinality $|C| \leq r$, where r is the number of radios at the access point. Note that the channel bit rate b_i, the target bit rate B, number of radios r and set of available channels $\mathcal{C}$ may be different for each access point.

The change here over our previous discussion is the inclusion of the bit rate constraint $\sum_{i \in C} b_i \geq B$. Such a bit rate requirement arises, for example, when striping data across multiple radios. One advantage over simply allocating a channel to every available radio is that it may be that fewer radios are sufficient to provide the required bandwidth, thereby reducing the load on the spectrum in dense WLAN deployments. This formulation also allows us to take explicit account of the different quality of each channel – this can be important in multi-radio settings where radios are heterogeneous e.g. some radios might be 802.11 based and others 802.16 based. We note that the bit-rate constrained channel allocation problem is also relevant to dynamic spectrum management in wired DSL lines (where cross-talk across wiring bundles is a significant source of interference) [18].

We introduce the following generalised version of the CFL algorithm to solve the multiple radio bit-rate constrained channel allocation problem.

Let c denote the number of available channels at an access point and the access point maintain a c element state vector p with element p_i corresponding to the probability of transmitting on the ith channel. Since we allow use of multiple radios, note that we do not require the p_i's to sum to one. Consider the following decentralized algorithm for updating p.

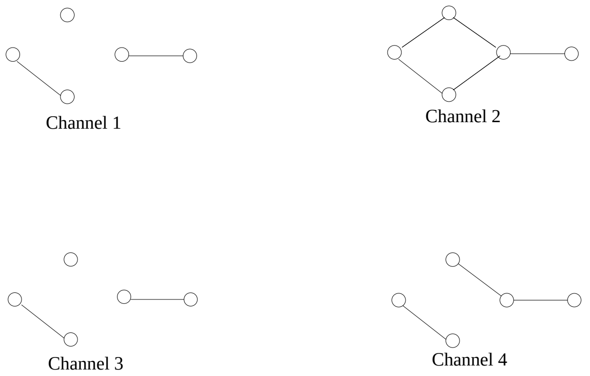

Fig. 10. Five AP example of multiple radios. Each graph gives the interference for a particular channel.

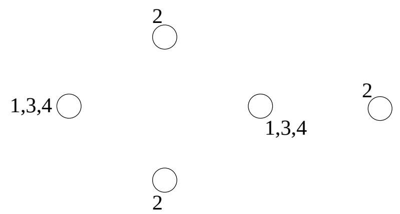

Fig. 11. Possible final result of multiple radio algorithm. The figure shows a successful channel allocation.

Generalised CFL Algorithm

1) Initialise $p = [r/c, r/c, \ldots, r/c]$
2) Pick a random ordering of the channels. In that order, toss coins to activate channel i with probability p_i. Stop immediately once the AP's target bit rate is met. This results in a set C of active channels.
3) If $\sum_{i \in C} b_i < B$, multiply every probability by $1 + b$, set $C = \emptyset$, and repeat[3] step 2. We note that the random selection process at step 2 above together with the redistribution of probability in step 6 below, ensures that there is a positive lower bound on the probability of any feasible allocation after a collision.
4) Sense the quality of the channels in set C. We obtain "success" on channel $i \in C$ if this does not interfere with any neighbouring WLAN and otherwise have a "failure".
5) If we have success on all active channels, update p as

$$p_i = 1 \; \forall i \in C, p_j = 0 \; \forall j \notin C \qquad (4)$$

i.e. on a successful choice we use the same set of channels for the next round. This ensures that any channel allocation that satisfies the target bit rate and also removes interference between all WLANs is an absorbing state.
6) Otherwise let S denote the set of channels which were successful, F the set of failed channels and I the set of inactive channels. Update p as

$$\begin{aligned} p_i &= 1 \; \forall i \in S, \\ p_j &= (1-b)p_j + b \; \forall j \in I, \\ p_k &= (1-b)p_k \; \forall k \in F. \end{aligned}$$

The lower bound on these probabilities after the node fails is much more important than the exact choice of parameters.
7) Return to 2.

[3]The precise procedure here is not important. The feasible active set may be found in any reasonable fashion provided any channel with nonzero p_i might be active and that channels with larger p_i are more likely.

This algorithm maintains the three key properties of the original CFL algorithm, namely (i) that if every WLAN is successful the system remains in this successful configuration henceforth; (ii) after a collision any feasible channel allocation is possible; and (iii) if one WLAN is failing the failure can propagate to neighbouring WLANs and force them (with some probability) to change their channel allocation. Hence by a similar proof to that for Theorem 1 the generalized CFL algorithm will converge with probability 1 to a non-interfering channel allocation satisfying the specified bit rate requirements, provided one exists.

In Figure 10 we present an example of the multiple radio problem. Suppose that every AP has bit rate demand 3 units; suppose that channel 2 has bit rate 3 units and that all other channels have bit rate 1 unit. Figure 11 illustrates a feasible channel allocation which is a result of the algorithm. Figure 12 shows the time history of the channel choices made by the leftmost node in Figure 10 as the algorithm proceeded.

X. Conclusions

In this paper we consider channel allocation to mitigate interference between wireless LANs. The channel allocation task is often formulated in the literature as finding a proper colouring of a single graph In this paper we demonstrate that this formulation may be unrealistic. Specifically, we show that the interference between WLANs can be channel dependent in which case a different conflict graph is associated with each channel. Channel allocation then corresponds to a multi-graph colouring problem. This potentially has profound

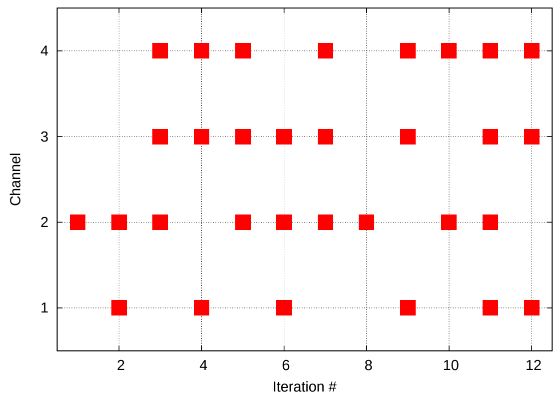

Fig. 12. Time history instance of multiple radio algorithm. The graph shows the channel choices made by the leftmost node in Figure 10 as the algorithm converges to the successful allocation illustrated in Figure 11.

implications as the behaviour of proposed colouring-based algorithms for channel allocation is unclear in a multi-graph context. We are, however, able to show that a recently proposed decentralized colouring algorithm does generalise to the multi-graph setting. We also present a new, extended version of this algorithm suited to a wide range of multi-radio architectures.

XI. Acknowledgements

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

References

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

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

[4] 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.

[5] 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.

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

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

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

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

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

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

[16] D.W.Malone, P. Clifford, D. Reid, D.J.Leith," Experimental Implementation of Optimal WLAN Channel Selection Without Communication". Proc DySpan 2007, submitted

[17] K. Duffy, N. O'Connell, A. Sapozhnikov, "Complexity analysis of a decentralised graph colouring algorithm". Hamilton Institute Technical Report, 2006.

[18] T. Starr, J.M. Cioffi, P. Silverman, "Unerstanding Digital Subscriber Lines", Prentice Hall: Upper Saddle River, NJ, 1999.