Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2011, Article ID 972051, 13 pages
doi:10.1155/2011/972051

Research Article

Efficient Load-Aware Channel Allocation in Wireless Access Networks

George Athanasiou, Ioannis Broustis, and Leandros Tassiulas

Department of Computer and Communications Engineering, University of Thessaly, 38221 Volos, Greece

Correspondence should be addressed to Ioannis Broustis, broustis@cs.ucr.edu

Received 30 October 2010; Revised 7 February 2011; Accepted 2 March 2011

Academic Editor: Abdallah Shami

Dense deployments of hybrid WLANs result in high levels of interference and low end-user throughput. Many frequency allocation mechanisms for WLANs have been proposed by a large body of previous studies. However, none of these mechanisms considers the load that is carried by APs in terms of channel conditions, number of affiliated users as well as communication-load, in conjunction. In this paper, we propose LAC, a load-aware channel allocation scheme for WLANs, which considers all the above performance determinant factors. LAC incorporates an *airtime cost* metric into its channel scanning process, in order to capture the effects of these factors and select the channel that will provide approximately maximum long-term throughput. We evaluate LAC through extensive OPNET simulations, for many different traffic scenarios. Our simulations demonstrate that LAC outperforms other frequency allocation policies for WLANs in terms of total network throughput by up to 135%.

1. Introduction

The growing demand for high-throughput wireless Internet connectivity has enabled the deployment of thousands of WLANs in urban areas, during the last decade. This, however, has resulted in increased amounts of interference and contention among cochannel access points (APs) [1]. As a consequence, the end-users (affiliated clients with those APs) end up enjoying very low throughputs in the long term. Towards addressing this problem, various performance enhancement mechanisms have been proposed, a set of which considers the efficient allocation of frequency bandwidth (i.e., the available channels) to the APs and their clients (i.e., the AP cells) [2–6]. However, each of these frequency allocation studies considers only a subset of the parameters that affect the performance of the network.

(a) *Number of associated clients*. The number of the clients that are distributed in a network affects the communication interference that is present in the network. In other words, the higher the number of the clients associated with an AP, the higher the interference effect in the neighboring cells.

(b) *Channel conditions*. The channel conditions reflect the projected interference (SINR) and thereby the achievable transmission rates for a specific channel.

(c) *Communication load*. The communication load (amount of traffic that the APs must forward to the associated clients or to the network) affects the achievable throughput. The use of channels that are more capable to serve the load in the cells is expected to improve the network throughput.

There are no studies that embrace all these parameters under a common frequency selection framework; therefore the previously proposed solutions are efficient only when certain conditions are met (as we discuss in detail, in Section 2).

As our contribution in this paper, we propose LAC, a load-aware channel assignment scheme that discovers the channel assignment that approximately *maximizes the AP throughput*, that is, the sum of throughputs achieved by all its affiliated clients. LAC employs the *airtime cost* metric [7] (the *airtime cost metric* was proposed in [8] as a routing metric (RM-AODV) and in [7, 9] for optimal user association in 802.11 networks). The airtime cost directly reflects the environmental conditions around each device (AP or client), in terms of interference and contention experienced due to concurrent transmissions. By adopting this metric, LAC discovers the most appropriate channel for every AP in a distributed fashion, by measuring (a) both the downlink and uplink channel conditions in terms of supported transmission rates and packet error probability and (b) the number of affiliated clients with every AP. As a result, the AP has a unified knowledge with regard to the quality of all its downlink and uplink connections.

With LAC, every AP and its clients perform a sequential scanning on all the available channels and collect measurements with regard to the above metrics. Furthermore, they exchange these measurements to compute the cumulative airtime cost for the AP cell that they belong to, and for every channel. After a set of iterations they select the channel wherein the airtime cost is minimal; this is the channel that will provide approximately the maximum long-term throughput, since the throughput can be represented as an inverse function of the airtime cost. In [9] we have analyzed the relationship of the airtime metric with the uplink channel long-term throughput. In this paper, we extend this analysis in order to identify the relationship of the airtime metric with the total throughput in the cell (for both uplink and downlink channels).

We evaluate LAC through extensive simulations in OPNET [10]. We simulate the network behavior with the following types of traffic: (a) fully-saturated, downlink UDP traffic, where the APs send traffic to their associated clients, (b) fully-saturated, bidirectional UDP traffic (where the source-destination pairs are chosen randomly), (c) VoIP traffic and (d) video traffic. We compare our scheme against the frequency selection approach, proposed in [2] and against two other simple channel allocation schemes. We present LAC's predominance in terms of the total network throughput, average packet dropping,and average transmission delay.

The rest of the paper is structured as follows. Section 2 provides the relevant previous work on channel allocation for WLANs. Section 3 describes the airtime metric that we employ, based on which LAC discovers the channel with the maximum long-term throughput. In Section 4, we present the design of LAC. We evaluate our protocol through simulations, in Section 5. Finally, Section 6 concludes this paper.

2. Related Studies

In this section, we discuss the most relevant previous work on efficiently assigning channels to APs in 802.11 wireless LANs and in 802.11 wireless mesh networks. A large set of studies on channel selection in WLANs exists; to the best of our knowledge, however, none of these mechanisms captures the total actual throughput at the AP cell for every particular channel.

The LCCS (least congested channel search) method [11] was the first effort towards allocating a set of available channels to wireless devices. With LCCS, devices (e.g., APs) periodically scan the set of available channels and select the one with the lowest *levels of contention* (as the name suggests). However, there are many topological scenarios where LCCS is unable to capture the total interference in the channel, as explained in [4]. Similarly, Leith and Clifford [12] propose a self-managed distributed channel selection scheme, wherein each AP passively measures the received power from the packets transmitted by neighbor APs.

Along similar lines, Kauffmann et al. in [2] propose a distributed frequency selection algorithm, which is proved to minimize the global interference in the network. Simply put, minimizing the total interference can result in improved user throughput. Towards addressing this objective, each AP measures the total received power from all neighbor APs for every channel and selects the channel with the minimum total power. This is performed at each AP by measuring the RSSI of the received beacon frames from all the neighbor APs at every channel. The authors show that their proposed algorithm manages to converge to the global optimum of the optimization criterion, that is, the minimization of interference across the entire network. However, this algorithm does not consider the number of clients in the network; it assumes purely downlink saturated traffic and that all APs have affiliated clients. Moreover, the channel with the minimum total interference does not guarantee maximal throughput at the cell. We compare LAC against the protocol in [2], in Section 5.

Moreover, the work in [6], by Mishra et al., belongs to a set of studies that propose a distributed channel hopping mechanism. The mechanism in [6], MaxChop, provides higher levels of fairness among users. Channel hopping, however, requires tight synchronization between AP and clients, while it is difficult to implement efficiently with off-the-shelf hardware. Note that the channel switching and the subsequent restoration of traffic at the new channel may take from 700 to 1000 msec [13]; this is prohibitive in terms of incurred overhead.

Lee et al. [14] take into account the *expected* traffic demand points in the network. Their channel allocation strategy seeks to assign frequencies in such a way that the signal strength at these demand points is maximized. As a further step, Rozner et al. in [3] also consider the current traffic demands at the WLAN. In particular, they show that, taking into consideration the current traffic demands at APs and clients, the quality of the channel assignment can be greatly improved.

Furthermore, centralized channel allocation algorithms have been proposed in [4, 5, 15]. Mishra et al. [4] propose a frequency allocation scheme, wherein clients play a large role in the decision for the best channel. Their proposed approach opts to perform joint load balancing and frequency allocation. However, the approach is based on conflict graph coloring and cannot be directly implemented in a distributed setting. Leung and Kim [5] present a formulation of the channel assignment problem for 802.11 WLANs, which is then proven to be NP complete. Then, they design and analyze a heuristic algorithm that attempts to minimize the effective channel utilization for the bottleneck APs. The authors in [15] propose a novel framework to model the load of WLAN cells considering intercell interference. They also present a frequency planning algorithm which is designed on the basis of the aforementioned load model. Their algorithm provides fair service to its users, while preserving high network utilization.

