

Adaptive Channel Allocation for IEEE 802.11 Wireless LAN

Ye Zhao and Kin K. Leung

Electrical & Electronic Engineering Department, Imperial College, London, United Kingdom

{ye.zhao | kin.leung}@imperial.ac.uk

Abstract – Today, IEEE 802.11 wireless LANs (WLANs) are widely deployed throughout the world. It is known that static channel assignments are not sufficient for these networks due to traffic variations. However, in order to maintain compatibility with the existing protocol standards, it is not possible to employ dynamic channel allocation methods to handle traffic-load fluctuations. Rather, adaptive assignment schemes seem to be appropriate.

In this paper, we enhance the MinMax algorithm for static channel assignment for 802.11 networks proposed earlier. Based on the enhanced MinMax algorithm, we propose here a centralized adaptive channel allocation scheme for 802.11 networks. Computer simulation using the ns2 software reveals that our proposed adaptive scheme achieves a noticeable improvement of data throughput over fixed assignment algorithms, while only a relatively small number of APs require channel changes as a means to reduce service interruption to users under the existing protocol standards.

Keyword: IEEE 802.11, channel assignment, channel utilization, interference.

1. Introduction

Today, wireless LANs (WLANs) are widely deployed all over the world to meet the growing demand for wireless data services. The IEEE 802.11 technology is particularly attractive due to its maturity and low cost. The 802.11 standard has evolved into three high-speed versions of the specification: 802.11a, b and g [1-3]. Among them, 802.11b networks were very popular and they are replaced by 802.11g recently due to improved data rate.

As the 802.11 access points (APs) are deployed everywhere, the coverage areas of APs are expected to overlap with each other. As a result, the radio interference between adjacent co-channel APs can cause their throughput performance to degrade. This is particularly so because there are only 3 non-overlapped channels available in the 2.4 GHz ISM band used by the 802.11 networks. Moreover, the limited number of available channels makes this interference degradation become unavoidable in dense WLAN environments. In order to prevent significant performance degradation in

the network, limited frequency channels should be assigned to APs in an appropriate and efficient manner.

It is well recognized that static channel assignments do not provide the best performance of wireless networks, because traffic load (thus interference) in a network usually varies in time. When comparing with static assignments, dynamic channel allocation typically yields better performance in terms of interference, throughput and delay at the expense of added complexity in the control mechanisms. Unfortunately, in the context of IEEE 802.11 WLAN, the existing specifications do not include any protocols or control mechanisms to support dynamic channel assignment. Although such may be considered in the future versions of the standards, it will be difficult to maintain backward compatibility with existing protocol standards.

Consider an 802.11 WLAN in an office environment. When colleagues are gathering for a meeting, network traffic at the conference room increases and maintains high until the meeting ends. Although such traffic variation has a time scale on the order of minutes, static channel assignment clearly cannot provide the optimal use of limited radio frequencies. On the other hand, some form of adaptive channel allocations where the channel changes at a rate matching the traffic variation is desirable. The same comment also applies to other WLAN operation scenarios including outdoor applications. Furthermore, in contrast to the protocol requirements for dynamic channel allocations at a time scale of msec, if designed properly, adaptive channel assignments do not require protocol changes to the existing standards. Of course, to make the adaptive algorithms “compatible” with the existing standards, it is highly desirable for the algorithms to re-assign an only small number of channels based on the current channel assignment, when appropriate changes of traffic load (or interference) are detected. These have been the motivation for our work in this paper.

We observe that WLANs in typical operating environments are often managed by a network management system. So, it is natural to add the capability of adaptive channel allocations to the centralized system. Therefore, we shall investigate a centralized, adaptive channel assignment algorithm. As a high-level view, APs periodically measure and estimate their traffic load and report the estimates to the management system. In turn, the latter instructs certain APs to change to new radio frequencies at an appropriate time, according to the adaptive algorithm. It is expected that the rate of channel changes is low, a secondary objective of the adaptive scheme is to reduce the need of channel changes, while attempting to “optimize” the throughput performance. This way,

service interruption for existing users can be minimized. Such is particularly needed because the 802.11 protocols do not support efficient change of radio frequencies.

