An Asynchronous Distributed Dynamic Channel Assignment Scheme for Dense WLANs

Micheal Drieberg¹, Fu-Chun Zheng², Rizwan Ahmad¹ and Sverrir Olafsson³
¹School of Electrical Engineering, Victoria University, Melbourne, Australia
²School of Systems Engineering, University of Reading, Reading, UK
³Mobility Research Centre, BT Group, Martlesham Heath, Ipswich, UK
E-mail: micheal.drieberg@research.vu.edu.au

Abstract—Wireless local area networks (WLANs) have changed the way many of us communicate, work, play and live. Due to its popularity, dense deployments are becoming a norm in many cities around the world. However, increased interference and traffic demands can severely limit the aggregate throughput achievable if an effective channel assignment scheme is not used.

In this paper, we propose an enhanced asynchronous distributed and dynamic channel assignment scheme that is simple to implement, does not require any knowledge of the throughput function, allows asynchronous channel switching by each access point (AP) and is superior in performance. Simulation results show that our proposed scheme converges much faster than previously reported synchronous schemes, with a reduction in convergence time and channel switches by up to 73.8% and 30.0% respectively.

I. INTRODUCTION

Once thought to be just a replacement for wires in local area networks (LANs), WLANs have found its way into our workplace, homes and personal space. Its popularity has resulted in dense deployment of WLANs in many cities around the world. While the increasing density of deployment works well to provide a myriad of users with seamless wireless broadband services, new challenges such as channel assignment has to be addressed effectively so that users can continue to enjoy creative and innovative services provided through WLANs.

Further aggravating the problem is the fact that the networks belong to different owners and hence are managed by different network administrators. This severely limits the amount of cooperation that can exist among different networks. On the other hand, each network’s load also changes dynamically. Therefore, an effective distributed and dynamic channel assignment scheme is needed to maximize the total throughput achievable to the networks.

The international standard for WLANs is the IEEE802.11x group of standards. IEEE 802.11b operates at 2.4 GHz with 3 non-overlapping channels while 802.11a at 5 GHz with 12 non-overlapping channels. The limited number of channels further exacerbates the channel assignment problem.

Channel assignment in WLANs has been researched actively. By using the traffic load information on each AP, Leung and Kim [1] proposed a heuristic algorithm based on subset search to minimize the channel utilization at the AP with the maximum utilization. Mahonen, Riihijarvi and Petrova [2] modeled the channel assignment problem as the classical graph colouring problem and proposed a heuristic algorithm called degree of saturation (DSATUR). However, these schemes require information about the whole network and can only be implemented in a centralized manner.

A distributed and dynamic channel assignment scheme was proposed in [3]. A key assumption in the work, however, is that all the APs can perform the algorithm and switch into the new best channel simultaneously or synchronously. Clearly, such an assumption is too strong, especially in a distributed scenario. The algorithm also assumes knowledge of the throughput function which may not be possible in practice.

In this work, we propose an enhanced channel allocation scheme, which has the following key features: (1) it is asynchronous, distributed, and dynamic, (2) it has a much lower complexity, and (3) it does not require knowledge of the throughput function. In other words, all the APs are now allowed to perform the algorithm and channel switching asynchronously. Simulation results shows that our proposed scheme not only converges significantly faster, but also achieves better throughput in dynamic environments.

II. SYNCHRONOUS DISTRIBUTED DYNAMIC CHANNEL ASSIGNMENT SCHEME

Lou and Shankaranarayanan proposed a synchronous distributed scheme in [3] where each AP calculates the best channel to switch to in the next period in order to maximize its local aggregate throughput. The scheme has the objective of improving the overall throughput in the entire network by improving the throughput at each of the AP. They assumed that each of the neighbouring APs will broadcast the number of nodes associated with it, hence give an indication of the associated load. It is also assumed that all the APs can perform the algorithm and switch into the new channel simultaneously. For simplicity, the APs are assumed to be located in a rectangular array (as in Fig.1, which will be explained later).
The first step of the algorithm in [3] is to determine the throughput of an AP located at \((i, j)\) when using channel \(k\), given by
\[
P_k(i, j) = \frac{\sum_{(m,n)\in S_i} N(i,j)}{\sum_{(m,n)\in S_i} f(\sum_{(m,n)\in S_i} N(m,n))}
\]  
(1)

where \(N(i,j)\) is the number of nodes associated with AP\((i, j)\), \(S_i\) is the set of neighbouring APs including itself that is using the same channel \(k\) and \(f(L)\) is the throughput function of a channel that is shared by \(L\) competing nodes.

