This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2010 proceedings

A Distributed Channel Assignment Algorithm for Uncoordinated WLANs

Xiaonan Yue Chi-Fai Wong S.-H. Gary Chan
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{csjohn, mwong, gchan}@cse.ust.hk

***Abstract*— IEEE 802.11 WLANs are becoming more and more popular in homes and urban areas. As opposed to traditional WLANs, the access points (APs) in these networks are often deployed by network non-specialists in an uncoordinated manner, leading to unplanned topology, interference and unsatisfactory throughput performance.**

We consider in this paper a distributed channel assignment algorithm for uncoordinated WLANs, where APs can self-configure their operating channels to minimize interference. We propose an efficient, simple and distributed algorithm termed CACAO (Client-Assisted Channel Assignment Optimization). In CACAO, an AP makes use of the traffic information fed back by its clients to make channel assignment decision. This leads to better knowledge on network environment and better channel assignment decision at the APs. We conduct extensive simulation study and comparisons using Network Simulator 2 (NS2). Our results show that CACAO out-performs other traditional and recent schemes in terms of throughput with similar level of fairness. Furthermore, it converges quite fast to reduce interference to a low level.

I. Introduction

In recent years, we have witnessed increasing penetration of wireless broadband Internet to our everyday life, mainly due to the availability of affordable Wi-Fi capable consumer products such as laptops, PDAs and gaming devices. In order to allow these devices to exchange data through the Internet, more and more people are setting up WLANs in their homes, common areas or wherever wireless connection is needed.

These WLANs share some common features such as *uncoordinated*, small in size, independently owned and managed. Though IEEE 802.11b/g defines a total of 11 overlapping channels, only 3 non-overlapping channels (namely channels 1, 6 and 11) can be used simultaneously without causing interference. With the limited number of non-overlapping channels, channel assignment becomes a critical performance factor for uncoordinated WLANs. As compared with traditional WLAN deployment (e.g, enterprise or campus environments), uncoordinated WLAN deployment (e.g, home environment) presents the following challenges in channel assignment algorithm design: (i) Uncoordinated WLANs are usually set up by novice users who very likely expect the devices to be plug-and-play, and (ii) The APs of different WLANs should operate independently, without any communication with each other.

This work was supported, in part, by the Hong Kong Innovation Technology Fund (ITS/097/09FP).

In this paper, we propose and study a novel distributed channel assignment scheme for uncoordinated WLANs. The APs auto-configure their channels depending on their local traffic information. Our approach, termed Client-Assisted Channel Assignment Optimization (CACAO), makes use of client feedback to perform channel assignment. Such feedback may be obtained from the proposed IEEE 802.11k standard for radio resource management [1], which defines a series of measurement requests and statistical reports between the AP and clients. With the feedback, APs are able to get a better view of network congestion, and hence can dynamically configure themselves to operate on the “best” channel to reduce interference among BSSs. Our approach also overcomes traffic distribution problem and addresses hidden interference problem.

We implement CACAO using NS-2 (version 2.30) and conduct extensive simulations. We compare it with the traditional LCCS, and state-of-the-art MAXChop [2], and Hminmax [3]. Our simulation shows that networks using CACAO achieve much lower interference and higher throughput as compared with the traditional and recent scheme. CACAO also achieves much better throughput without sacrificing fairness.

The rest of the paper is organized as follows. In Section II, we survey related works and present some of the weaknesses of traditional channel assignment approaches. The CACAO algorithm and its implementation issues are presented in Section III. In Section IV, we discuss illustrative simulation results and comparisons. We conclude in Section V.

II. Related Work

Much work has been done with primary objective to reduce interference or increase the total throughput of the system. AP placement schemes optimizing power level and network throughput have been proposed in [4]. Measurement-based WLAN deployment schemes have been proposed in [5], [6]. As opposed to our work, these studies assume network administrators conducting site survey and propagation modeling before network deployment. Athanasiou et al. propose LAC [7], a distributed scheme making use of load information. However, this scheme cannot dynamically adjust channel assignment according to traffic pattern. Our approach is a dynamic one where we assign channels depending on the *existing* network conditions.