Efficient channel selection is essential in 802.11 mesh deployments too, for minimizing contention and interference among cochannel devices. However, the requirements of a channel allocation policy there are different from a channel allocation policy applied in WLANs. A critical requirement for the efficient routing of packets is the identification and use of interference-limited wireless links. Therefore, intermediate mesh hops along a route need to operate in frequencies, where contention and interference are as low as possible, especially in highly dense mesh deployments.

Alicherry et al. [16] study the joint channel allocation and routing problem, assuming that traffic demands and network topology are known. They present an LP formulation of the problem, and they propose a centralized algorithm that maximize the aggregate throughput.

Raniwala and Chiueh propose in [17] a tree-based mesh architecture, called *Hyacinth*, where local channel usage and channel load information is exchanged, and the channel allocation is based on this information. They approach the joint problem of channel assignment and routing in wireless mesh networks.

Ramachandran et al. [18] propose a measurement-based centralized approach to provide efficient channel allocation for radios. They perform channel-to-interface assignment based on channel reuse possibilities which in turn depend on interference.

Wang and Garcia-Luna-Aceves [19] present JARS, a joint channel assignment, routing, and scheduling scheme, which incorporates the efficiency of underlying channel assignment and scheduling information into the routing metric calculation so that the route with the maximal joint spatial and frequency reuse is selected.

The authors in [20] develop a protocol that assigns contiguous channels with the goal of evenly spreading the load across the multiple channels. Neighboring nodes greedily adjust their channel ranges according to channel conditions to achieve an overall pattern of partially-overlapping bandwidths that maximizes the network throughput.

Lastly, an interesting work [21] addresses the threats to channel assignment in wireless mesh networks resulting from node misbehaviors and presents a generic verification framework to detect such misbehaviors. The authors develop a concrete verification scheme based on this framework and an existing distributed channel assignment schemes.

Most of the previous approaches assume the availability of a global network view. Our work differentiates from these approaches by providing efficient channel selection in a distributed manner.

Our work embraces the aforementioned parameters under a common frequency selection framework, since there are no approaches in the literature that consider all the factors that affect the channel selection, in conjunction. A new metric, called *airtime cost*, is used in our framework in order to discover the most appropriate channel for every AP in a distributed fashion. The *airtime cost* is able to capture the effects of the factors discussed in previous approaches and therefore to select the channel that will provide approximately maximum long-term throughput in the network.

3. A Metric for Channel Allocation

In this section, we present the metric that is used by LAC in order to effectively select the most appropriate channel in every cell.

The metric that is used in our load-aware channel allocation scheme is called *airtime metric*, and it is an approximation of per packet latency (as shown in [9] for the uplink channel). The airtime metric was first discussed in the 802.11s [8] standard, for the purposes of load-aware routing (RM-AODV routing protocol). This metric reflects the load on a wireless router (AP) in terms of the average delay a transmission of a unit size packet experiences. RM-AODV, which is the default routing protocol in 802.11s-based wireless mesh networks, employs the airtime metric in order to provide routes with the minimum total *airtime cost*. We adopt this metric in LAC for the purposes of our proposed channel selection functionality. In particular, we consider the *airtime cost* for the individual AP-client links; we measure the total *airtime cost* for the current cell, and we use it as a metric to select an appropriate channel (as we discuss later).

Formally, the airtime metric of station $i \in U_a$, where U_a is the set of stations associated with AP a that communicate using channel c, is given as

$$C_{a,c}^{i} = \left[O_{\text{ca}} + O_p + \frac{B_t}{R_i^{a,c}} \right] \frac{1}{1 - e_{pt}^{c}}. \tag{1}$$

In (1), O_{ca} is the channel access overhead, O_p is the protocol overhead, and B_t is the number of bits in the test frame (the transmission of test frames is necessary, in order to derive values for the computation of the airtime cost). Some representative values for these constants, for 802.11g, are: $O_{\text{ca}} + O_p = 1.25\,\text{ms}$ and $B_t = 8224\,\text{bits}$. Furthermore, $R_i^{a,c}$ and e_{pt}^{c} are the current transmission rate and frame-error rate, respectively, in Mbps, for the test frame size B_t in channel c. In other words, the estimation of e_{pt}^{c} corresponds to transmissions of standard-size frames B_t at the current transmit bit rate $R_i^{a,c}$.

The average *airtime cost* (in one direction: uplink or downlink) of AP a with N_a users, that operates on channel c is

$$\begin{aligned} \overline{C_{a,c}} &= \frac{1}{N_a} \sum_{i=1}^{N_a} \left[O_{\text{ca}} + O_p + \frac{B_t}{R_i^{a,c}} \right] \times \frac{1}{\left(1 - e_{pt}^{c}\right)} \\ &= \left[\frac{1}{N_a} \sum_{i=1}^{N_a} \left(O_{\text{ca}} + O_p \right) + \frac{1}{N_a} \sum_{i=1}^{N} \frac{B_t}{R_i^{a,c}} \right] \times \frac{1}{\left(1 - e_{pt}^{c}\right)}. \end{aligned} \tag{2}$$

The Case for Uplink Traffic. First, we discuss the relationship of the *airtime metric* with the average uplink network throughput. In [22, 23], the average uplink throughput in a single cell environment is calculated under saturation and decoupling approximations. The saturation approximation states that there are always packets backlogged on every user. Meanwhile, with the decoupling approximation it is assumed that when there are N users, the aggregate attempt process of $(N-1)$ nodes is independent of the back-off process of any given node. We consider the simple case, when all nodes have the same back-off parameters, each node is the transmitter for a single flow, and all packets have equal lengths. As derived in [22], the single flow average uplink saturation throughput $\theta(a,i,c)$ in channel c, of node i that is associated with AP a is given by (3),

$$\frac{1}{\theta(a,i,c)} = \frac{1}{\mu(1-\mu)^{N_a-1} L_i^a x_i^a} \times \left[1 + N_a\mu(1-\mu)^{N_a-1}\left(T_0 - T_c + \frac{1}{N_a}\sum_{q=1}^{N_a}\frac{L_q^a}{R_q^{a,c}x_q^a}\right) + \left(1-(1-\mu)^{N_a-1}\right)T_c\right], \tag{3}$$

where μ is the attempt rate (probability) in the equilibrium, $N_a = \sum_{q=1}^{N} x_q^a$ is the number of STAs (Stations) associated with AP a (x_i^a is a binary value that takes value 1 in case STA i is associated with AP a and 0 otherwise), T_0 is the fixed overhead with packet transmission, T_c is the fixed overhead for an RTS (Request To Send) collision, and L_i^a is the size of the packet of STA i when transmitting to AP a. Due to the exponential back-off behavior of the nodes and the decoupling approximation, it can be shown that the attempt probability of a node accessing the channel can be determined in terms of a given collision probability γ as

$$G(\gamma) = \frac{\sum_{k=0}^{K}\gamma^k}{\sum_{k=0}^{K}\gamma^k b_k}, \tag{4}$$

where K is the maximum number of attempts allowed by the protocol, and b_k is the mean back-off at the kth attempt. Meanwhile, the probability of collision of an attempt by a node is given by

$$\Gamma(\mu) = 1 - (1-\mu)^{N_a-1} \tag{5}$$

due to the decoupling approximation [22]. The behavior of the system at equilibrium is governed by the solution of the fixed-point equation

$$\gamma = \Gamma(G(\gamma)). \tag{6}$$

The solution of this equation yields the collision probability, from which the attempt rate in the equilibrium μ can be determined from (4).