The next step is to find the channel \(k_m\) for which the throughput \(P_k(i,j)\) is maximized:
\[
k_m = \arg \max_{k\in C} P_k(i, j)
\]  
(2)

where \(C = \{1,2,\ldots,K\}\) is the discrete channel set (e.g. \(K = 3\) for IEEE 802.11b).

The final step is to switch into channel \(k_m\) with probability \(p\), the switching probability. The authors of [3] showed that the algorithm will incur adverse effects if \(p = 1\) because all the APs will simply oscillate back and forth at two channel settings. The optimal value of the switching probability was found to be \(p = 0.5\), that gives the best overall throughput in the shortest convergence time. Results in [3] show that the algorithm offers up to 70% improvement in overall throughput in less than 20 iterations.

In Fig. 1, we have used the same graphic style as in [3]. Fig. 1a shows one realization of the network grid with the initial channel of each AP and number of associated nodes selected randomly. The size of the circle in each square is proportional to the number of associated nodes of each AP. The different shades represent the different channels used by each of the AP. Fig. 1b shows the channel assignment after the algorithm has converged. Clearly, neighbouring APs with high number of associated nodes will be assigned different channels whenever possible to maximize the throughput.

III. ASYNCHRONOUS DISTRIBUTED DYNAMIC CHANNEL ASSIGNMENT SCHEME

In this section, we propose an enhanced asynchronous distributed dynamic channel assignment scheme that is simple to implement and does not require knowledge of the throughput function. It also allows each access point (AP) to carry out the algorithm and switch to the new best channel asynchronously.

A. Simplified Algorithm

In fact, the algorithm described in Section II can be simplified, so that it does not require any knowledge of the throughput function \(f(L)\) and also reduces the implementation complexity. First we define the number of associated nodes of an AP,
\[
N(i,j) = N_A
\]  
(3)

and the sum of all the associated nodes of its neighbouring AP that is in the set \(S_k\) excluding itself,
\[
\left(\sum_{(m,n)\in S_i} N(m,n)\right) - N_A = N_k.
\]  
(4)

We can rewrite the maximization in Eq. (2) as
\[
k_m = \arg \max_{k\in C} P_k(i, j)
\]
\[
= \arg \max_{k\in C} \frac{N_A}{N_A + N_k} f(N_A + N_k)
\]
\[
= \arg \min_{k\in C} N_k
\]  
(5)

where \(C = \{1,2,\ldots,K\}\) is the discrete channel set, \(N_A > 0\) is a constant and independent of \(k\) and \(N_k \geq 0\). Eq. (5) is due to the fact that \(f(L)/L\) is normally a monotonically decreasing function. Therefore, \(k_m\) is the channel \(k\) that corresponds to the minimum \(N_k\).

This implies that in order to determine the best channel to switch to in the next period, \(k_m\) each of the AP just has to determine the channel that has the least number of associated neighbour nodes. This result is also intuitive since it concurs with the expectation that one should use the channel that has the least number of nodes already using it.

This result has a very important implication and application. In a practical scenario, the throughput function \(f(L)\) may not be known or even impossible to estimate because the exact function depends on many factors including the exact distribution of the packet payload and data rates used by each individual node. Therefore, a simplified algorithm that is not dependent on the knowledge of the throughput function is very practical and useful. Furthermore, this simplified implementation is also efficient, i.e. it reduces the algorithm by 2 multiplicative operations per AP per updating period.

Though throughput function \(f(L)\) are generally strictly monotonically decreasing, it is worth noting that there are some exceptions found in practice. The natural question that arises now is whether the proposed simplified algorithm is still able to match the performance of the original algorithm under these circumstances. In order to determine this, let us consider 2 practical throughput functions obtained for IEEE 802.11b and IEEE 802.11a standard specifications based on the basic access mode of the medium access control (MAC). IEEE 802.11b uses direct sequence spread spectrum (DSSS) while IEEE 802.11a employs Orthogonal Frequency Division Multiplexing (OFDM) physical layer. The throughput functions \(f(L)\) is derived from the analytical model given in [4, 5]. Obtaining the throughput function \(f(L)\) is non-trivial. For each point in the function, the corresponding throughput can be found by solving two nonlinear equations by numerical methods. A typical set of parameters are shown in Table 1 and the corresponding throughput functions are plotted in Fig. 2, which is generated using the throughput model in [4, 5].
TABLE 1
IEEE 802.11b DSSS AND IEEE 802.11a OFDM SPECIFICATIONS AND ADDITIONAL PARAMETERS

