Frequency Allocation for WLANs Using Graph Colouring Techniques

Janne Riihijärvi, Marina Petrova, Petri Mähönen
Aachen University, Department of Wireless Networks, Kackertstrasse 9, 52072 Aachen, Germany
E-mail: jar@mobnets.rwth-aachen.de

Abstract

At present, no standard frequency allocation mechanism exists for Wireless LAN access points. In this article, we introduce a number of techniques based on graph colouring algorithms, and demonstrate their effectiveness using simulations. We also suggest a preliminary message format the access points could employ to exchange information regarding the wireless channel, and elaborate on the possible protocol architectures that could be used in the actual channel allocation process.

1. Introduction

During the past few years the number of Wireless Local Area Networks (WLANs for short) has experienced rapid growth. Not only has the number of deployed company and university networks increased, but also public hotspots have become common, and their number is still exploding rapidly. However, as wireless access points (APs) become densely deployed, problem of frequency selection arises. With the common 802.11b WLAN technology, only three non-overlapping channels are available, and no standard mechanism exists for the access points to dynamically select the channel to be used as to minimise interference with other APs.

2. Colouring in frequency allocation

In this section we shall formulate the frequency allocation problem for WLANs in terms of graph-theoretic colouring problem, and introduce various algorithms for solving these problems, paying attention to their suitability for implementation in the WLAN context. But first we recall shortly the statement of the colouring problem that is of interest in the frequency allocation context. For a comprehensive exposition on basic graph theory we refer the reader to [2].

Suppose we are given a simple graph $G = (V, E)$, that is, a graph consisting of a set of vertices V, and set of edges E connecting the vertices so, that loops (edges connecting a vertex to itself) and multiple edges between vertices are not allowed. Then a *vertex colouring* of G is a map $c : V(G) \to F$, where F is a set of *colours*, usually some small subset of positive integers. We shall call a colouring *admissable*, if $c(V_i) \neq c(V_j)$ for all *adjacent* V_i and V_j (that is, for those vertices connected by an edge). We call an admissable colouring minimising $|c(V)|$ (that is, the size of the colour set used) an *optimal* colouring. The number of colours used by the optimal colouring is called the *chromatic index* of the graph.

2.1. Interference graphs

We shall now formulate the channel allocation problem in terms of the terminology introduced in the previous section. Given a collection $\{V_i\}$ of access points (or radio transcievers in general), we shall form an *interference graph* $G = (V, E)$ as follows. The vertex set V is simply identified with the set $\{V_i\}$. The set of edges E is constructed as the union of those pairs $\{V_k, V_l\}$ of vertices, that correspond to access points V_k and V_l that would interfere with each others' radio traffic should they be assigned to use the same channel. Finally, we let F, the set of "colours", to be the collection of channels available to the access points. Now the channel allocation problem is simply finding of an admissable colouring of G with the colour set F.

Naturally the size of the colour set is greatly technology and legislation -dependent. In most European countries, $F = \{1, 2, \dots, 13\}$ for the IEEE 802.11b and 802.11g technologies, of which the subset $F' = \{1, 6, 11\}$ corresponds to the non-overlapping channels. For the IEEE 802.11a the set of non-overlapping channels is considerably larger.

In the following we shall discuss three classes of colouring algorithms for solving the colouring problem under different constraints.

2.2. Classical colouring with heuristics

It is well known that the colouring problem is NP-hard. While on the outset this might make it seem unsuitable for the present use due to the exponentially increasing computational requirements as the number of access points is increased, this fortunately turns out not to be the case. Namely, a number of efficient heuristics have been developed, leading to polynomial-time colouring algorithms that give very good approximations to optimal colourings.

A particularly simple, yet effective heuristic called the "degree of saturation" was proposed by Brélaz in [1]. It is defined as the number of differently coloured neighbours of a vertex (or, equivalently, as the number of non-admissable colours of a vertex).

The algorithm itself consists then of the following steps:

1. Initialise the degrees of saturation of all vertices to zero.
2. Select the uncoloured vertex of highest degree of saturation. If more than one vertex have the same degree of saturation, choose the one with the highest number of uncoloured neighbours.
3. Colour the selected vertex in a greedy manner, that is, using the smallest colour admissable.
4. Update the degrees of saturation of the uncoloured vertices neighbouring the one coloured in the previous step.