In addition, $L_i^a/\theta(a,i,c)$ is the average per-packet delay at the equilibrium that STA i sends to AP a in channel c and is given by

$$\frac{L_i^a}{\theta(a,i,c)} = \frac{1}{N_a\mu(1-\mu)^{N_a-1}} + (T_0 - T_c) + \frac{1-(1-\mu)^{N_a}}{N_a\mu(1-\mu)^{N_a-1}}T_c + \frac{1}{N_a}\sum_{q=1}^{N_a}\frac{L_q^a}{R_q^{a,c}}x_q^a. \tag{7}$$

The first three terms in (7) represent the delay due to channel contention and protocol overheads, while the last term corresponds to the average transmission time of an L-length packet by a node in the cell. Clearly, the transmissions of RTS (request to send), CTS (clear to send), DATA, and ACK (acknowledgment) frames may also get corrupted not only due to collisions but also due to channel errors. The work in [24] assumes that the wireless channel can be modeled by an appropriate Gilbert model with known transition probabilities. In wireless networks with dynamically changing conditions, however, such an assumption is not practical. Therefore, in our work, the current frame error rate in channel c, e_{pt}^c is measured by users and AP. For each transmission attempt, the packet would be in error due to channel errors with probability e_{pt}^c. Clearly, the average number of attempts until successful transmission would be $1/(1-e_{pt}^c)$. In addition, for each transmission attempt, the average delay that is expressed in (7) is experienced. The product of $1/(1-e_{pt}^c)$ and the average per-packet delay of a client approach the average delay until successful transmission of this client:

$$\frac{L_i^a}{\theta(a,i,c)} \times \frac{1}{1-e_{pt}^c} = \left[\frac{1}{N_a\mu(1-\mu)^{N_a-1}} + (T_0 - T_c) + \frac{1-(1-\mu)^{N_a}}{N_a\mu(1-\mu)^{N_a-1}}T_c + \frac{1}{N_a}\sum_{q=1}^{N_a}\frac{L_q^a}{R_q^{a,c}}x_q^a\right] \times \frac{1}{1-e_{pt}^c}. \tag{8}$$

Comparing (2) and (8) we can clearly see that the average uplink *airtime cost* of AP a is an approximation of the average per-packet delay expressed in (8). In (2), $O_{ca} + O_p$ is the delay due to channel contention and protocol overhead, and $1/N_a\sum_{i=1}^{N_a}(B_t/R_i^{a,c})$ is the average transmission delay of a B_t-length packet (we can assume that $L_i^a = B_t$).

The Case for Downlink Traffic. We now discuss the relationship of the *airtime metric* with the average downlink network throughput. As derived in [2], the long-term average downlink saturated throughput for each client in cell a and for channel c is

$$\theta(a,c) = \frac{M_a}{\sum_{q=1}^{N_a} 1/R_q^{a,c}}, \tag{9}$$

where M_a is the fraction of time that AP a acquires the wireless channel. Therefore, the per-packet transmission delay towards each client, in channel c, is

$$\frac{L_i^a}{\theta(a,c)} = \frac{1}{M_a}\sum_{q=1}^{N_a}\frac{L_q^a}{R_q^{a,c}}. \quad (10)$$

Now, given the packet error rate e_{pt}^c, the average number of transmission attempts to successfully deliver a packet to the receiver, in channel c, is $1/(1-e_{pt}^c)$ (as explained before). Consequently, the lower bound of the expected transmission time for an STA to receive a packet successfully is

$$\frac{L_i^a}{\theta(a,c)} \times \frac{1}{1-e_{pt}^c} = \left[\frac{1}{M_a}\sum_{q=1}^{N_a}\frac{L_q^a}{R_q^{a,c}}\right] \times \frac{1}{1-e_{pt}^c}. \quad (11)$$

Finally, comparing (2) and (11) we claim that the airtime cost metric approximates the average downlink per-packet delay that each STA faces. In particular, M_a is directly related to the $O_{ca} + O_p$ overhead that is considered by the airtime cost metric. Hence, the average airtime cost is a representative metric that captures the downlink channel performance, in addition to the uplink one.

The main contribution of our work is that our scheme captures the performance of the available channels that can be used by the APs in the network, measuring the average *airtime cost* for both uplink and downlink (*airtime cost* for AP a, in channel c):

$$C_a^c = \overline{C_{a,c}^{\mathrm{up}}} + \overline{C_{a,c}^{\mathrm{down}}}, \quad (12)$$

and applies a channel allocation methodology where the channel with the minimum C_a^c is chosen. This also approximates the maximum throughput in the cell.

Various studies have shown that the number of erroneously received packets increases and the transmission rate decreases, in the presence of interfering cells in the network [2, 25]. Our proposed *airtime metric* takes into account the packet error rate as well as the transmission rate; hence, it reflects the performance at a particular communication channel. In the next section, we describe our channel allocation protocol; we explain how the *airtime metric* is used in order to optimize the allocation of the available channels and improve the network throughput.

4. LAC: Our Channel Allocation Scheme

In this section, we describe LAC, our load-aware channel selection mechanism. Our analysis from the previous section clearly shows that the airtime metric reflects the performance of the WLAN in terms of AP throughput. Hence, determining the channel with the lowest airtime cost will provide approximately the maximum long-term throughput to the clients within a cell. This is the target of LAC. LAC is AP-centric, that is, the channel choice is made by the AP at every cell. Note that the clients also play a very important role in the channel decision, by informing the AP with regard to the uplink channel conditions. This way the AP has knowledge with regard to both the downlink and uplink states. In a nutshell, LAC operates at both the AP and client ends. At every scanned channel, the AP and its clients measure the downlink and uplink channel properties and exchange them through their control and data transmissions. This information is subsequently used by the AP to select the channel with the minimum cumulative airtime cost. We explain the scanning operations of LAC in what follows.

Step 1 (computing the downlink airtime cost). At the nominal start of LAC, AP A_i of cell i initiates the downlink airtime cost calculation and informs its clients by setting a special bit into the beacon frame. Then it calculates the average downlink airtime cost for the links with its clients, through the link performance-measurement procedure, described in detail in [7]. In brief

(i) A_i calculates the frame error rate e_{pt}^c for each downlink communication on channel c, based on previous measurements (e.g., by measuring the percentage of the dropped packets in a time window),

(ii) A_i stores the transmission rate $R^{A_i,c}$ for each downlink communication on channel c,

(iii) A_i computes the airtime cost for each downlink communication on channel c, and

(iv) A_i computes the average downlink airtime cost on channel c.

Step 2 (computing the uplink airtime cost). The clients of A_i read the airtime cost bit from A_i's beacon frame transmissions, and they further calculate their individual uplink costs [7]

(i) STAs calculate the frame error rate e_{pt}^c on channel c in their uplink communication, based on previous measurements (e.g., by measuring the percentage of the dropped packets in a time window).

(ii) STAs store their uplink transmission rates on channel c.

(iii) STAs calculate their individual uplink airtime costs on channel c and inform the A_i.

(iv) A_i calculate the average uplink airtime cost on channel c.

STAs may include their costs into their measurement report messages towards A_i, as per [26]. Alternatively, they can also piggy-back this information through their data packets transmissions towards A_i. In this work we follow the latter approach; we discuss this choice later. Through this step, A_i receives information about the uplink channel qualities from all its clients.

Step 3 (deciding if the current channel is appropriate). A_i receives the client reports and computes the average airtime cost for both uplink and downlink, for the access level. If this is higher than a predefined threshold, T, then A_i remains in the same channel; otherwise it initiates a channel discovery process.

Step 4 (computing the cumulative airtime cost at the next available channel). A_i and its clients switch to the next channel and repeat Steps 1 to 3. If all available channels have been visited, A_i finally selects the channel with the minimum average airtime cost (for both uplink and downlink).