978-1-4244-5176-0/10/$26.00 ©2010 IEEE

Mishra et al. propose a dynamic channel assignment algorithm called CFAssign-RaC to achieve load-balancing based on a "conflict set coloring" formulation [8]. Ahmed et al. proposed an algorithm using successive refinement to solve a joint channel assignment and power control problem [9]. Kauffmann et al. propose a measurement-based self-organization approach for channel assignment [10]. As opposed to our work, all of them focus on networks where all the participating devices belong to the same enterprise under the same network administration. Later proposed traffic-aware approaches in [11] and [12] make further advance by considering traffic pattern. However, their algorithms are still done in a centralized way, which means these works are not so applicable to uncoordinated WLANs we study in this paper. Wong et al. propose PACA [13], a fully distributed channel assignment algorithm making use of local information. Yang et al. propose FLEX [14], a distributed architecture for APs to dynamically access spectrum according to user demands. However, these approaches require communication among APs, which is not possible for uncoordinated WLANs we consider here.

In [2], the authors propose a distributed algorithm called MAXChop, which addresses channel assignment based on standard graph coloring formulation. However, periodical communication among APs are needed for MAXChop to function, and it does not consider dynamic traffic pattern. Arbaugh et al. propose Hminmax [3] and formulate the channel assignment problem as a weighted coloring graph problem. Their approach is based on the interference experienced by clients. CACAO differs by taken into account the real traffic load of *both* APs and clients, which leads to better interference mitigation and hence better performance. We will compare these two recent schemes (MAXChop and Hminmax) with CACAO in this work.

III. CACAO Channel Assignment

In this section, we propose CACAO, a distributed algorithm to achieve high throughput to all BSSs in Section III.A. We then discuss its implementation issues in Section III.B.

A. CACAO Algorithm

The distributed CACAO algorithm utilizes information gathered by APs and clients on interference conditions to minimize the local objective function by switching to a channel that has least expected interference. As mentioned, the algorithm is very simple, and does not require any communication among neighboring APs. It adaptively settles to some minimal interference among the BSSs depending on network traffic.

We illustrate CACAO algorithm for ap_i in Algorithm 1. Firstly, every AP runs the initialization routine when it boots up. During boot-up period, each AP randomly assigns itself to a channel chosen from the k non-overlapping channels, because there are no previous traffic information gathered to decide on channel assignment. Then, an AP periodically runs the optimization routine. It computes the sum of expected interference level with regard to each nearby BSS for each channel to be used for the next time interval. Due to the fact that many online applications (like video streaming) have rather steady data transfer rate, the amount of traffic observed for the current time interval is a good estimation for the traffic in the next time interval. After the calculation and comparison of expected interference level for each channel, the AP chooses the channel that yields the least total interference and switch to that channel. At every time period, each AP independently chooses the channel assignment to minimize the objective function locally. Clearly, the whole network does not need any synchronization.

Algorithm 1 CACAO Algorithm

CACAO(ap_i)
1. Initialization - Initial Assignment
    $ap_i.c \leftarrow rand(k)$
2. Optimization - Repeated for each AP
    2.a $GatherStatistic()$
    2.b $c_t = ComputeInterfere()$
    2.c $SwitchTo(c_t)$

$GatherStatistic()$ is a procedure that is used to gather statistics from clients. It returns the traffic information collected by clients. Then the AP performs computation and comparison to decide which channel it will use for the next time period by procedure $ComputeInterfere()$. This routine computes the expected value of interference level that the entire BSS will likely experience for the next time interval. The channel with the least interfered traffic will be chosen. Finally the $SwitchTo()$ routine assigns the AP and its associated clients the channel.

B. Implementation Issues

A client sequentially switches to a channel to measure and collect the channel utilization information (channel utilization query process). We consider only non-overlapping channels. While the node "snoops" another channel, the AP buffers packets for it. This can be done by using the Power Saving Mode (PSM) available in IEEE 802.11 [15]. As mentioned, the query process can be done by any protocol, such as the currently discussed 802.11k (Radio Resource Management specification for interoperability).