The steps 2–4 are repeated until all vertices are coloured.

To complement this heuristic description of this "DSATUR" algorithm, we shall also give a more rigorous description. We shall denote the neighbourhood of i by $\gamma(i)$, the degree of saturation of i by $d(i)$, and the colour assigned to i by $c(i)$. We use the convention $c(i) = 0$ if i is uncoloured. The set of uncoloured vertices is denoted $U = \{i \in V \mid c(i) = 0\}$, and we identify a set with a sing le element with the element itself. With this notation, the DSATUR algorithm for a graph (V, E), with large enough set of colours C (taken to be a subset of the set of natural numbers) can be expressed in the following form:

```
$\forall i \in V : d(i) := 0, c(i) := 0$
$U := V$
while ($|U| > 0$) {
    $S := \arg\max_{i \in U} d(i)$
    if $|S| > 1$: $j := \arg\max_{i \in S} |U \cap V(\gamma(i))|$
    else $j := S$
    $c(j) := \min\{i \in C \mid c(k) \neq l \; \forall l \in V(\gamma(j))\}$
    $\forall i \in V(\gamma(j))$: if $c(j) \cap c(V(\gamma(i))) = \emptyset$: $d(i) := d(i) + 1$
    $U := U - j$
}
```

Given an implementation using suitable data structures the algorithm runs in $O(m \log n)$ time, where m and n are the size and the order of the graph, respectively. In practise we have found that the constant factor hidden by the O-notation is so small, that even when very limited processing power is avalable, the algorithm can be run on a rather dense graph of hundreds of vertices in less than a second.

The DSATUR algorithm has also other attractive properties (for full discussion the reader is referred to [7], and references therein). In addition to processing time, the associated memory consumption scales in a very acceptable manner. Also the colourings produced by the algorithm are very good, actually optimal for large classes of graphs. Final, and in our application vital property is that the algorithm can be implemented in a purely deterministic manner. Similarly to the case of, for example, routing protocols, this allows the colouring algorithm to be used in a distributed manner, without the need of a central "channel assigment authority".

2.3. On-line colourings

Due to technological limitations the access points should not change channels while serving users. This is because there is no standard technique to instruct the user equipments to change their channels, leading to broken connections. We therefore would do best to perform the updates to the colourings as a new access point V_k is added to the interference graph so, that the existing channel allocation are unaffected, if possible.

Fortunately a class of colouring algorithms, called online graph colouring algorithms exists that can be applied for such a channel allocation problem. The downside is, that typically these algorithms have a very poor worst-case performance (see, for example [4] and references cited therein) in terms of the colours used. The absolute upper limit for the performance ratio r in [4] is shown to be

$$r = O\left(\frac{n}{\log^2 n}\right).$$

For an example on-line colouring algorithm in the channel assignment context, with a detailed analysis the reader is referred to [6].

Even with their downsides, on-line colouring algorithms might prove valuable assets, especially in the cases of technologies with larger number of non-overlapping channels. Of course, also a hybrid scheme in which "regular" colouring algorithm is run whenever the performance of the online colouring becomes too poor might also be worth pursuing.

2.4. T-Colouring

Finally, we consider algorithms that are suitable for interference graphs so dense, that the channel set F' of *non-overlapping* channels is not large enough for admissable colouring. In this case we perform the colouring using the complete channel set, and impose a distance-type of condition for the channels of adjacent vertices. The corresponding colouring problem is called the T-colouring problem.

More precisely, the main change compared to "regular" colouring algorithms is the introduction of a set T of integers, representing disallowed colour separations. More concretely, for a T-colouring function c_T we have the additional requirement that

$$c_T(V_1) - c_T(V_2) \notin T$$

holds for all neighbour vertices V_1 and V_2. The concept of T-colouring in fact originated in the frequency assignment context in the classic paper of Hale [3].

A vast body of knowledge has been accumulated during the past two and a half decades on the T-colouring problem. For a nice, albeit a bit dated review, see [5].