4.1. LAC Properties. We now discuss some properties of our LAC mechanism.

(1) *Band dwell duration*. Calculating $C^c_{A_i}$ in cell i for channel c requires that A_i and its clients conduct link measurements in order to derive values for the transmission rate and frame error rate, as explained in the previous section. Towards collecting accurate values for these parameters, a sufficient amount of time of residing at a particular channel is to be consumed by all devices of cell i. Note, however, that (a) the measurement collection process for every channel is performed through data exchange between AP and clients, and therefore traffic keeps flowing in the network even during channel scanning. In other words, with LAC, traffic does not stop flowing between AP and clients, as with other channel allocation protocols (e.g., [2, 12], etc.). (b) In typical today's WLANs, where clients are statically located most of the time, the levels of interference are not fluctuating to a large extent. Thus, LAC does not need to be executed frequently, as we discuss below. Hence, one can expect that due to (a) and (b) above, the projected overheads because of the channel dwell duration are minimal.

(2) *Convergence and frequency of invocation*. The set of Steps 1–4 belong to an iterative process, where the set of available channels is rescanned by the APs, until convergence has been reached. This is because whenever AP A_i decides upon a certain channel c_i, the cumulative airtime cost for the neighbor cells of A_i will likely change for channel c_i; this will *force* the APs of those cells to keep scanning for a potentially better channel than c_i. Our simulations in Section 5 show that in a static deployment with 20 number of APs and 40 number of clients, convergence is reached rather quickly. We expect that in a more dynamic topology, where clients join or deassociate from the network, the invocation of LAC would be more frequent, since the violation of threshold T would occur more often. This problem is not so prominent with the schemes proposed in [2, 12], since those mechanisms do not consider the potential transmissions of clients. Due to this, however, these approaches yield a much poorer performance than LAC, as we discuss in the following section. As far as the computation of the threshold T is concerned, we must mention that this is a system designer decision. T depends on the topology of the network and the communication load. Therefore, the system designer must adapt the threshold T according to the system characteristics. An automatic mechanism that can be used in order to adapt threshold T and provide a balanced network operation is presented in [9].

(3) *Embedding the airtime cost value into control and data frames*. With LAC, clients piggy-back their individual uplink airtime costs into their data transmissions towards the AP. As we mentioned earlier, another way of sending this information to their AP is through probe response frames, or special management report frames [26]. Note however that, as reported in [27], such control frames are not always received intact by their destinations, since they are not acknowledged. Therefore, the converge of mechanisms that depend on these frames may be delayed, since the AP does not manage to collect accurate information within the channel dwell duration [27]. Thus, we consider that the airtime cost value is repeatedly piggy-backed into every data packet transmission, for reliability.

5. Evaluating Our Load-Aware Channel Allocation Scheme

In this section, we evaluate our load-aware channel allocation scheme through extensive OPNET [10] simulations. We compare our scheme against the frequency selection approach, proposed in [2] and against two other simple channel allocation schemes. We present LAC's predominance in terms of the total network throughput, average packet dropping, and average transmission delay.

5.1. Simulation Set-Up Details. We have implemented LAC in OPNET [10], taking into account the 802.11 protocol operations. We have modified the beacon and data frames of 802.11 to facilitate the information exchange process that our protocol design requires. The clients and the APs are uniformly (at random) distributed in a 1000 m × 1000 m simulation area. All nodes use a default transmit power of 20 dBm. We simulate the network behavior with the following types of traffic: (a) fully saturated, downlink UDP traffic, where the APs send traffic to their associated clients, (b) fully saturated, bidirectional UDP traffic (where the source-destination pairs are chosen randomly), (c) VoIP traffic, and (d) video traffic. We repeat our simulations with both 802.11a and 802.11g modes of operation. We compare the network performance of LAC against the following.

(i) *Single-channel assignment (SC)*. This is a very common approach, where the APs use a preselected default channel (e.g., channel 6). The network device manufacturers design their products to operate on a default channel when they are turned on. Therefore, we believe that this situation is very common in real deployed wireless networks. We have performed several measurements, and we have observed the aforementioned operational characteristics.

(ii) *Random-channel allocation strategy (RC)*. The owners of the APs, in absence of build-in mechanisms that capture the interference in the environment, select randomly a communication channel. This is a common approach too.

(iii) *Gibbs-based Frequency Selection (GFS)*. We consider a scheme that is very similar to the one proposed in [2]; we call this scheme GFS. With GFS, each AP iteratively scans all the available channels, and greedily selects the channel with the (per channel) minimum aggregate received power (i.e., the minimum sum of RSSI values plus noise) from *all neighbor APs*. The design of GFS assumes purely downlink saturated traffic; that is, packets are assumed to flow only from the APs towards their clients (please see [2] for more details).

We evaluate the efficiency of LAC in selecting the most appropriate channel by measuring the total achieved network throughput, the average end-to-end delay, and the average dropped data packets in the network.

5.2. Simulation Results and Observations. We present our simulation experiments and the interpretations thereof, in what follows.

5.2.1. Applying Downlink UDP Traffic. To begin with, we opt to compare the performance with LAC against the performance with the other approaches, with downlink traffic (e.g., online movie downloading). For this, we apply saturated downlink UDP traffic in a network with 20 APs and 40 STAs, all uniformly randomly distributed in the area. In other words, we assume that the APs always have packets, to send to their associated clients. We use the default UDP packet size (1500 bytes). Figure 1 depicts the average total network throughput when there are 11 (802.11a) and 3 (802.11g) orthogonal channels available. The best performance is achieved by LAC, which uses the airtime metric to capture the cell performance at every channel. GFS underperforms because the number of the associated clients is not considered; the latter assumes that the imposed interference on an AP comes only from its neighbor APs, while the interference at the client is assumed to be approximately equal as that at the AP. Note however that in this scenario GFS does not achieve much poorer throughput than LAC, since the experiment involves downlink traffic only. In other words, the above assumption of GFS is *weak* in a purely downlink traffic scenario. Nevertheless, our load-aware scheme performs better than the other three channel selection approaches. In particular, LAC outperforms GFS by 8% and RC by 59% when 802.11a is used, and by 20% and 104%, respectively, when 802.11g is used. The improvement is higher with 802.11g networks because there are only 3 orthogonal channels available (less than the number of the available channels in 802.11a) and therefore, a random channel allocation policy is more likely to result to increased amounts of interference in the network (the channel reusability is increased). Our mechanism minimizes the interference in this environment and improves significantly the network performance. On the other hand, in 802.11a networks the improvement is not so high because the channel reusability of a random allocation policy is limited (as compared to 802.11g).

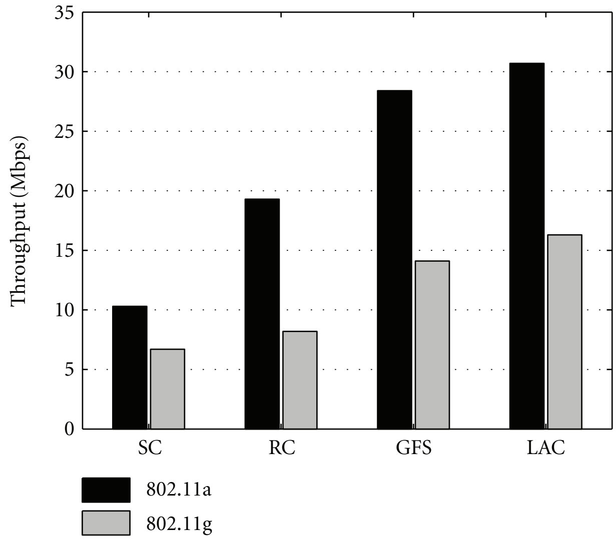

Figure 1: Total network throughput with saturated downlink UDP traffic.

5.2.2. Applying Bidirectional UDP Traffic between Random Source-Destination Pairs. We are also interested in the network performance with LAC, in more realistic (than downlink-only) traffic scenarios (e.g., gaming applications). In this case, each client sends saturated UDP traffic to another randomly selected client in the network. Note that the client-destination may be associated to a different AP than the client-source (in our simulations, APs are connected through a wireline Ethernet network, and they use the Ethernet interface to exchange packets with other cells). Hence, both uplink and downlink UDP traffic takes place in every cell (uplink traffic in the client → AP (source) link and downlink traffic in the AP → client (destination) link). Figure 2 depicts the average total network throughput, achieved with the different channel selection approaches. We observe that GFS underperforms LAC to a large extent (LAC improves network throughput achieved by GFS by 27% in 802.11a and by 47% in 802.11g). This is because GFS does not take into account the fact that also clients may be sending traffic towards their APs; hence, the uplink channel conditions and the load of each communication channel are not considered by GFS. Note that in our simulations we have clients being interfered by neighboring APs. Besides, the interference may be different than the one experienced by their affiliated APs. Hence GFS's assumption does not hold here. Finally, we observe that LAC improves the total network throughput as compared to RC, by 70% in 802.11a and by 135% in 802.11g.