After a client finishes its channel utilization query process, it switches back to the original channel and sends the report to its AP. We show in Table I the major elements of statistic report used in CACAO. The report mainly contains four fields, $Channel$, $LoadObserved$, $NumofNeighbor$ and $MyLoad$. $Channel$ (CH) is the reporting channel number. $LoadObserved$ (LO) stores the total amount of traffic observed (in kbits/s) from other networks in the recorded interval. $NumofNeighbor$ (NN) stores the number of nodes (both clients and APs included) contributing to $LoadObserved$. $MyLoad$ (ML) stores the client's own out-going traffic load in the time interval.

Regarding traffic measurement, some work has made convincing argument that client side approach will be able to probe and get good information on the wireless environment [8]. In CACAO, interference reports at client side are created by ideally idle clients. They enter a interference measurement

TABLE I

MAJOR ELEMENTS OF STATISTIC REPORT USED IN CACAO.

Channel (CH)	Load Overserved (LO)	Num of Neighbor (NN)	My Load (ML)

process, collect information, generate statistics and send the report back to APs for further calculation. A client that performs such action randomly picks a channel and switches to it. At the same time AP needs to buffer packets that are designated for the specific client. There are two conditions when the client needs to switch back to the original channel, (i) if the client needs to send data to the AP and (ii) if the predefined interval of interference measurement ends. While the client is operating on another channel, it only listens and records the amount of the traffic during the specified time interval. There are two things that we need to pay attention. (i) The interval of a client listening to another channel needs to be reasonably short, mainly because the AP has limited memory space and can not buffer large content for the client; (ii) There may be situations where all the clients are always actively communicating using the channel. This means that the previous mentioned random channel measurement approach by idle client can not be applied. To ensure that there is always some clients doing channel measurement, we assign randomly selected clients to switch to other channels for a small fixed period of time. AP only measures its operating channel for efficiency.

IV. ILLUSTRATIVE SIMULATION RESULTS

In this section, we evaluate the proposed CACAO using packet-level simulations with NS-2 network simulator. We compare the performance of CACAO with three other channel assignment algorithms namely, LCCS which is commonly implemented on APs, MAXChop algorithm and Hminmax algorithm.

A. Simulation Environment

Unless otherwise stated, we use the values specified in Table II in our simulation. First we generate topologies by random AP placement inside a rectangle bounding box. For each AP, the number of associated clients is randomly drawn from 1 to some maximum value. After the number of associated clients is determined, we randomly put them inside the coverage area of their associated AP. For simulations where UDP traffic is used, we generate CBR traffic with constant data packet size using the value set in the specification table. For simulations with TCP protocol, FTP application is created, and no data rate is specified. RTS/CTS is turned off to better simulate the real environment. This is also the default setting in most commercial APs.

B. Illustrative Results

We compare the average throughput per flow of CACAO, Hminmax, MAXChop and LCCS in Figure 1. For all algorithms, the average user throughput decreases with the increase of the number of independent WLANs due to the increase of interference and contention. When the number of independent WLANs is small, the throughput achieved by all algorithms are roughly the same and reaches the maximum throughput. As the number of WLANs increases, the performance of LCCS decreases dramatically due to the fact that it uses only AP side static channel assignment. The performance of Hminmax and MAXChop are much better compare to LCCS. However, CACAO still outperforms those two by considering client side traffic and trying to minimize the total interference. Figure 1 shows that CACAO channel assignment algorithm can achieve higher throughput for both UDP and TCP connections, compare to traditional approaches and the state of the art algorithms. It also shows that the traffic information gathered by clients is helpful for the AP to select a channel that can avoid interference between independent WLANs.

TABLE II

SIMULATION SETTINGS.