3. Simulation setup

To obtain quantitative, albeit rather simplified understanding on the performance of the proposed scheme a number of simulations were performed with ns-2. As ns-2 does not allow for modeling of cochannel interference, focus was on three-colourable interference graph topologies, and classical graph colouring heuristics. Different channels were used by instantiating multiple instances of the class "WirelessChannel", and connecting the access points and nodes to the channel instances according to the obtained channel allocation.

The colouring algorithm itself was implemented in Java as an external program. Due to reasons discussed above, we chose the DSATUR-algorithm as the classical heuristic algorithm to be used. From the tcl-file describing the simulation setup, a script was used to extract only the access points into a separate tcl-file used to determine the interference graph topology. This was accomplished by attaching sources and sinks into each access point, and recording the pairs between which communication was possible. The graph so obtained was then given to the Java-program for colouring, and in the final stage tcl-files were created to give both the coloured and the random channel assignments, based on the original scenario file. Finally, traffic flows consisting of CBR-traffic over UDP and/or FTP-traffic over FullTCP connections were put into place. In these simulations the nodes were static. Mobility will be considered in future simulations, but was not perceived as a critical component at the present time, as the users of, say, WLAN hotspots are typically rather static.

Three parameters were considered in evaluating the effectiveness of the colouring approach:

- The total number of collisions occuring in the network. To obtain this statistic ns-2 source code was augmented to give extra output regarding the events taking place in the wireless link.
- The (aggregate) throughput of the network.
- In the case of TCP traffic, the perceived connection round-trip-time and its variance. Both of these statistics were obtained from the FullTCP implementation of ns-2, and were taken from randomly selected TCP end-nodes.

Especially the TCP-related performance figure are of interest, as poor link conditions will be reflected in high variance of the RTT, due to retransmissions taking place in the link-layer.

4. Simulation results

In this section we shall illustrate the results obtained from the simulations described above. We would like to emphasise that these simulations are not meant to give a *precise* quantitative effectiveness of the performance of the colouring approach. To accomplish that a massive number of simulations have to be performed, with carefully controlled location distributions for the access points and the corresponding nodes serviced. Also the channel model and the reception process used at the nodes should be modelled more carefully than is done at present in ns-2. However, as the reader will see, the obvious consistency of the results clearly shows that the colouring algorithm works markedly better than the random assignment of colours, and definitely warrants further studies and *measurements*.

4.1. Number of collisions

We shall begin by considering the number of MAC collisions as the function of the number of access points, as new access points are being randomly placed on the simulation area, and the channel assignments are performed for *all* the access points. (As discussed above, the incapability of ns-2 to simulate co-channel interference more or less makes the simulation of the on-line colouring algorithms impossible.) Figure 1 shows the results for pure UDP traffic.

The results for pure TCP traffic are likewise shown in figure 2.

In the mixed traffic case the results are qualitatively unchanged, as can be seen from figure 3.

4.2. Aggregate throughput

While the number of MAC collisions is a clear and simple indicator of the link conditions, a better characteristic to measure the overall performance of the wireless network is the aggregate throughput achieved for the applications.

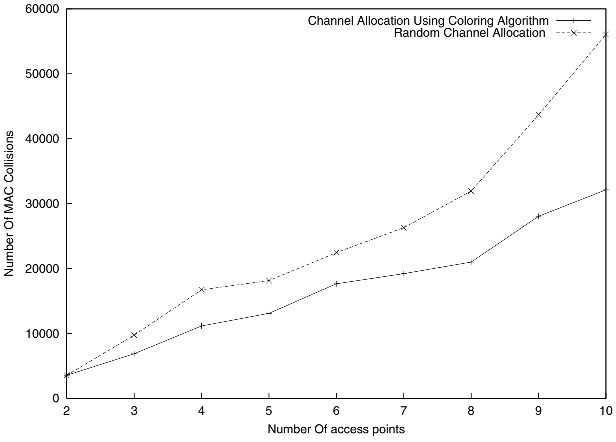

Figure 1. Number of MAC collisions as a function of the number of access points in a network carrying purely UDP traffic.