Next, we seek to observe the throughput with LAC as the network density increases, in terms of number of clients. For this, we progressively increase the number of clients (uniformly, at random, distributed) from 5 to 70 in the network, while we maintain the same number of APs (20).

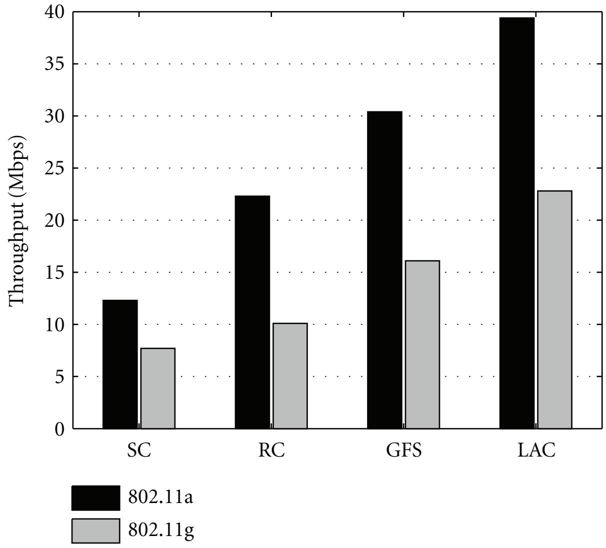

Figure 2: Total network throughput with both-directions UDP traffic.

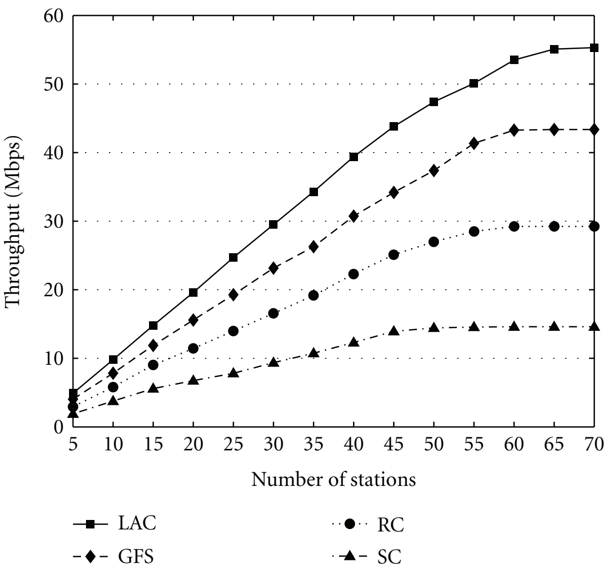

Figure 3: Total network throughput versus number of clients.

Figure 3 depicts the throughput results for the four channel selection strategies. We observe that the performance with LAC is similar to the one with the other 3 policies, when the number of clients (and therefore the load) is low. However, as the load increases, LAC manages to provide much higher throughputs due to its load-aware channel allocation strategy.

Moreover, we measure the average total network throughput as we increase the number of APs, from 10 to 80 (meanwhile, we deploy twice the number of clients: 10 APs—20 clients, 20 APs—40 clients, etc.). Figure 4 depicts the throughput gains with LAC. We observe that our scheme manages to scale much better than all other 3 approaches. In other words, the maximum achievable total network

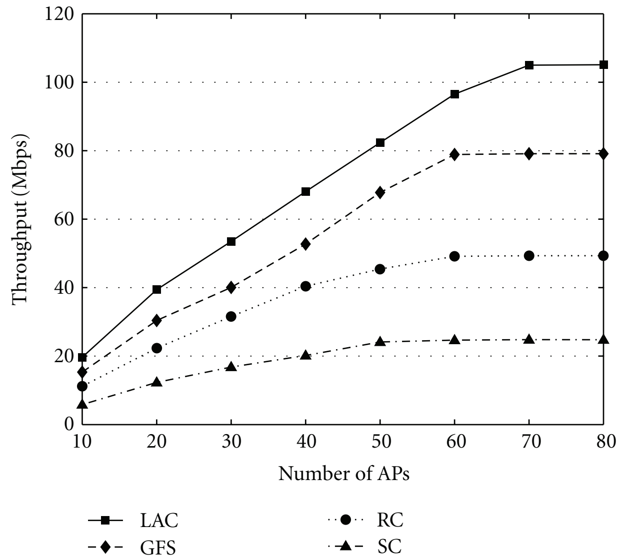

Figure 4: Total network throughput versus number of APs.

throughput is reached when the network includes 10 more APs for the case of LAC than in the case of GFS. The reason behind this is that LAC is able to capture and handle the interference in an efficient way and provide approximately maximum network throughput. LAC "stretches" the network capabilities.

5.2.3. Simulating LAC with VoIP Traffic. In order to observe the performance of our protocol with delay-sensitive applications, we utilize varying, parallel, end-to-end VoIP traffic sessions. The simulation setup is the same as for the previous bidirectional experiments; that is, VoIP traffic is exchanged among clients in our network—hence we again have both uplink and downlink traffic at every cell. Figures 5 and 6 present the network performance with VoIP.

In particular, Figure 5 depicts the average end-to-end delay of VoIP packet transmissions. We observe that LAC achieves low end-to-end delays, due to its sophisticated channel allocation strategy. GFS achieves quite good performance in low load communication conditions. However, as the number of the supported VoIP sessions increases, the channel setting with GFS is unable to efficiently support them. Unfortunately, GFS considers just the downlink channel characteristics, and therefore it is enabled to support highly loaded VoIP traffic (where both uplink and downlink traffic is introduced in the system). Finally, Figure 6 shows the average number of dropped data packets due to channel errors and contention. The performance of LAC is impressive, since packet dropping is kept in very low levels, as compared to the other strategies.

5.2.4. Simulating LAC with Video Traffic. We measure the performance of our protocol when video traffic is introduced in the network. In order to simulate MPEG-4 video traffic over a WLAN, we have imported video traces to OPNET. Specifically, we have obtained MPEG-4 video traces from

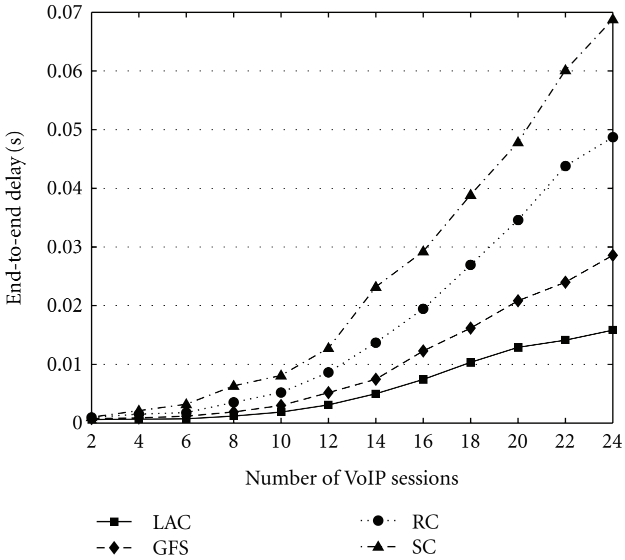

Figure 5: Average end-to-end delay with VoIP traffic.