Medium Access Protocol	IEEE 802.11b
Radio Propagation Model	Shadowing
Link Basic Rate	2 Mbps
Link Data Rate	11 Mbps
RTS/CTS	OFF
Bounding Box Size	1500m by 1500m
Maximum Clients for an AP	8
Client Association Range	100m by 100m
Transmission Range	250m
Interference Range	550m
Simulation Duration	45s
Packet Size	1000 bytes
Number of Simulation Trails	20

In the following, we use a sample topology setting (12 APs and 63 clients) shown in Figure 2. In this graph every solid dot represents an AP, big circles indicate the transmission range of the AP and the empty dots represent clients. The generation of this topology is mentioned in the previous section.

The distribution of throughput per client of each algorithm is shown in Figure 3. Since LCCS algorithm does not consider traffic pattern and only assigns channels at the very beginning, the distribution of throughput is not normal. There are many clients that experience poor performance while some nodes experience very good performance. On the other hand, MAXChop algorithm and CACAO algorithm are able to assign channels dynamically so the performance is more fair for all the clients. Since MAXChop algorithm focuses more on creating a fair environment for all WLANs by using channel hopping, the throughput of nodes appear more normally distributed and more clients' throughput are close to the average throughput achieved in the whole system. The throughput distribution of Hminmax algorithm is similar to CACAO because they all use a similar objective function. So we only show results of LCCS, MAXChop and CACAO for comparison here.

Figure 4 shows the evolution of cost function in the network over time. A setting containing TCP connections and FTP applications is used in this simulation. The whole system is dormant at the beginning. FTP applications start randomly during time slot 1. CACAO algorithm is implemented in a decentralized and desynchronized manner so that APs make

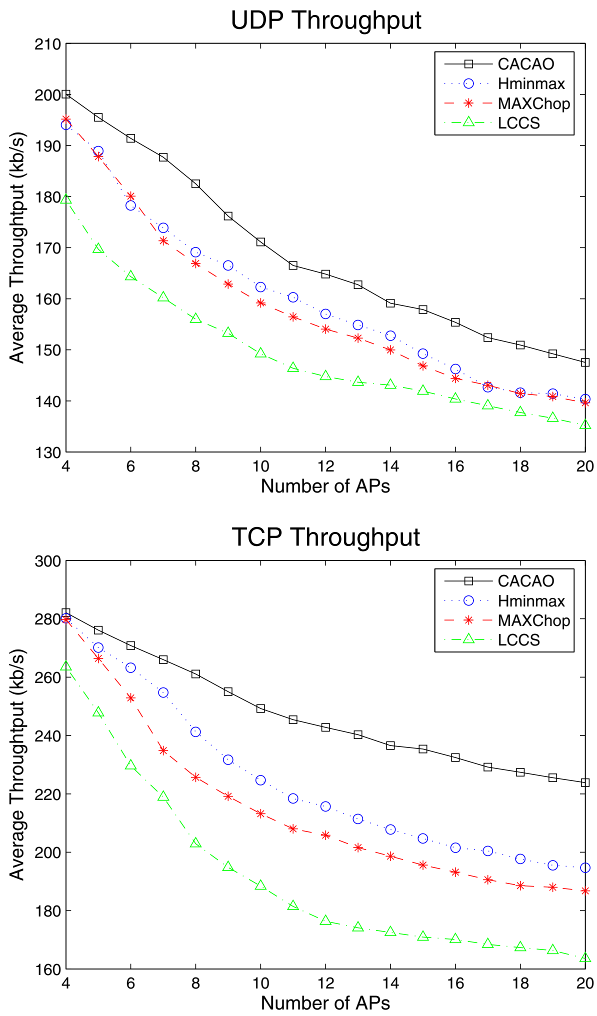

Fig. 1. Average throughput per flow against number of APs for different schemes.