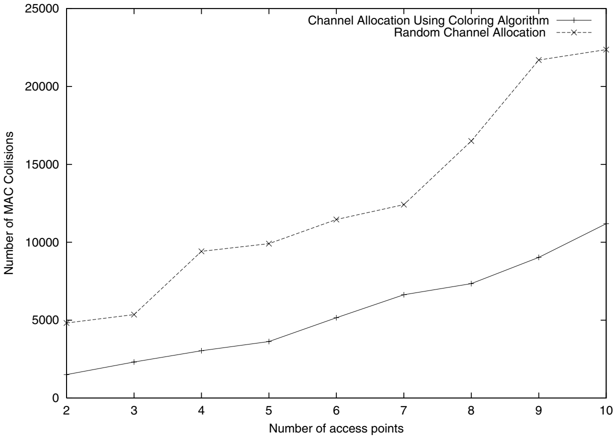

Figure 2. Number of MAC collisions in a network carrying purely TCP traffic.

Figure 4 shows the development of the aggregate UDP throughput as the number of access points is increased. With each access points, five stations are added to the network. We see that with the application of the colouring algorithm an almost linear growth of throughput is obtained (which should be the case if the traffic through different access points is non-interfering), which is an improvement from the rather jumpy and considerably slower growth in the random case.

As seen in figure 5, the situation is largely unchanged in the case of pure TCP traffic, except for the slightly lower absolute throughput, which is due to the higher protocol overhead.

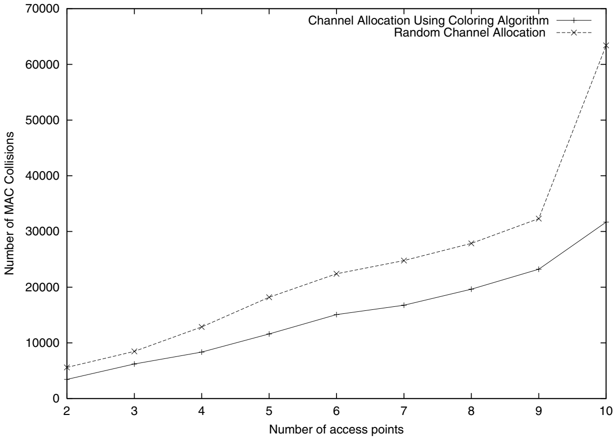

Figure 3. Number of collisions in the heterogeneous traffic case.

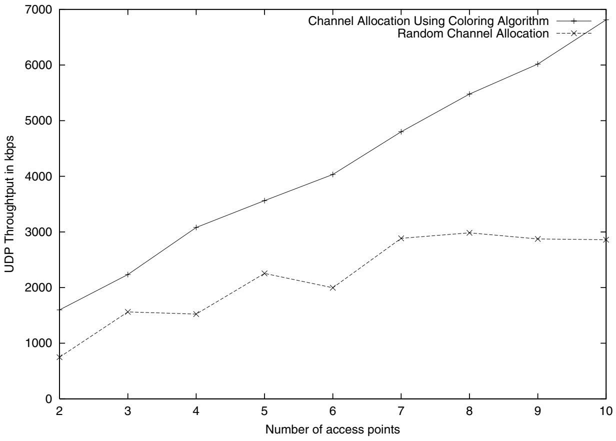

Figure 4. Aggregate throughput of the network carrying UDP traffic.

Finally, in figure 6 we see that the overall conclusions regarding the improvement of the throughput as graph colouring is used remain unchanged in the presence of both UDP and TCP traffic.

4.3. TCP traffic conditions

In the case of TCP traffic, the results are depicted in figure 7.