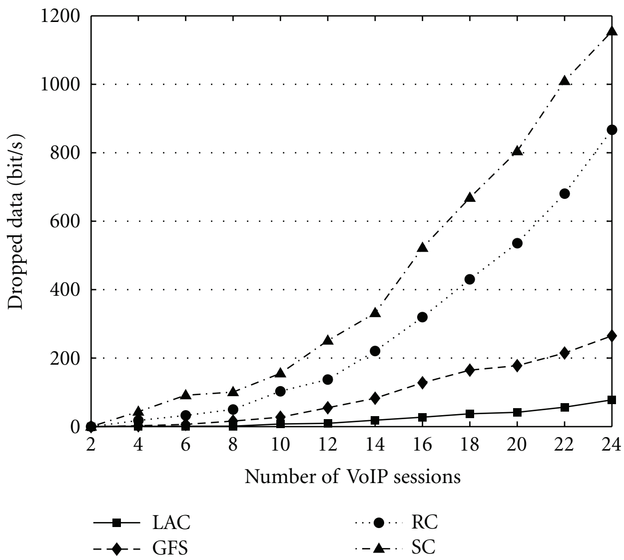

Figure 6: Average dropped data with VoIP traffic.

a 60 m movie: *Jurassic Park*, which is available in [28]. Then we have created a traffic profile in OPNET where a transmission interval of 20 ms is introduced. Table 1 contains more information about the parameters of the video traffic that we use in our experiments. We keep the same simulation setup as for the previous experiments, and we measure the performance of LAC compared to SC, RC, and GFS while the number of the video connections (sessions) that are supported in the network varies.

Figure 7 depicts the average end-to-end delay of video data transmission. The load-aware channel allocation policy introduced by LAC keeps the transmission delay at low levels, while the performance of SC and RC is disappointing. The performance of GFS is very good in low load conditions, but in high load conditions the performance drops. As we can see in Figure 7, when we support more than 8 video connections we face huge transmission delays with SC and RC in the system. The packet size varies in video traffic (in contrast to CBR traffic), and the amount of the transmitted data is large. Therefore, the interference and the packet congestion is increased in the network. LAC takes into account the communication interference and provides a balanced network operation. Figure 8 shows the average delivery ratio in the video data packet transmission, and Figure 9 depicts the average amount of propped data while the number of the video connections increases. These simulation results confirm that SC and RC are unable to efficiently support more than 8 or 10 video connections in the network and that LAC provides the best network performance.

Table 1: Video traffic parameters.

Parameter	Value
Coding format	QCIF (176 × 144)
Transmission interval	20 ms
Frame rate	25 frames/sec
Frame structure	IBBPBBPBBPBB
Mean frame size	0.77 KByte
Compression ratio YUV : MP4	49.46

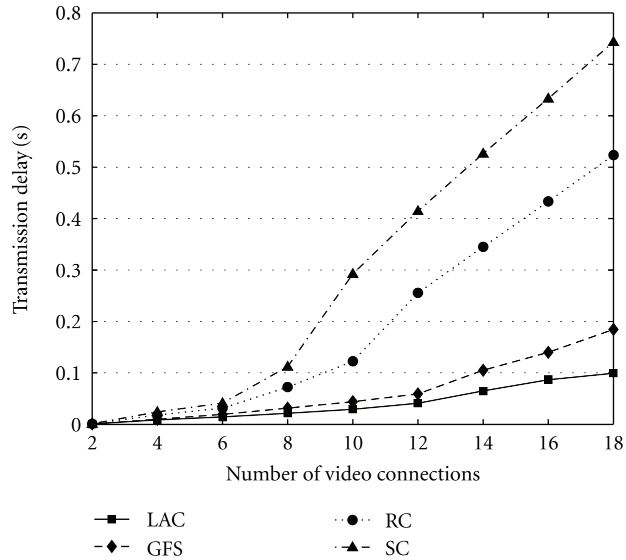

Figure 7: Average end-to-end delay with video traffic.

5.2.5. Supporting Different Types of Traffic. Now we opt to measure the performance of the compared schemes in a simulation scenario which approaches real life network operational conditions. We make good use of the capabilities that the OPNET simulation environment offers, supporting in this way several applications in parallel and therefore testing the adaptability of our protocol under different traffic demands in the network. Specifically, we introduce the following applications: HTTP (5 clients—heavy browsing), E-mail (8 clients), FTP (5 connections—500 Kb file size),

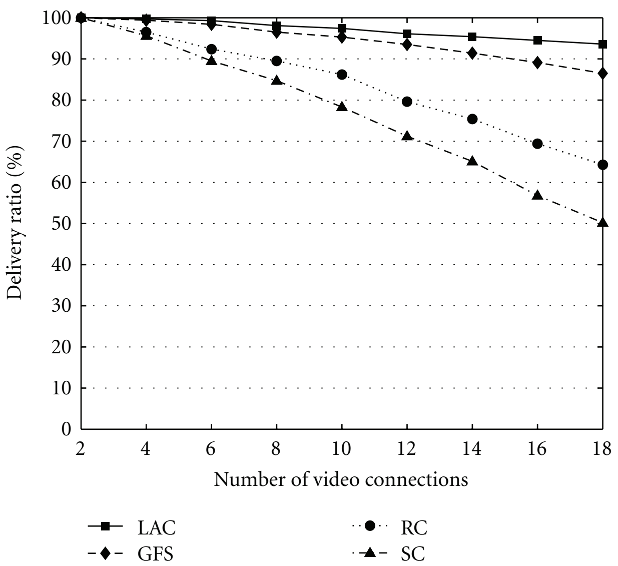

Figure 8: Average delivery ratio with video traffic.

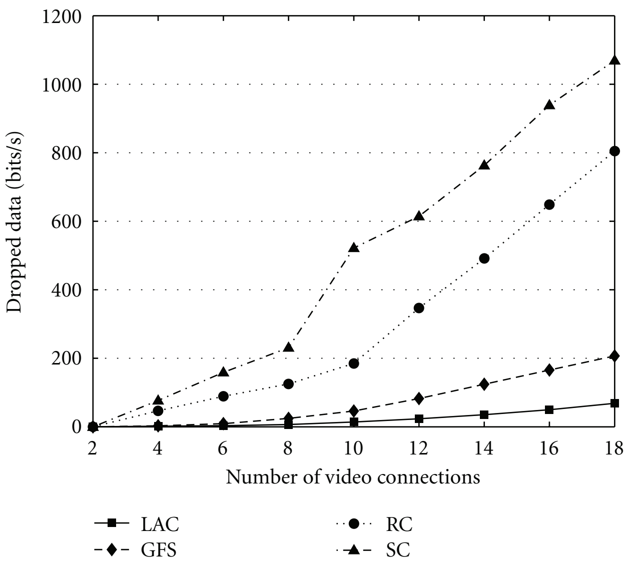

Figure 9: Average dropped data with video traffic.

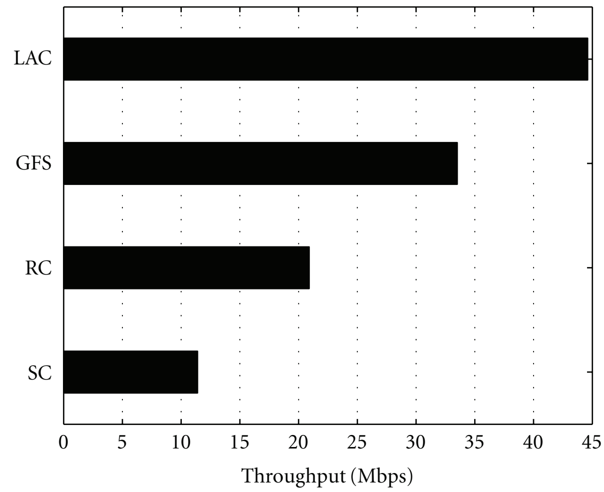

Figure 10: Total network throughput supporting different types of traffic.