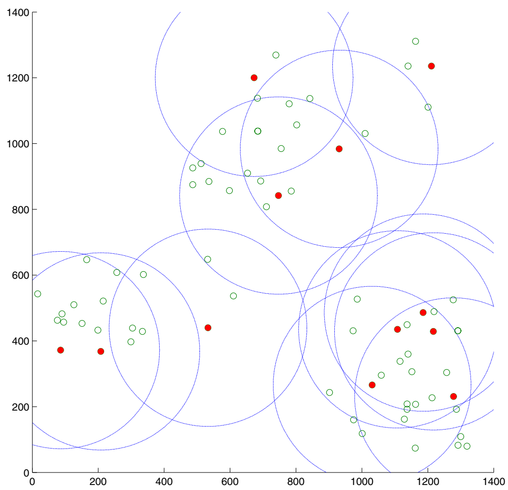

Fig. 2. An example of the topology under study.

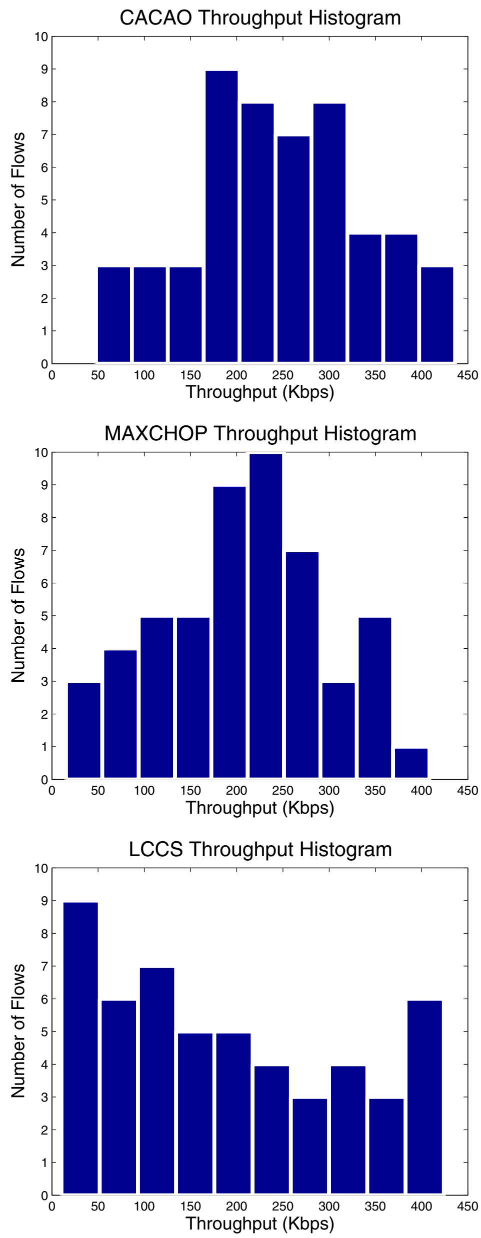

Fig. 3. Throughput distribution for different schemes.

channel switching decisions at different time. As shown in the figure, the objective function value increases dramatically because of the sudden traffic pattern change. During time slot 2, some APs start to change their operating channels to avoid interference, as a result there are some small drops of the objective function value. However, the traffic information collected during time slot 1 is not complete and not accurate because some of the traffic starts in the middle of this time slot. So the best choice of channel can not be decided until time slot 3. Since all the APs have gathered correct traffic information during time slot 2, they can assign themselves good channels that avoid a lot of interference at time slot 3. From Figure 4 we can see that every channel switching at time slot 3 results in a big drop of the objective function value. After time slot 3, objective function value does not change

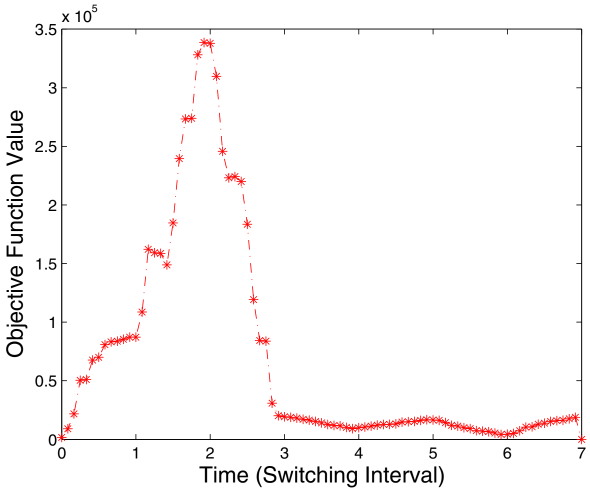

Fig. 4. Objective function value over time.

much and channel switching does not happen as often as in time slot 2 and 3. CACAO algorithm is able to adjust to the changes and reduce the objective function value. Also, after the traffic pattern changes, the objective function value does not fluctuate a lot and maintains at a relatively low level.