Clearly the RTT behaviour is substantially more "calm"

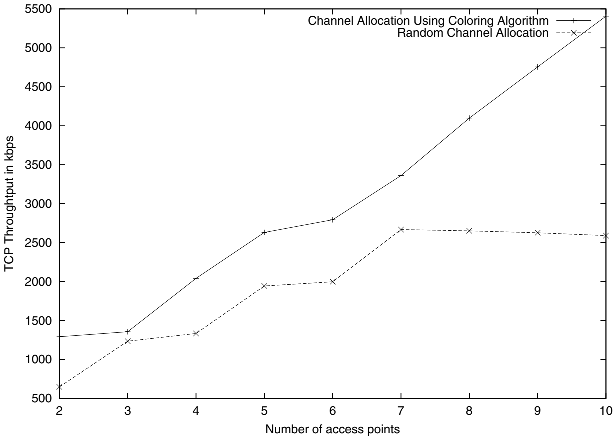

Figure 5. Aggregate throughput of the network carrying TCP traffic.

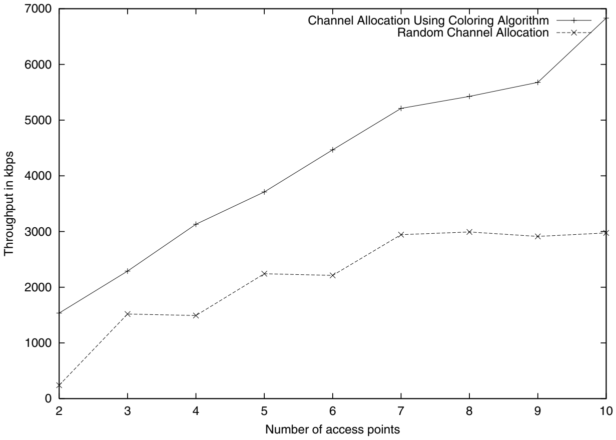

Figure 6. Aggregate throughput in the heterogeneous traffic case.

when the channel allocation using the colouring algorithm is applied. Due to the structure of the TCP data flow this leads to faster recovery times when transmission errors occur (not a rare event in wireless LANs), and the overall responsiveness of the connection is also enhanced. This is especially important for the case of interactive applications, already starting on the level of web browsing.

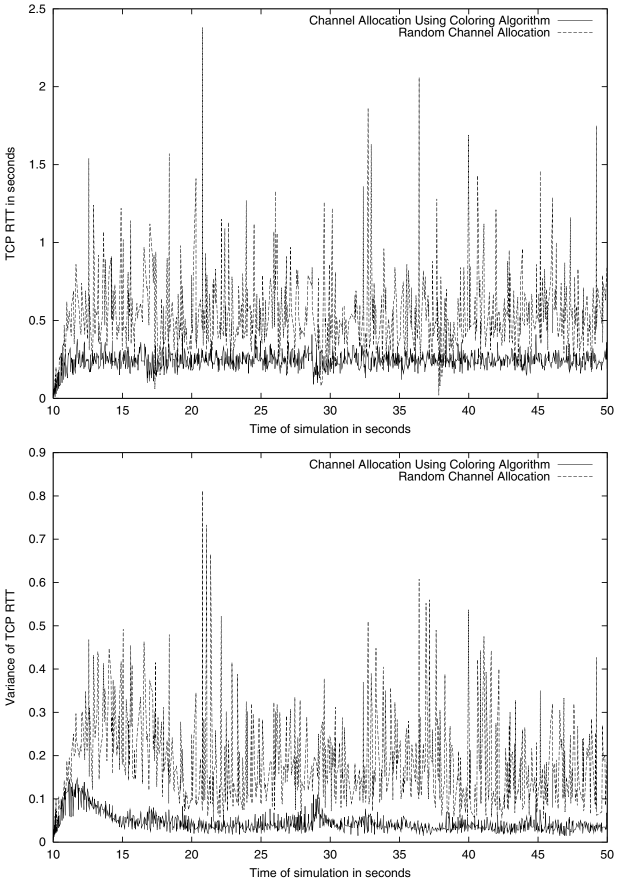

Figure 7. TCP RTT and its variance.

4.4. More complicated scenario with TCP traffic

To show that our scheme carries benefits even in a slightly more realitic case of a larger network with mobile nodes, and varying communication ranges, we performed simulations with the following characteristics:

- Twelve access points, with communication ranges ranging between 100 meters and 250 meters.
- Number of nodes served by an access point was randomly chosen from $\{5, 6, 7\}$.
- Traffic sources and sinks for both UDP and TCP were randomly allocated.
- Nodes move randomly within the access point serving radius'.