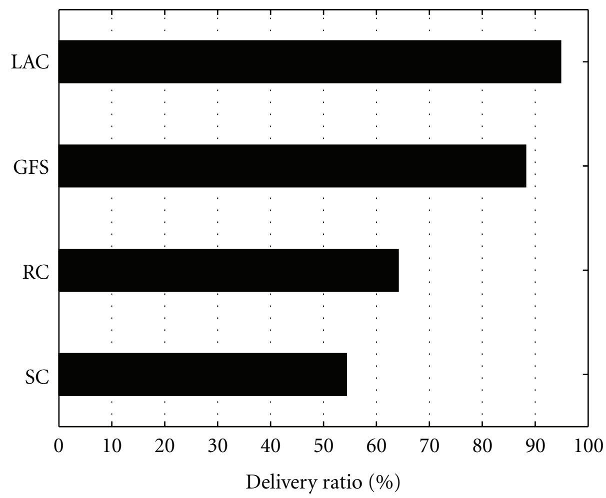

Figure 11: Average delivery ratio supporting different types of traffic.

VoIP (8 connections), and video (5 connections—MPEG-4 traces).

In Figure 10, we observe the average total network throughput when the aforementioned applications are supported. LAC uses the airtime metric to capture the cell performance at every channel and achieves the best performance. In particular, LAC outperforms GFS by 34% and RC by 114%, and therefore LAC can efficiently support the aforementioned applications in parallel. Figure 11 depicts the percent of the successfully transmitted packets in the network. LAC achieves high average delivery ratio while SC and RC are enable to support the traffic demands in this simulation scenario. The performance of GFS is quite good especially in low load communication conditions. Its performance drops when the load in the networks is increased to high levels, and therefore the average delivery ratio is average.

5.2.6. LAC Convergence. In Section 4, we provided a discussion with regards to the achievable convergence of our scheme. In order to visualize the discussion points, we examine how quickly LAC converges, and the propagation effect that is caused by a potential performance variation in the network when LAC is applied (we discuss later why a performance variation happens). We keep the same simulation settings with the previous experiments and we apply bidirectional UDP traffic in the network.

In the first scenario, we simulate a network with 20 APs and 40 STAs, all uniformly randomly distributed in the area. We apply LAC, and a convergence is reached after a finite number of iterations. Then, we randomly choose an AP in the network and we vary its communication characteristics (e.g., we manually change the channel that the AP uses) in

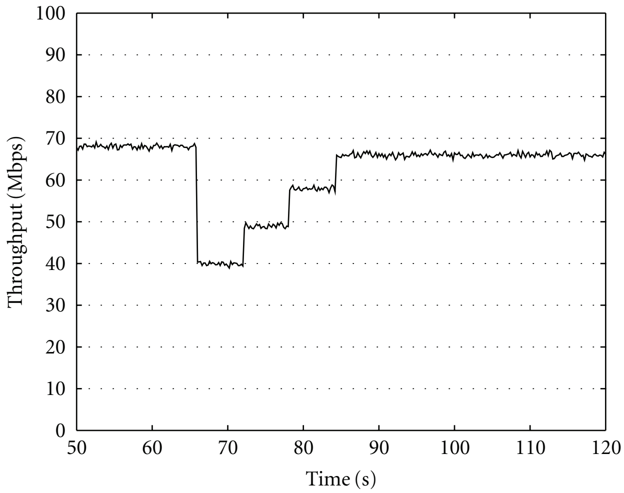

Figure 12: LAC convergence.

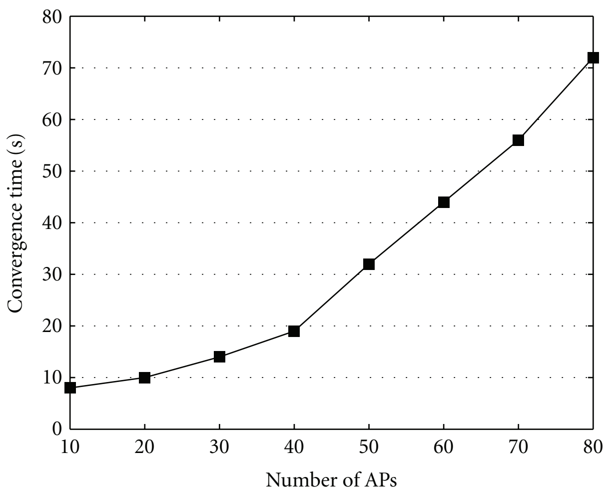

Figure 13: Convergence time versus number of APs.

order to force it to re-execute LAC. In other words, the performance of this AP exceeds the threshold T that we have introduced in LAC. Therefore, LAC must be re-executed to "redeem" the efficient network performance. Figure 12 depicts the realtime total network throughput. We observe that the network performance has reached a stable state (LAC has converged). Then, we randomly choose an AP and we manually change the communication channel that is used (selected by LAC). In this way we modify the interference in the network. In Figure 12, we can observe the adaptation of LAC when this performance variation happens. After a small number of iterations the network reaches again a stable state (LAC reconverges). The time that passes till convergence in our simulation scenario is close to 20 ms.

Furthermore, we measure the time that passes until convergence, as we increase the number of APs from 10 to 80 (meanwhile, we deploy twice the number of clients: 10 APs—20 clients, 20 APs—40 clients, etc.). We repeat the previous experiment while we vary the number of the APs in the network, in order to measure the LAC average convergence time in different topologies and communication scenarios (we run several experiments to get accurate results). Figure 13 demonstrates that the average convergence time is less than 20 sec when we support up to 40 APs in the network. After that point, the average convergence time is significantly increased due to the large number of the APs that must execute LAC and share the available channels in the network.

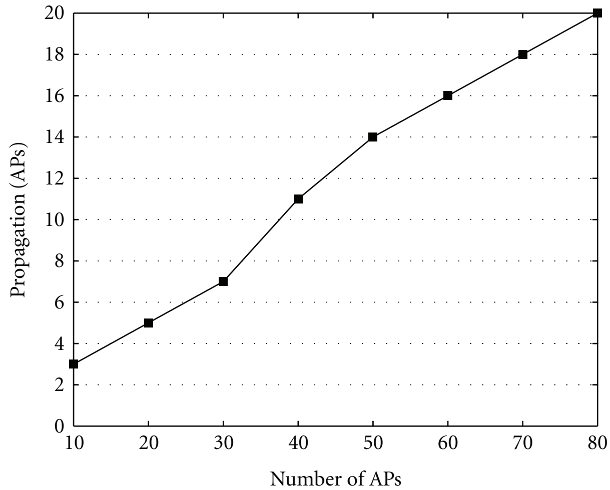

Figure 14: Propagation effect when the AP that "fires" the execution of LAC is placed at the center of the network.

Finally, we opt to observe the propagation effect that is caused by our scheme in the network. As we have described before, our mechanism reaches a stable state after a finite number of iterations (convergence). The network performs efficiently in this state. However, a potential performance variation in the network may trigger the re-execution of our mechanism. The performance variation may be caused by increased interference (new STAs/APs are turned on in the network or the transmission power of the current STAs/APs changes), variations in the network topology (mobile STAs), variation in the environment (buildings, walls), and so forth. After convergence, LAC turns into monitor mode in order to "act" in case that the performance exceeds the threshold T that is defined by the system designer. The re-execution of LAC in one or more cells may initiate a "domino" effect. In other words, the new channel set that outcomes from the re-execution of LAC in a cell may affect the performance of the neighboring cells. Therefore, LAC must be re-executed in the neighboring cells in order to select a better channel set that will improve their performance. We call this "domino" as propagation effect. It is true that the threshold T controls the sensitivity of a potential LAC propagation effect in the network. We support the fact that the system designer must manage a tradeoff that is present during the execution of LAC. High threshold values affect the sensitivity and the efficacy of LAC, and low threshold values may trigger frequent unavailing LAC re-executions (propagation effect). Consequently, the system designer must adapt the threshold T according to the system characteristics. An automatic

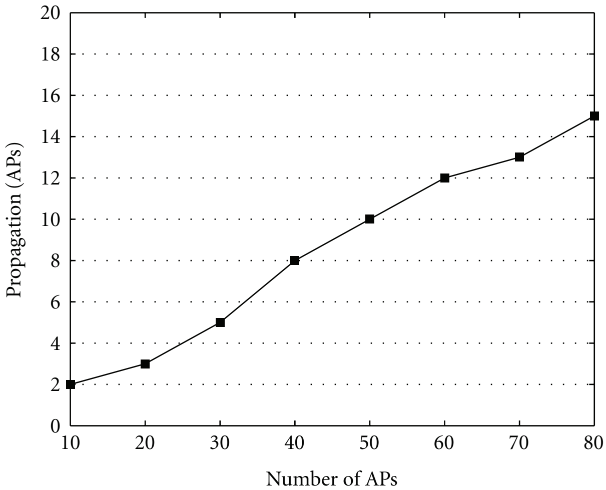

Figure 15: Propagation effect when the AP that “fires” the execution of LAC is placed at the edge of the network.

