UNIVERSITY OF THESSALY

Cross-Layer Resource Allocation Algorithms for Wireless Networks

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy
in
Computer and Communications Engineering
by
George Athanasiou
December 2009

Dissertation Committee:

Prof. Leandros Tassiulas, Chairperson
Prof. Catherine E. Houstis
Dr. Iordanis Koutsopoulos

The Dissertation of George Athanasiou is approved by:

_____

_____

_____

Committee Chairperson

University of Thessaly, Volos

ACKNOWLEDGMENTS

I would like to express my gratitude to all those who made this dissertation possible. First of all, I would like to thank my mentor Prof. Leandros Tassiulas for his restless effort to introduce me to the fascinating, but demanding, world of research. As my supervisor, he has constantly forced me to remain focused on achieving my goal. His observations and comments helped me to establish the overall direction of the research and to move forward with investigation in depth. I am very grateful to the remaining members of my dissertation committee, Dr. Iordanis Koutsopoulos and Prof. Catherine E. Houstis. Their academic support and personal cheering are greatly appreciated. I would like to express my sincere thanks to Dr. Thanasis Korakis for generously sharing his time and knowledge in our cooperative work. Besides, I would like to thank my friend Dr. Ioannis Broustis for sharing his stimulating ideas and expertise during the last part of my thesis. I thank Prof. Shivendra S. Panwar for hosting me as a summer intern in the summer of 2006, in Electrical and Computer Engineering Department of Polytechnic Institute of New York University, New York, USA. Furthermore, I thank my friend Dr. Alberto Lopez Toledo for giving me the opportunity to work at Telefonica Research, Barcelona, Spain, in the summer of 2009. My collaboration with him has been an invaluable experience. I would like to thank my friends and lab-mates, who provided help and made the long journey a lot more enjoyable and memorable. Finally, I would like to thank my family for their love and support. There are no words that can express my gratitude and appreciation for all they have done for me. The least I can do in recognition to their love is to dedicate this thesis to them.

To my parents

ABSTRACT OF THE DISSERTATION

Cross-Layer Resource Allocation Algorithms for Wireless Networks

by
George Athanasiou

Doctor of Philosophy, Graduate Program in Computer and Communications Engineering
University of Thessaly, Volos, Greece
Prof. Leandros Tassiulas, Chairperson
December 2009

The management of the available resources in wireless networks is an important task that must be executed in an efficient way, in order to optimize the network performance. In this thesis we design a *Cross-Layer Resource Allocation Framework* that can be applied in IEEE 802.11-based wireless networks and includes: sophisticated user association policies, fast handoff mechanisms, optimized TCP performance over mutli-AP association mechanisms and load-aware channel allocation policies.

The user association mechanism specified by the IEEE 802.11 standard does not consider the channel conditions and the AP (Access Points) load in the association process. Employing the mechanism in its plain form in wireless mesh networks, we may only achieve low throughput and low user transmission rates. We design a new association framework in order to provide optimal association and network performance. In this framework, we propose a new channel-quality-based user association mechanism inspired by the operation of the infrastructure-based WLANs. Besides, we enforce our framework by proposing an airtime-metric-based association mechanism that is aware of the uplink and downlink channel conditions as well as the communication load. We then extend the functionality of this mechanism in a cross-layer manner taking into account information from the routing layer, in order to fit it in the operation of wireless mesh networks. Last, we

design a hybrid association scheme that can be efficiently applied in real deployments to improve the network performance. We evaluate the performance of our system through simulations and testbed experiments, and we show that wireless mesh networks that use the proposed association mechanisms are more capable in meeting the needs of QoS-sensitive applications.

According to the IEEE 802.11 standard, the STAs get information about the active APs in their neighborhood by scanning the available channels and listening to transmitted beacons. We propose an IEEE 802.11k compliant framework for cooperative handoff where the STAs are informed about the active APs by exchanging information with neighboring STAs. Besides, the APs share useful information that can be used by the STAs in a handoff process. In this way we minimize the delay of the scanning procedure. We evaluate the performance of our mechanisms through OPNET simulations. We demonstrate that our scheme reduces the scanning delay up to 92%.