As an example of the results obtained, in figure 8 the round-trip-time of a representative node is depicted, together with the corresponding variance estimator. Large difference may be observed between the randomly assigned channels, and the channel assignment based on the colouring algorithm.

To sum up the simulation results presented, the performance of networks with frequncy allocations done using the colouring approach compares clearly and consistently favourably to performance in networks with ran-

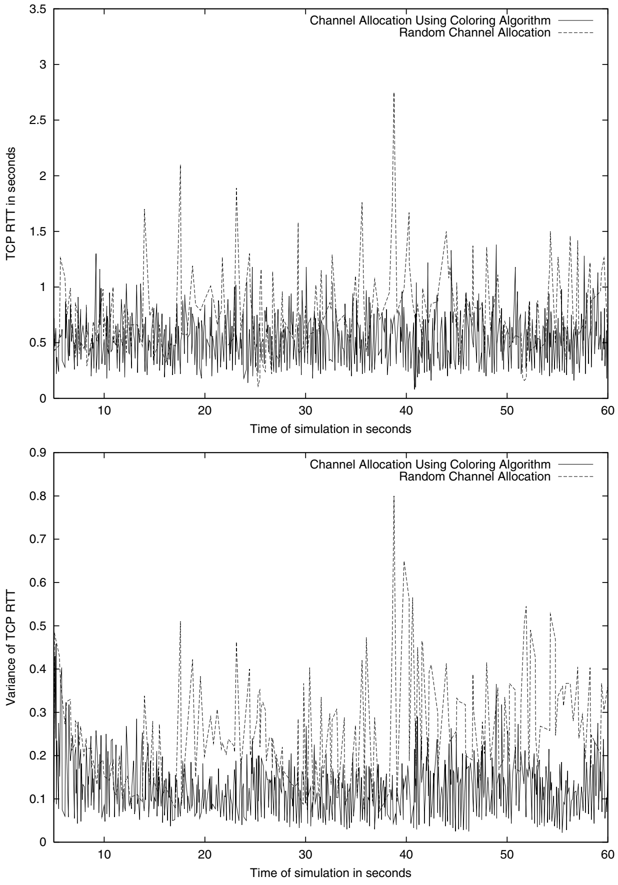

Figure 8. TCP RTT and its variance in a more realistic simulation scenario.

domly assigned frequencies. In fact, many of the existing networks employ even worse channel allocation, with all access points using the *same* channel (factory default).

5. Protocol considerations

In this final section of the present article, we shall discuss the choices to be made when implementing an automated channel assignment mechanism to access points. We shall begin by describing the common message format that all the different AP-to-AP communication mechanisms can use, and then go on to discuss various techniques the APs can use to share the information necessary to construct the interference graph.

5.1. Radio neighbourhood description messages

The minimum set of data the access points need in order to construct the interference graph is neighbour information. For this, a message consisting of one neighbour entry N_i per neighbour observed is sufficient, preceded by some mandatory header information. For this preceding header, the following structure is rather minimal:

MY_NEIGHBOURS	MY_MAC_ADDR
MY_MAC_ADDR (cont.)	

32 Bits

Here the first 16-bit field gives the number of neighbour entries following the header, and the second 48-bit field gives the MAC address of the wireless interface.

In the simplest colouring application the neighbour entries N_i could in principle consist of only of the 48-bit MAC-addresses the station originating the message can hear. However, to enable the use of T-colouring and the online colouring algorithms, information regarding the present channels assigned to the access points, and some measure of the level of the received signal is necessary. Thus we propose the following format for the neighbour entries.

CH	SNR	MAC_ADDR
MAC_ADDR (cont.)		

32 Bits

Here the first two 8-bit fields give the channel the corresponding neighbour was heard using, and the measured SNR. Main problem here is, of course, the measurement of the SNR. Different 802.11-chipsets give results that are not always consistent, thus leading to discrepancies in the information supplied in the messages. However, for the time being we are probably forced to live with these difficulties. This might in part change with the upcoming 802.11k standard, that will focus on radio resource measurements in the frequency bands the WLANs use.