The rest of this paper is organized as follows. In Section 2, we review existing work on channel assignment for 802.11 networks. In section 3, we first propose ways to improve the static algorithm proposed in [8]. Then, it is extended into an adaptive assignment algorithm. In section 4, we study the performance of the algorithm by computer simulations using ns2. Finally, we present our conclusion in section 5.

2. Existing work

The channel allocation for 802.11 networks has been an active area of research. Riihijarvi, Petrova, and Mahonen [5, 6] propose using classical graph coloring algorithms to assign frequency channels for WLANs. Additionally, a dynamic allocation scheme based on neural network models is proposed by Luo and Shankaranarayanan [7] to improve throughput in dense WLAN environments. Besides these algorithms, Leung and Kim [8] propose a fixed channel allocation scheme by using a heuristic algorithm to minimize the maximum effective channel utilization of APs in the network. We refer to it as the MinMax algorithm in this paper. Since the latter algorithm is the basis of our new adaptive method, we first provide an overview of the MinMax algorithm, which will help us identify its shortcomings in the next section.

By the carrier-sensing-multiple-access (CSMA) protocol, an AP with traffic ready to transmit first determines if the assigned channel is busy or idle. Since APs usually have much more traffic to send than individual terminals when the network operates in the infrastructure mode, let us consider transmissions only by APs here. Clearly the channel busy status can be due to a single transmitting AP or a group of multiple APs transmitting simultaneously. So interferers for each AP can be classified into different groups. Let $C_i(1)$ denote a set of interfering APs, called *class-1 interferers* for AP i , where transmission by any one AP in the set can cause enough interference for AP i to detect channel busy. Likewise, let $C_i(2)$ be a set of pairs of two interfering APs where simultaneous transmission by any pair of APs in the set can cause enough interference for the AP i to sense channel busy. Pairs of APs in $C_i(2)$ are called *class-2 interferers*. Higher classes of interferers can be classified in a similar way. However, because the probability of having simultaneous transmission at more than two interfering APs is much smaller compared to class-1 and class-2 interferers, they can be ignored.

Assume a network with M access points, indexed from 1 to M . Let ρ_i be the offered traffic load for AP i in terms of channel utilization without interference from any source. Since the CSMA protocol prohibits APs from transmitting when the channel is sensed busy, we define the *effective channel utilization*

U_i as the fraction of time at which the channel can be sensed busy or is used for transmission by AP i , and ρ_i is channel utilization without interference from any source. That is,

$$U_i = \rho_i + \sum_{k=1}^N X_{ik} \left[\sum_{j \in C_i(1)} \rho_j X_{jk} + \sum_{(m,n) \in C_i(2)} \rho_m \rho_n X_{mk} X_{nk} \right] \quad (1)$$

where N is the total number of channels available. Further, $X_{ij} = 1$ if AP i is assigned with channel j and 0 otherwise. This definition is valid under constraints that

$$U_i < 1 \text{ and } \rho_i < 1 \quad (2)$$

for all AP $i = 1$ to M .

The objective function of the MinMax algorithm is to minimize the utilization at the most stressed bottleneck AP:

$$\text{Minimize } \text{Max}\{U_1, U_2, \dots, U_M\} \quad (3)$$

over the assignment indicator $\{X_{ij}\}$. The problem has been proved to be NP-complete, and the heuristic algorithm for allocating channels was proposed as follows:

1. Generate a random, initial channel assignment for the network, which is treated as the best assignment obtained so far. Let the maximum effective channel utilization for the assignment be denoted by V .
2. Based on the best assignment, identify the AP (say i) with the highest effective channel utilization. In case of tie, one such AP i is chosen randomly as the "bottleneck".
3. For the bottleneck AP i , identify its current assigned channel, say k . For each available channel n from 1 to N with $n \neq k$ and each co-channel AP (say j) in $C_i(1)$, temporarily modify the channel assignment by re-assigning only AP j with channel n . Based on (1), re-compute the maximum effective channel utilization, denoted by W_{jn} for the new assignment. After completing such testing for all such n and j , let W be the minimum among all the W_{jn} 's.
4. Compare W with V and perform the following:
 - a. If $W < V$, then replace V by W and record the associated new assignment as the new best solution. Continue with Step 2.
 - b. If $W = V$, then with a pre-specified probability δ , replace V by W and record the new assignment as the best solution. Continue with Step 2.
 - c. If $W > V$, a local optimum has been reached. Continue with Step 5.
5. Repeat Step 1 to 4 with a number of random, initial assignments. The final solution is chosen to be the best among the local suboptimal assignments.