In the aforementioned research approaches we considered single AP association. However, the high bandwidth demand of applications such as P2P and video-streaming has recently driven the need for connecting to multiple APs and bonding the ADSL backhauls via 802.11 connections. Since 802.11 APs and stations are usually single-radio, the communication to the set of APs naturally requires a time-division multiple access (TDMA) policy. However, the TDMA approach introduces delay in the end-to-end transmissions that can adversely affect the TCP throughput. We first perform an in-depth experimental analysis of how multi-AP TDMA affects the observed round-trip time of TCP packets. Then we introduce a model that accurately predicts the performance of TCP on such environments. Based on this model, we propose a local resource allocation algorithm (minmax disconnection time) that minimizes this TCP degradation with a very low computational cost. The algorithm splits the original TDMA allocation in slots of shorter sizes such as the time that a station is disconnected from any AP is minimized. We show that the proposed scheme can improve up to 1.5 times the aggregate throughput observed by the station compared to a standard TDMA allocation. We show that the performance of the algorithm is very close to the theoretical upper-bound in several simulation scenarios.

Dense deployments of hybrid 802.11-based WLANs result to 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. 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%.

We then extend the previous mechanisms in order to provide efficient channel selection in 802.11 mesh deployments, for minimizing contention and interference among co-channel devices and thereby supporting a plurality of QoS-sensitive applications. We propose ARACHNE, a routing-aware channel selection protocol for wireless mesh networks. ARACHNE is distributed in nature, and motivated by our measurements on a wireless testbed. The main novelty of our protocol comes from adopting a metric that captures the end-to-end link loads across different routes in the network. ARACHNE prioritizes the assignment of low-interference channels to links that (a) need to serve high-load aggregate traffic and/or (b) already suffer significant levels of contention and interference. Our protocol takes into account the number of potential interfaces (radios) per device, and allocates these interfaces in a manner that efficiently utilizes the available channel capacity. We evaluate ARACHNE through extensive, trace-driven simulations. We observe that our protocol improves the total network throughput, as compared to three other channel allocation strategies.

Contents

List of Figures

Chapter 1

Introduction

1.1 Association in 802.11-based Wireless Networks

In recent years, IEEE 802.11 wireless local area networks (WLANs) have experienced widespread deployment in enterprises, public areas and homes. A typical WLAN consists of a wired/wireless backbone connecting a number of wireless cells in a seamless manner. If the backbone is all wireless, then the network is called a *Wireless Mesh Network* [3], [65], where each backbone node operates both as a wireless router that forwards packets of other nodes, and as a wireless bridge transmitting the packets of its clients. The mesh network can be considered as an infrastructure network that disposes characteristics of its ad-hoc equivalent. Such a network is self-organized and self-configured, exploiting all the benefits of the dynamic nature of an ad-hoc network. There is an active interest and ongoing research on implementing community and commercial mesh networks, e.g., [26].

In 802.11 WLANs, a client associates with a single access point (AP) that carries all traffic to and from the client. The client-AP association is an important issue, which greatly affects the network performance, fairness and the provision of user quality of service (QoS) requirements. In current implementations, each client station (STA) scans the wireless channel to detect the APs nearby, and associate itself with the AP that has the strongest received signal strength indicator

(RSSI). With this approach, it is expected that a STA associates itself with the closest/strongest AP. Recent studies have shown that this policy can lead to inefficient use of the network resources [6], [17], [18]. The most important disadvantage of RSSI-based user association is that RSSI does not provide any information about the current load of the AP in terms of its average downlink/uplink throughput. Moreover, the association problem in wireless mesh networks must be dealt with differently compared to the problem in conventional infrastructure wireless networks without wireless backbones. In particular, the packet routing delay on the backbone can no longer be considered negligible due to lower physical transmission rates and wireless channel contention.