The signalling overhead in terms of messages obviously scales as $O(N^2)$ in terms of the number N of nodes participating to the frequency coordination process. The size of each message is 64 bits per radio neighbour (plus 64 bits for the header). Thus, with typical access point densities the messages can usually be carried inside single frame, without having to resort to IP-layer fragmentation.

When specifying a complete protocol for use, a slightly more modular structure, also carrying optional information for the colouring algorithms (such as the T-set used) might turn out to be necessary. The above message structure is only meant to be the initial building block to construct further suggestions on.

5.2. AP-to-AP communications

For most conceivable variants of the colouring based channel allocation the basic first step is the scanning of the channels for other access points, and formulating the radio neighbourhood information message as described above. Main differences in protocol operations would result from the selection of any of the various methods the access points can use to communicate this information amongst themselves.

The simplest way of disseminating the neighbourhood messages would be flooding via the radio interface. However, this is slightly inpractical as there is no common control channel in wireless LANs that all the access points listen to continuously. Thus, in the case of already operational WLAN, the access points should somehow coordinate the sending of the neighbourhood information, and change to the same channel for this procedure, or send the information on several channels. Unfortunately this would result in breaking the associations with the stations the access points serve, so this probably cannot be done except during times of very small number of users. The 802.11h promises to change this by introducing mechanisms for dynamic frequency changes, but, being designed to be interoperable solely with 802.11a, does not seem to be facing a bright future, at least in the short term.

More promising approach is to use the infrastructure network connecting the access points, especially in the case where the same (say) company owns all the access points. In the case of access points being wireless routers as well, the MAC address heard over can be mapped into IP addresses using a simple ARP-call, and any IP-based protocol can then be used for the final delivery of the neighbourhood information. Of course, if multicast support in routers (wireless and fixed) becomes commonplace, using geography-based multicast groups for access points might also turn out to be a good option.

As the number of hot spots in the public places is increasing there is a probability that the number of WLANs by different providers (companies) is getting larger so the environment gets heterogeneous. In such a case all the access points will not be connected to the same 'ethernet cable' so there will be no possibility to match the MAC addresses to IP addresses via the ARP protocol and furthermore deliver neighbourhood information. In order to be able to establish collaboration between neighbouring nodes from different networks it might be convenient having a service discovery protocol providing this capability. One possibility can be using location discovery capabilities to find the neighbours.

6. Conclusions

The application of the colouring algorithm to the channel allocation problem in WLANs presents an attractive solution to automated interference minimisation in the infrastructure mode. Depending on the exact boundary conditions imposed by the access point deployment model, number of well-studied and effective colouring algorithms are known. First simulation studies indicate that the presented form of channel assignment works effectively, and provides real benefits when contrasted with uncoordinated channel selections. Finally, numerous protocol solutions seem to be available for implementing the described channel allocation mechanisms, showing that the scheme described can be deployed in real-life test networks for more realistic, measurement-based performance studies. These performance evaluation activities will be carried out in future work.

7. Acknowledgments

We would like to thank DFG (Deutsche Forschungsgemeinschaft), European Union (6HOP-project) and Aachen University for financial support. We would also like to thank Harish Koka for assistance in performing the simulations.

References

[1] D. Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22:251–256, 1979.

[2] R. Diestel. *Graph Theory*. Springer-Verlag, 2000.

[3] W. K. Hale. Frequency assignment: theory and applications. *Proceedings of the IEEE*, 68:1497–1514, 1980.

[4] M. M. Halldórsson and M. Szegedy. Lower bounds for on-line graph coloring. In *Proceedings of the 3rd annual ACM-SIAM symposium on Discrete algorithms*, 1992.

[5] F. S. Roberts. t-colorings of graphs: recent results and open problems. *Discrete mathematics*, 93(2–3):229–245, November 1991.

[6] Y.-T. Tsai, Y.-L. Lin, and F. R. Hsu. The on-line first-fit algorithm for radio frequency assignment problems. *Information processing letters*, 84:195–199, 2002.

[7] J. Turner. Almost all k-colorable graphs are easy to color. *Journal of Algorithms*, 9:63–82, 1988.