6. Test if the effective channel utilizations of all APs are less than 1. If so the final assignment is feasible. Otherwise, it is considered that no feasible solution exists for the network under consideration.

In essence, considering the offered traffic load and interference (by detection of channel busy), the MinMax algorithm tends to assign different channels to APs with high traffic load and/or strong mutual interference to each other. The algorithm improves the aggregate throughput significantly in networks with unbalanced traffic distributions. However, the algorithm is static and cannot cope with changes of traffic load. Hence, an adaptive method is proposed as follows.

3. Adaptive scheme

Before presenting our adaptive scheme, we propose ways to enhance the original MinMax algorithm as follows. Then, the enhanced scheme is used to build the adaptive method.

3.1. Algorithm Enhancement

The main part of the MinMax algorithm is the heuristic process from step 2 to step 4 in Section 2. It is responsible for probing for a local optimum assignment by taking greedy steps from a given initial assignment. Our enhancement is achieved by correcting two aspects of deficiency.

The first deficiency of the original algorithm is that step 3 does not attempt to change channel at the bottleneck AP itself, which may be more effective than to change channels for its interferers. This can be illustrated by a simple example where three APs A, B, C along a straight line (Fig. 1). Assume that A and C do not interfere with each other, but both of them interfere with B, and that all APs are using the same channel. So AP B is clearly the bottleneck. The best solution to this problem is to let B change its channel. However, the original algorithm tries different channels at only A and C. To correct this insufficiency, we allow step 3 to test different channels at the bottleneck AP as well.

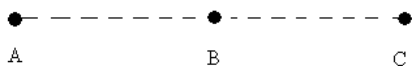


Figure. 1 A simple example illustrates the deficiency of the original MinMax algorithm.

The second deficiency of the original algorithm is its inadequacy for multiple bottlenecks. The comparison in step 3 and step 4 determine that the algorithm finalizes a channel change only when the maximum effective channel utilization among all APs can be reduced by it. (Though step 4b can make channel changes without improvement with a probability δ , it is not enough.) However, when there are multiple bottleneck APs, changing a channel at any one of them may achieve only a local improvement to nearby bottleneck APs while far-away bottlenecks are unaffected. In this case, as the maximum effective channel utilization is not reduced, the original algorithm does not even accept the local improvements. To improve, we let the algorithm to examine all bottlenecks in the network

rather than select one of them randomly and also adjust the comparison bases in step 3 and step 4, to achieve the best local improvement.

We outline the enhanced algorithm (from step 2 to step 4) becomes:

2. Based on current best assignment, compute the maximum effective channel utilization, say V , and record all APs with this highest channel utilization in a “bottleneck list”.
3. For each AP i in the “bottleneck list”, identify its current assigned channel, say k_i . Then for each available channel n from 1 to N with $n \neq k_i$ and for AP i and each co-channel AP (say j) in $C_i(1)$, temporarily modify the channel assignment by reassign only AP i or AP j with channel n . Re-compute the maximum channel utilization W_{in} or W_{jn} for the new assignment where the channel assignment for all APs rather than AP i remains unchanged. After completing such testing for all such i , n and j , let W_{\min} be the minimum among all the W 's.
4. Compare W_{\min} with V and perform the following:
 - a. If $W_{\min} < V$, then record its associated channel assignment as the new best solution. Continue with step 2.
 - b. Otherwise, a local optimum has been reached. (No improvement can be done to any bottleneck in the network.) Continue with Step 5.

By considering possible channel change for the bottleneck AP itself and for all bottleneck APs if multiple of them exist, the modified algorithm can yield better local optimal assignments than the original MinMax algorithm. Numerical examples will be presented in Section 4.