<table>
<thead>
<tr>
<th></th>
<th>IEEE 802.11b DSSS</th>
<th>IEEE 802.11a OFDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Payload</td>
<td>1024 bytes</td>
<td>1024 bytes</td>
</tr>
<tr>
<td>MAC Header</td>
<td>224 bits</td>
<td>224 bits</td>
</tr>
<tr>
<td>PHY Header</td>
<td>192 µs</td>
<td>20 µs</td>
</tr>
<tr>
<td>DATA</td>
<td>224 bits + packet payload / data rate + PHY Header</td>
<td>246 bits + packet payload / 4 x data rate + PHY Header</td>
</tr>
<tr>
<td>ACK</td>
<td>112 bits / control rate + PHY Header</td>
<td>134 bits / 4 x control rate + PHY Header</td>
</tr>
<tr>
<td>Slot Time</td>
<td>20 µs</td>
<td>9 µs</td>
</tr>
<tr>
<td>SIFS</td>
<td>10 µs</td>
<td>16 µs</td>
</tr>
<tr>
<td>DIFS</td>
<td>50 µs</td>
<td>34 µs</td>
</tr>
<tr>
<td>Data Rate</td>
<td>11 Mbps</td>
<td>54 Mbps</td>
</tr>
<tr>
<td>Control Rate</td>
<td>1 Mbps</td>
<td>24 Mbps</td>
</tr>
<tr>
<td>CWmin</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>CWmax</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Propagation Delay</td>
<td>1 µs</td>
<td>1 µs</td>
</tr>
<tr>
<td>ACK Timeout</td>
<td>320 µs</td>
<td>50 µs</td>
</tr>
</tbody>
</table>

Referring to Fig. 2, we note that the throughput function for 802.11a is strictly monotonically decreasing. However, the corresponding throughput function for 802.11b is not strictly monotonically decreasing. We note that the function increases when the number of nodes is very small, i.e. when there are only 2 to 3 competing nodes. When there are very few nodes, the channel may not be used optimally due to wastage in empty slots during the backoff operation. This behaviour is caused by the setting of the CWmin which is the initial size of the contention window.

Thankfully, most practical systems are designed such that the maximum throughput occurs when the number of nodes is around 5 or less. It is important to note also that even in this increasing part of the throughput function, Eq. (5) still holds because composite function $f(N_A + N_k)$ is still monotonically decreasing. Simulations have confirmed that the simplified algorithm indeed gives the same performance as the original algorithm under both throughput functions.

B. Asynchronous Scheme

We define a synchronous scheme (e.g. the one described in Section II) as a scheme where the algorithm and the channel switching (when applicable) are carried out simultaneously by all the APs involved. In other words, there is a predefined period where the information about each of the neighboring APs’ load and current channel is obtained. At the end of every period, all the APs will simultaneously carry out the algorithm and channel switching. This process is then repeated for every period.

The assumption of having all the APs perform the algorithm and switching simultaneously is too strong especially in a distributed scenario. One of the major obstacles will be the inability to disseminate synchronizing information to a network of arbitrary and unknown size in a timely manner. Apart from this, the amount of overhead will be prohibitive especially in larger networks. We also noticed that setting the switching probability to $p = 0.5$ instead of having all the APs always switch into the best channel increases the convergence time for the algorithm. This was only made necessary due to the simultaneous or synchronous switching of all the APs.

We propose herein an asynchronous scheme, which allows each of the APs to carry out the algorithm and channel switching (when applicable) asynchronously. Specifically, in the predefined period, each AP chooses a random time at which it will carry out the algorithm and channel switching. The algorithm is carried out with the APs’ current information about its neighboring APs’ load and their current channels.

In every period, each AP would have a chance to carry out the algorithm and channel switching. Immediately after performing the algorithm and channel switching if necessary, the AP will broadcast the latest number of associated nodes and the new channel it is using. Alternatively, the neighboring APs may gather this information through its own or its associated nodes’ measurements (e.g. IEEE 802.11k [6]). In this way, all the neighboring APs will always have the most updated information to use with the algorithm. Therefore, each AP can always switch into the best channel, i.e. $p = 1.0$, without inducing the oscillating effect and reach convergence significantly faster than the corresponding synchronous scheme.

Since the proposed work improves the convergence time and not the total throughput in static environments, we will show the improvement in total throughput performance in dynamic environments in the simulation.
C. Further Enhancements

In this section, we present further enhancements to the simplified algorithm described in Section III-A. We noticed that one situation which needs to be addressed further is when multiple channels are having the same minimum number of neighbour nodes in Eq. (5). We define these channels as “equal candidate channels” for $k_m$ in the set $E$,

$$E = \left\{ \arg \min_{k \in C} N_k \right\}.$$  \hspace{1cm} (6)

Now, if $|E| = 1$, then $k_m$ is unique, which is equivalent to Eq. (5). In cases where $|E| > 1$, we have to establish rules for selecting $k_m$.