V. Conclusion

In recent years, WLANs have become more and more popular due to the pervasiveness of wireless devices including cell phones, laptops etc. Many of these WLANs are independently set up by novice users. Since these WLANs are set up by uncoordinated and inexperienced users, developing an automatic and yet efficient channel switching algorithm becomes very important to these WLANs.

In this paper, we propose CACAO, a distributed algorithm that tries to minimize interference level in the network. CACAO overcomes the weaknesses of traditional approaches. It is simple, effective, completely distributed and hence scalable. APs using CACAO algorithm gather channel and interference condition with other WLANs with the help of associated clients. This information helps the AP to make better decision on channel assignment. Through client reports of channel information, the APs can assign channel with minimal interference.

Using NS, we implement LCCS, recently developed MAX-Chop algorithm and Hminmax algorithm and our algorithm CACAO and run extensive simulations to collect data. CACAO improves network throughput significantly with little compromise in fairness. It is also adaptive to traffic condition to keep interference at a low level.

References

[1] IEEE 802.11k - Radio Resource Measurement Enhancements, http://grouper.ieee.org/groups/802/11/Reports/tgk_update.htm.

[2] A. Mishra, V. Shrivastava, D. Agrawal, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," in *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2006.

[3] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for WLANs," *ACM SIGMOBILE MC2R*, 2005.

[4] Y. Lee, K. Kim, and Y. Choi, "Optimization of AP placement and channel assignment in wireless LANs," in *LCN '02: Proceedings of the 27th Annual IEEE Conference on Local Computer Networks*. Washington, DC, USA: IEEE Computer Society, 2002, p. 0831.

[5] I. Broustis, K. Papagiannaki, S. V. Krishnamurthy, M. Faloutsos, and V. Mhatre, "Mdg: measurement-driven guidelines for 802.11 wlan design," in *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, 2007.

[6] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill, "Designing high performance enterprise wi-fi networks," in *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, 2008.

[7] A. George, B. Ioannis, and K. L. T. Thanasis, "Lac: Load-aware channel selection in 802.11 wlans," in *IEEE PIMRC*, 2008.

[8] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, "A client-driven approach for channel management in wireless LANs," in *Proceedings of IEEE INFOCOM '06*, April 2006.

[9] N. Ahmed and S. Keshav, "A successive refinement approach to wireless infrastructure network deployment," *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 1, pp. 511–519, 0-0 2006.

[10] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot, "Measurement-based self organization of interfering 802.11 wireless access networks," in *Proceedings of IEEE INFOCOM '07*, May 2007.

[11] E. Rozner, Y. Mehta, A. Akella, and L. Qiu, "Traffic-aware channel assignment in wireless LANs," in *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 2. New York, NY, USA: ACM, 2007, pp. 43–44.

[12] ——, "Traffic-aware channel assignment in enterprise wireless networks," in *In Proc. of ICNP*, 2007.

[13] C.-F. Wong, S.-H. Chan, and J.-C. Chen, "PACA: Peer-assisted channel assignment for home wireless LANs," in *Proc. IEEE GLOBECOM'06*, 2006, pp. MMC05–6.

[14] L. Yang, L. Cao, H. Zheng, and E. Belding, "Traffic-aware dynamic spectrum access," in *WICON '08: Proceedings of the 4th Annual International Conference on Wireless Internet*, 2008.

[15] R. Chandra and P. Bahl, "Multinet: connecting to multiple IEEE 802.11 networks using a single wireless card," in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, March 2004.