3.2. Adaptive Channel Allocation

Based on this enhanced MinMax algorithm, we devise a centralized adaptive scheme for adapting channel assignment according to the time fluctuation of traffic load at various APs. The algorithm operates as follows:

1. We divide time into equal interval periods, referred to as adaptive period. (For practical considerations of WLANs, we suggest the period to be 1 to 10 minutes.)
2. Every AP in the network constantly monitors its offered traffic load (including that sent by its associated terminals). Towards the end of an adaptive period, each AP also estimates and informs the central controller of the offered traffic load in the next adaptive period.
3. Upon receiving the traffic predictions from all involved APs, the central controller runs the enhanced Minmax algorithm (i.e., step 2 to step 4 in the last section) to reassign channels in the network. New channel assignments are forwarded to the involved APs for use in the next adaptive period.

It is worth noting that the heuristic algorithm attempts to achieve the “maximum” improvement with few channel re-assignments. Such helps reduce as much as possible the service interruption to existing users, especially in light of non-existence of protocol supports for channel change, and communication overhead between the central controller and affected APs to change to new channels.

The new adaptive algorithm requires reasonably accurate predictions of offered traffic load associated with each AP in the next adaptive period. Although traffic load among APs can demonstrate both temporal and spatial dependency, for simplicity, we consider only the time dependency here and treat the offered traffic load at each AP in consecutive periods as an independent, discrete-time series. A method for traffic prediction based on monitoring channel states and packet transmissions at each AP is given in the following.

Since transmission of a single data packet (frame) can involve transmissions of a Request-to-Send (RTS) frame, the Clear-to-Send (CTS) frame, the data frame, and the associated acknowledgment (ACK) frame. Due to protocol timing specified in the 802.11 standard, all these frames are sent in sequence without interruption by other transmissions. Thus, we define time required to transmit all these frames associated with one data packet as the *packet channel time* (see Fig. 2). When the RTS/CTS hand-shake is not used, the packet channel time consists of transmission times for the data packet and ACK, plus the involved idle times defined by the protocol standard. Since packet length is expected to be different for data packets sent or received by an AP, packet channel time is further classified as the *transmitting* and *receiving* (packet) channel time in the AP’s perspective.

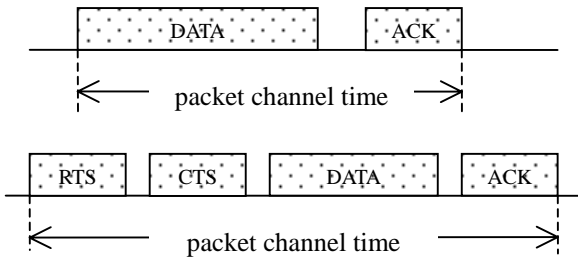


Figure. 2 Packet channel time corresponds to two-way and four-way exchange protocols.

Let $T_i(k)$ and $R_i(k)$ (msec/packet) be the average transmitting and receiving channel time measured by AP i in the k^{th} adaptive period, respectively. In addition, each AP i also monitors the packet sending and receiving rate, $\gamma_i(k)$ and $\mu_i(k)$ (packets/msec) during the k^{th} period.

With these parameters, we adopt two algorithms to predict the average channel time and packet rate for the next $(k+1)^{th}$ period. A simple exponential smoothing [9] is used to predict the transmitting and receiving channel time, $\hat{T}_i(k+1)$ and $\hat{R}_i(k+1)$, in the next

period as

$$\hat{T}_i(k+1) = (1-\alpha) \cdot \hat{T}_i(k) + \alpha \cdot T_i(k) \quad (4)$$

$$\text{and } \hat{R}_i(k+1) = (1-\alpha) \cdot \hat{R}_i(k) + \alpha \cdot R_i(k) \quad (5)$$

where α is chosen to be 0.4 for the best performance based on our extensive numerical experiments. To adequately predict the packet rate under varying traffic-load conditions, a Holt-Winters process [9] is applied to predict the next values of packet rate, $\hat{\gamma}_i(k+1)$ and $\hat{\mu}_i(k+1)$, as

$$\hat{\gamma}_i(k+1) = (1-\beta) \cdot [\hat{\gamma}_i(k) + L_i^\gamma(k)] + \beta \cdot \gamma_i(k) \quad (6)$$

$$\hat{\mu}_i(k+1) = (1-\beta) \cdot [\hat{\mu}_i(k) + L_i^\mu(k)] + \beta \cdot \mu_i(k) \quad (7)$$