In this thesis ([7], [8]), we propose easy-to-implement client association mechanisms for wireless mesh networks. The proposed mechanisms are all designed to be compliant with the current 802.11 standards, and they are shown to significantly improve the network performance in terms of the throughput and the end-to-end delay under different operational scenarios. The main aspects of the proposed association mechanisms are that they take into account the loads of APs, end-to-end delay on the wireless backbone and the transmission error probabilities of the uplink and downlink channels.

In particular, we investigate new association mechanisms for both coordinated and uncoordinated wireless mesh networks. In *coordinated mesh networks*, adjacent cells operate at orthogonal channels, and thus, transmissions of the clients in a cell do not interfere with the transmission of those in other adjacent cells. In contrast, in *uncoordinated mesh networks*, cells may operate at the same channel, and thus, a client cannot transmit at the same time when another client in an adjacent cell is transmitting.

As explained in detail in Section 3.1, for uncoordinated mesh networks the most important parameter for association is the quality of the channel between the client and the AP. This is mainly due to the fact that when all adjacent APs operate at the same channel, all nodes overhear each other, and thus, the load observed in one cell is the same as in the adjacent cells. In order to be compliant with the 802.11 standards while estimating the channel quality, we employ the procedures defined under 802.11h [44]. Based on the estimated channel conditions, we propose a greedy *channel*

quality based user association mechanism that selects the AP that can provide the maximum bi-directional link rate.

In contrast, in coordinated mesh networks, the loads of the individual cells become important in the association decision. In order to embrace the per AP load and backbone routing delay into a single metric, we use the *airtime metric* as the main association decision variable. The airtime metric was previously considered in the recommendation of IEEE 802.11s task group for the use of Radio Metric - Ad Hoc On Demand Distance Vector (RM-AODV) protocol as the default routing protocol. Airtime metric reflects the load on a wireless router (AP) in terms of the average delay a transmission of a unit size packet experiences. In our proposed *cross-layer* association protocol, a client calculates the airtime costs of all APs in its vicinity and it associates with the AP that provides the minimum airtime cost. In the calculation of the airtime cost, the client combines the airtime cost corresponding to the per-AP channel conditions, and the airtime cost corresponding to the end-to-end delay between the candidate ingress and egress APs for the client's connection. Thus, the association decision based on the end-to-end network performance has the promise to support QoS-sensitive services and applications such as Voice over IP.

However, real wireless mesh networks are neither completely coordinated nor completely uncoordinated. In real mesh networks, some of the adjacent APs operate in the same channel while the rest may operate on different orthogonal channels. For such networks, we propose a *hybrid* association mechanism that brings together the best aspects of the two proposed association mechanisms. The hybrid association policy adapts to the varying operational conditions of the network, and depending on the observed load of the network the client chooses to associate based on the channel quality or the airtime cost of the APs. Our simulation results demonstrate that hybrid policy can provide high performance under all network conditions.

1.2 Handoff in 802.11 Wireless Networks

The IEEE 802.11 [42] *wireless local area networks (WLANs)* were originally designed to give a solution to the significant problem of tangled cables of the end user devices. The stations (STAs) are wirelessly connected to the available access points (APs) and the APs are connected to a wired backbone network. The evolution of these networks include *mesh networks* where a wireless backbone is set up in order to support end-to-end wireless user communication [3]. No matter whether the backbone is wired or wireless, the STAs must somehow associate with an AP in order to get network connection. During the handoff procedure, a STA must scan all the available channels for a specific period of time in order to be aware of all the active APs in the neighborhood. Then, it must decide which AP is the optimal for the handoff following some optimization criteria and start a negotiation with this AP in order to become part of the network.