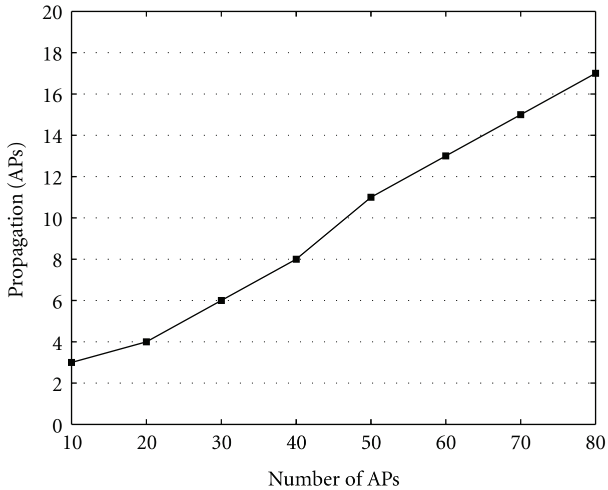

Figure 16: Propagation effect when the AP that “fires” the execution of LAC is placed at random of the network.

mechanism that can be used in order to adapt threshold T and provide a balanced network operation is presented in [9].

We measure the number of the APs that re-execute LAC due to a possible fluctuation in the network performance. We examine three scenarios where the AP that “fires” the execution of LAC is placed at the center, at the edge and at random in the network. In Figures 14, 15, and 16 we observe the propagation effect of the aforementioned three scenarios as we increase the number of APs, from 10 to 80 (we run several experiments in order to get accurate results). As we can see, the highest propagation effect is present when the AP that “set off” the execution of LAC is located at the center of the network. That happens because at the center of the network the number of the APs that are affected by the increased interference is large and LAC must be executed by all these APs. Contrarily, when the AP is located at the edge of the network, the propagation effect is quite low.

6. Conclusions

In this paper, we propose LAC, a load-aware channel selection mechanism for 802.11-based WLANs. LAC adopts the airtime cost metric, which was originally proposed in the 802.11s standard, and which provides an estimation for the average packet transmission delay. LAC employs a channel scanning procedure that converges to the channel with the minimum average airtime cost (for both uplink and downlink), thus managing to provide high cell throughput. We compare LAC against 3 other channel selection approaches, and we show that it outperforms all of them, for different traffic scenarios and network densities. Efficient channel allocation is very important in dense deployments of WLANs, where the interference in increased and the end-user throughput is very low. LAC is able to capture the communication environment characteristics and effectively provide a distributed channel allocation methodology which guarantees high network performance.

Acknowledgment

This work has been supported by EU through the FP7 project OPNEX 224218.

References

[1] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, “Self-management in chaotic wireless deployments,” in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MOBICOM ’05)*, pp. 185–199, September 2005.

[2] 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 the 26th IEEE International Conference on Computer Communications (INFOCOM ’07)*, pp. 1451–1459, May 2007.

[3] E. Rozner, Y. Mehta, A. Akella, and L. Qiu, “Traffic-aware channel assignment in enterprise wireless LANs,” in *Proceedings of the 15th IEEE International Conference on Network Protocols (ICNP ’07)*, pp. 133–143, October 2007.

[4] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, “A client-driven approach for channel management in wireless LANs,” in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM ’06)*, April 2006.

[5] K. K. Leung and B. J. Kim, “Frequency assignment for IEEE 802.11 wireless networks,” in *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC ’03)*, vol. 3, pp. 1422–1426, October 2003.

[6] A. Mishra, V. Shrivastava, D. Agrawal, S. Banerjee, and S. Ganguly, “Distributed channel management in uncoordinated wireless environments,” in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM ’06)*, pp. 170–181, Los Angeles, Calif, USA, September 2006.

[7] G. Athanasiou, T. Korakis, O. Ercetin, and L. Tassiulas, "Dynamic cross-layer association in 802.11-based mesh networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 2090–2098, May 2007.

[8] IEEE 802.11s: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Simple Efficient Extensible Mesh (SEE-Mesh) Proposal.

[9] G. Athanasiou, T. Korakis, O. Ercetin, and L. Tassiulas, "A cross-layer framework for association control in wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 1, pp. 65–80, 2009.

[10] "OPNET-Radio/Wireless Models," http://www.opnet.com.

[11] J. Geier, "Assigning 802.11b access point channels," in *Proceedings of the WiFI Planet Conference*, 2002.

[12] D. J. Leith and P. Clifford, "A self-managed distributed channel selection algorithm for WLANs," in *Proceedings of the 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOPT '06)*, pp. 1–9, Ireland, April 2006.

[13] B. Vedantham, S. Kakumanu, S. Lakshmanan, and R. Sivakumar, "Component based channel assignment in single radio, multi-channel ad hoc networks," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking, (MOBICOM '06)*, pp. 378–389, September 2006.

[14] Y. Lee, K. Kim, and Y. Choi, "Optimization of AP placement and channel assignment in wireless LANs," in *Proceedings of the 27th Annual IEEE International Conference on Local Computer Networks (LCN '02)*, Tampa, Fla, USA, November 2002.

[15] S. J. Han, Y. Bejerano, and M. Smith, "A novel frequency planning algorithm for mitigating unfairness in wireless LANs," in *Computer Networks*, vol. 54, no. 15, pp. 2575–2590, October 2010.

[16] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM '05)*, pp. 58–72, 2005.

[17] A. Raniwala and T. C. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05.)*, pp. 2223–2234, March 2005.

[18] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, April 2006.

[19] X. Wang and J. J. Garcia-Luna-Aceves, "Distributed joint channel assignment, routing and scheduling for wireless mesh networks," in *Computer Communications*, vol. 31, no. 7, pp. 1436–1446, May 2008.

[20] E. Chai and K. G. Shin, "M-Polar: channel allocation for throughput maximization in SDR mesh networks," in *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM '10)*, March 2010.

[21] M. Kim and P. Ning, "SeCA: a framework for secure channel assignment in wireless mesh networks," *Computer Communications Elsevier Journal*, vol. 34, no. 4, pp. 567–576, 2011.

[22] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[23] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New insights from a fixed point analysis of single cell IEEE 802.11 WLANs," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pp. 1550–1561, March 2005.

[24] N. Gupta and P. R. Kumar, "A performance analysis of the 802.11 wireless LAN medium 13 access control," *Communications in Information and Systems*, vol. 3, no. 4, pp. 279–304, 2004.

[25] D. Niculescu, "Interference map for 802.11 networks," in *Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference (IMC '07)*, pp. 339–350, October 2007.

[26] IEEE 802.11 WG, *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Radio Resource Measurement, IEEE 802.11k/D3.0*, IEEE, New York, NY, USA, 2005.

[27] I. Broustis, K. Papagiannaki, S. V. Krishnamurthy, M. Faloutsos, and V. P. Mhatre, "Measurement-driven guidelines for 802.11 WLAN design," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 722–735, 2010.

[28] "MPEG-4 and H.263 Video Traces for Network Performance Evaluation," http://www.tkn.tu-berlin.de/research/trace/trace.html.