Note that the Holt-Winters prediction is a modified exponential smoothing by including a trend component (L). The trend parameters, $L_i^\gamma(k)$ and $L_i^\mu(k)$, are estimated by

$$L_i^\gamma(k+1) = (1-\omega) \cdot L_i^\gamma(k) + \omega \cdot [\hat{\gamma}_i(k+1) - \hat{\gamma}_i(k)] \quad (8)$$

$$L_i^\mu(k+1) = (1-\omega) \cdot L_i^\mu(k) + \omega \cdot [\hat{\mu}_i(k+1) - \hat{\mu}_i(k)] \quad (9)$$

From our extensive experiments, β and ω are set to 0.75 and 0.8, respectively, to yield the best performance. Finally, the offered load of AP i in the $(k+1)^{th}$ period is given by

$$\hat{\rho}_i(k+1) = \hat{\gamma}_i(k+1) \times \hat{T}_i(k+1) + \hat{\mu}_i(k+1) \times \hat{R}_i(k+1) \quad (10)$$

Note that our traffic prediction by (10) is based on monitoring traffic sent by the AP or its terminals on the channel (i.e., the so-called “carried traffic”). Thus, the prediction approach is appropriate only when the effective channel utilization U_i in (1) is less than 1. Although our algorithm does not explicitly consider the constraint $U_i < 1$, minimizing the maximum U_i in (3) by the algorithm automatically enhances the chance of satisfying constraint $U_i < 1$ for all APs, thus making the prediction appropriate.

4. Performance evaluation

In this section, we use computer simulation to validate our enhanced MinMax algorithm and then study our adaptive scheme based on the enhanced method.

4.1. Validating the Enhanced Algorithm

We apply the enhanced MinMax algorithm as well as the original algorithm to a network layout for which the optimal channel assignment is known. The network is consisted of 21 cells (Fig. 3), each of which is represented by a hexagon and served by an access point at its center. APs in adjacent cells are 150 meters apart. The radio link between any pair of APs is characterized by a path-loss model with an exponential of 4.0. Transmit power for each AP is 30mW, and the CStreshold for channel-busy detection is set to -71.5dBm (7.0e-8 mW). All APs (and their respective

terminals) have identical traffic load. Three channels are available for assignment. For the setting, the optimal assignment is that no adjacent cells use the same channel.

We apply both original and enhanced MinMax algorithms to the network using only one initial assignment where every AP used the same channel. Results are shown in Fig. 3a and 3b, respectively. The 3 channels are represented in red, blue and yellow color. The enhanced algorithm yields a better solution than the original one. In (b), at most two adjacent cells share the same channel, while in (a) there are cases where four adjacent cells use the same channel.

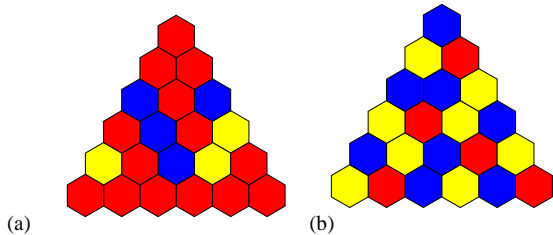


Figure. 3 Channel allocations for network with 21APs by (a) the original algorithm and (b) the enhanced algorithm.

4.2. Evaluating the Adaptive Channel Allocation

To study the performance of our proposed adaptive algorithm, we build network simulation using ns-2 [10]. The adaptive scheme is realized as new classes embedded in ns2. Moreover, two fixed channel allocation schemes, the *graph coloring algorithm* and *static MinMax algorithm*, are implemented in C++ external programs to serve as basis for comparison.

In the simulation model, we populate 25 APs randomly and uniformly in an area of 650×650 square meters. By carefully selecting parameters in class “WirelessPhy”, the interference range and receiving range for an AP are set to 200 meters and 170 meters, respectively. Each AP is assigned with 10 terminals that are randomly placed within the receiving range of the AP. All APs and terminals are stationary and the RTS/CTS protocol is used in every data frame transmission.