One option is to select the lowest numbered channel as $k_m$, i.e.

$$k_m = \min(E).$$  \hspace{1cm} (7)

However, this leads to longer convergence time and the skewing of higher probabilities to lower numbered channels.

As a result, we propose the following rules to further enhance the algorithm.

1) Stay on the current channel if it is one of the equal candidate channels: if $k_{m,j-1} \in E$, then

$$k_m = k_{m,j-1},$$  \hspace{1cm} (8)

where $k_{m,j-1}$ is the channel that the AP is currently occupying which is also the channel that gives the maximum throughput for the previous predefined period of the algorithm.

2) Otherwise, select $k_m$ from $E$ on a random basis.

The proposed enhancement reduces the number of iterations and channel switches required for convergence. It also makes each equal candidate channel equally probable when averaged over all the APs.

IV. SIMULATION RESULTS

We used a 10x10 grid of 100 APs for the simulations. Each of the APs will interfere with each of the 8 neighbouring APs when using the same channel. The number of nodes associated with each AP is randomly selected from 1 to $AP_{\text{max}} = 10$. Unless mentioned otherwise, the number of channels for 802.11b and 802.11a are $K = 3$ and $K = 12$ respectively. All results are obtained using the average of 1000 independent realizations with randomly generated channel and number of associated nodes.

A. Simplified Algorithm

Fig. 3 shows the average throughput against the number of iterations for both the original algorithm and simplified algorithm described in Section III-A. It is clear that the simplified algorithm gives the same performance as the original algorithm without requiring any knowledge about the throughput function but with reduced computational complexity. The results are consistent for both the 802.11b and 802.11a throughput functions which show that the simplified algorithm is still valid for non-strictly monotonically decreasing functions found in practical systems.

B. Asynchronous Scheme

Fig. 4 shows the average throughput against the number of iterations for both the synchronous scheme with the optimal $p = 0.5$ and asynchronous scheme with $p = 1.0$. The asynchronous scheme clearly outperforms the synchronous scheme with a significant reduction in convergence time for both 802.11b and 802.11a.

Table 2 shows the average number of iterations for both schemes to reach convergence. The asynchronous scheme shows up to 77.39% and 52.38% reduction in convergence time and channel switches required to reach convergence. The reduction for 802.11a is lesser because having 12 channels represent a much less “dense” scenario compared to 802.11b with only 3 channels.

In order to simulate the dynamics of the network, we modeled the loads of all the APs to change periodically with the same period as the channel switching. The changes in the loads are normally distributed with mean = 0 and standard derivation $\sigma$ being a percentage of $AP_{\text{max}}$. A higher standard
deviation reflects a network that is more dynamic. Furthermore, the number of nodes associated with each AP is lower and upper bounded by 1 and $AP_{\text{max}}$ respectively. We only show the performance for the denser scenario i.e. based on 802.11b.

Fig. 5 shows the average throughput for both schemes in a dynamic environment where the standard deviation is 100% of $AP_{\text{max}}$. As expected, asynchronous scheme outperforms synchronous scheme. As shown in Table 3, the improvements increases in more dynamic scenarios and is as much as 12.73% when the standard deviation is 100% of $AP_{\text{max}}$. This is due to the shorter convergence time of the asynchronous scheme which allows the algorithm to track the changes in the network more closely than the synchronous scheme. Although not shown due to space constraint, an impressive gain can still be achieved with 802.11a.

C. Further Enhancements

Performance results of further enhancements described in Section III-C is presented in Table 4. The enhancements proposed reduce the convergence time and channel switches significantly i.e. reduction of up to 52.76% and 71.61% respectively. We note that the reduction for 802.11a is significantly more because having 12 channels increases the probability of having equal candidate channels, which the enhancements capitalizes. An interesting feature of the enhanced algorithm is that it only requires 1 iteration to converge when the number of channels is more than the number of interfering neighbours as in the case of 802.11a.

V. CONCLUSION

We have presented an enhanced asynchronous distributed and dynamic channel assignment scheme that is simple, does not require knowledge of the throughput function and allows each AP to asynchronously switch to the best new channel. Simulation results show that our proposed scheme converges much faster and requires significantly less channel switches than the synchronous scheme. We have also shown its superior throughput performance in dynamic scenarios.

REFERENCES