We simulate the network for 1000 seconds. In the first 200 seconds, every AP (and its terminals) has a constant offered traffic load of 0.5. After that, each AP changes its traffic load once every time period of 45 seconds where time period is indexed by t . Specifically, each AP i picks a random change W_i independently where W_i is uniformly distributed in the range of $[-0.1, 0.1]$ or $[-0.2, 0.2]$. Then the offered traffic load for AP i in the next time period $t+1$, according to the following formula:

$$\rho_i(t+1) = \rho_i(t) \cdot [1 + W_i] \quad (11)$$

where $\rho_i(t)$ is the offered traffic load of AP i at time period t . If $\rho_i(t+1)$ exceeds 1.0 or goes below 0, the AP i selects a new, random change W_i in $[-0.1, 0.1]$ or $[-0.2, 0.2]$ to ensure that $\rho_i(t+1)$ lies between 0 and 1.

Figures 4 and 5 show the UDP throughput performance in the 802.11 network as a function of

time for our proposed adaptive scheme and the two static channel allocation schemes. In Fig. 4, the traffic variation parameter W_i for each AP i is chosen randomly from $[-0.1, 0.1]$. Using the existing channel allocation as the initial assignment, the adaptive algorithm is invoked once every 60 seconds to determine the new channel assignment according to the new estimates of offered traffic load. As shown in the figures, during the constant traffic period (i.e. from 0 to 200 seconds), our adaptive scheme has similar performance with the static MinMax scheme. Nevertheless, the MinMax and thus the adaptive algorithms perform better than the coloring method. This is so because APs are placed randomly in the service area and some of them can be close to each other, thus causing strong mutual interference. The MinMax and adaptive schemes do consider such interference in their channel assignments, while the coloring method does not. However, when the traffic load starts to vary after 200 seconds, the adaptive scheme adapts the channel allocation to the variations and provides a throughput improvement of 10 to 15% over the static MinMax and graph coloring algorithms. As one would intuitively expected, the adaptive algorithm can yield a larger amount of improvement over the other two schemes as the traffic load demonstrates a larger degree of fluctuation, as shown in figure 5 where W_i is from $[-0.2, 0.2]$.

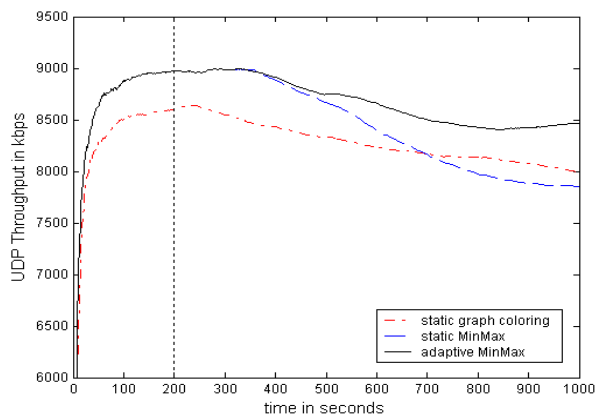


Figure. 4 Compare aggregated UDP throughput for W_i varying in the range of $[-0.1, 0.1]$.

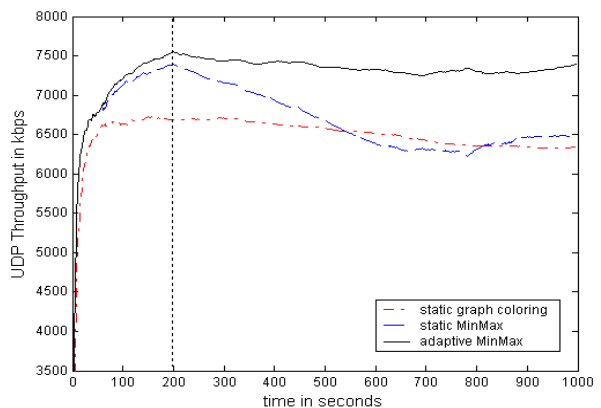


Figure. 5 Compare aggregated UDP throughput for W_i varying in the range of $[-0.2, 0.2]$.

One potential advantage of the proposed adaptive algorithm is to improve throughput by re-assigning channels for a relatively small number of APs. To reveal this characteristic, Fig. 6 shows the percentage of APs requiring channel changes throughout the simulation. For w_i varying in $[-0.1, 0.1]$ and $[-0.2, 0.2]$, at most 10% and 20% of APs, respectively, require channel re-assignment to adapt to traffic fluctuations. Usually, only a very small fraction of APs require channel re-assignments. Since channel changes take place at a very modest frequency, service interruption to users can be kept to an acceptable level despite a lack of protocol supports for efficient channel changes in the existing 802.11 standards. At the same time, the proposed adaptive algorithm is adequate to handle the traffic variations expected in typical office environments without resorting to use of dynamic channel allocation methods that clearly cannot be supported by existing 802.11 standards.

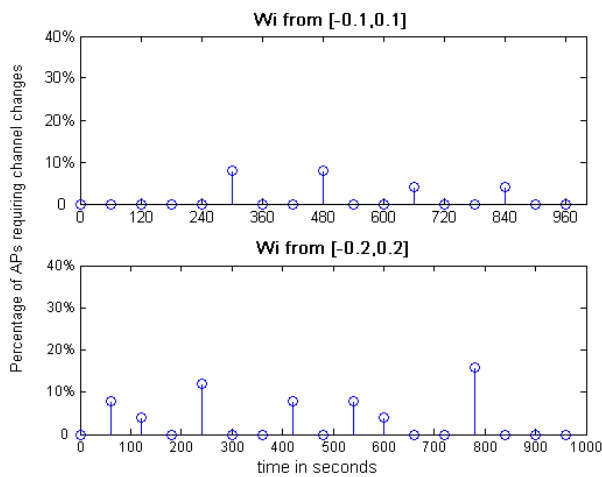


Figure. 6 Percentage of APs requiring channel changes.

5. Conclusions

We notice that static channel assignments are not sufficient for 802.11 networks due to traffic-load variations. However, in order to maintain compatibility with the existing protocol standards, we believe that it is not feasible nor is it necessary to employ dynamic channel allocation methods to improve system performance in light of traffic-load fluctuations. Rather, adaptive assignment schemes seem to be appropriate.

In this paper, we have proposed a centralized adaptive channel allocation scheme for 802.11 networks based on the MinMax algorithm proposed in [8]. Computer simulation using the ns2 software reveal that our proposed adaptive scheme achieves a noticeable improvement on the throughput performance over fixed assignment algorithms, while only a relatively small number of APs require channel changes in expected environments.

REFERENCES

[1] IEEE 802.11a, "Part 11: Wireless LAN Medium

Access Control (MAC) and Physical Layer (PHY) Specification: High-Speed Physical Layer Extension in the 5 GHz Band," 1999.

[2] IEEE 802.11b, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: High-Speed Physical Layer Extension in the 2.4 GHz Band," 1999.

[3] IEEE 802.11g, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band," 2003.

[4] Bob O'Hara and Al Petrick, *The IEEE 802.11 Handbook: a Designer's Companion*, IEEE Press, December 1999

[5] J. Riihijarvi, M. Petrova, and P. Mahonen, "Frequency allocation for WLANs using graph colouring techniques", *Proc. of IEEE Conference on Wireless On-Demand Network Systems and Services 2005*.

[6] J. Riihijarvi, M. Petrova, and P. Mahonen, "Automatic channel allocation for small wireless local area networks using graph colouring algorithm approach", *Proc. of IEEE Conference on Personal, Indoor and Mobile Radio Communications 2004*.

[7] H. Luo and N. K. Shankaranarayanan, "A distributed dynamic channel allocation technique for throughput improvement in a dense WLAN environment", *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing 2004*.

[8] K. K. Leung, and B. J. Kim, "Frequency Assignment for IEEE 802.11 Wireless Networks", *Proc. of IEEE Veh. Tech. Conference, 2003*.

[9] B. Abraham and J. Ledolter, *Statistical methods for forecasting*, New York: Wiley, 1983

[10] K. Fall and K. Varadhan, *The ns Manual*, March 2005, available at <http://www.isi.edu/nsnam/ns/>