The described procedure introduces significant delays. Under the existing technology, the STA must spend enough time in each channel in order to be sure that it is aware of all the available APs that operate in the specific channel. Moreover, it must repeat this process for all available channels. The average scanning delay is 250-500 *msec* (depending on the 802.11 hardware that is used) [6]. These delays generate a significant problem in the association procedure. The situation is even worse if we consider that the same schemes are used in the handoff phase. Ideally, in a handoff scenario we would like the STA to move from one cell to the other seamlessly. It is obvious that this is impossible with the existing technology due to the delays we described above.

In this thesis ([9], [10]) we propose a cooperative handoff framework that can be applied in both *WLANs* and *wireless mesh networks*, and speeds up the basic handoff procedure. The scheme is independent from the underlying association/handoff decision protocol that is used in the network. In this framework we utilize mechanisms for information sharing and radio measurement defined by 802.11k [45]. The STAs that initialize a handoff procedure, take advantage of 802.11k-based mechanisms and cooperate with neighboring STAs/APs in order to exchange significant information. In this way we avoid sequential channel scanning and AP probing. The main outcome of our

framework is that it eliminates the delays that are introduced in the system during the 802.11-based scanning/probe phases. Therefore, it efficiently supports seamless STAs handoff from one cell to another.

1.3 Multiple AP Association in 802.11 Wireless Networks

ADSL has become the 'de-facto' standard for residential broadband access to the Internet. In addition, the density of ADSL deployments with WLAN connectivity tends to be high, specially in urban areas [39]. The interplay between these two technologies introduces interesting technical challenges and opportunities that can be exploited.

First, WLAN access rates are typically an order of magnitude higher than that of typical ADSL connections, which usually are the bottleneck in the end-to-end path [87]. Second, the set of ADSL links in the neighborhood are generally under-utilized [87]. As a consequence, the potential of bonding the APs backhaul bandwidth via wireless is attractive.

However i) APs usually operate on independent radio-channel and ii) users typically connect to these APs with single-radio commodity 802.11 cards. Because a single-radio card cannot simultaneously connect to more than one AP, a solution is to rely on the standard 802.11 Power Saving (PS) mode to implement a Time-Division Multiple Access (TDMA) by sequentially cycling through the APs in a round-robin fashion [24].

Unfortunately, such a TDMA approach hurts the TCP performance by increasing the observed round-trip-time (RTT). To illustrate this effect, let us consider the scenario in Fig. 1.1, where a station is connected to N APs. We focus on the time that the station spends connected to one of them, say AP_1. We denote the *wireless period* as the total amount of time to cycle through all the APs, and the *duty cycle* as the percentage of time the station spends on AP_1. At the beginning of the duty cycle, the station connects to AP_1 and starts receiving the buffered TCP data packets. While connected, the station normally receives TCP data and sends back TCP ACKs, that arrive to the server with a certain end-to-end wired delay. These TCP ACKs will trigger the transmission of new

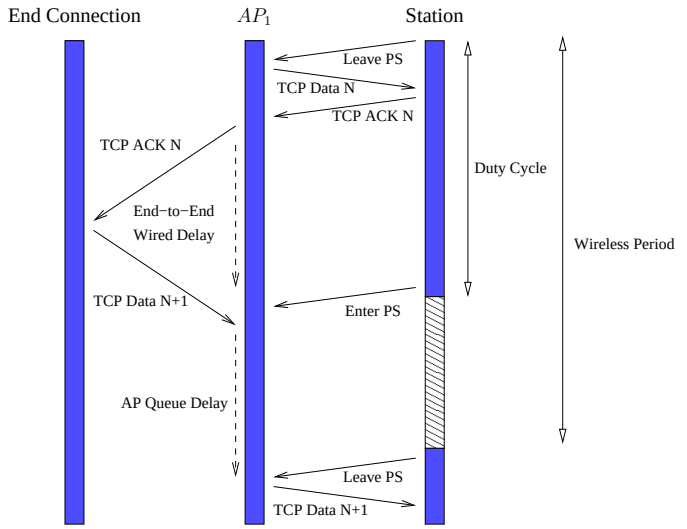


Figure 1.1: Relation between TCP congestion control and time-division access to multiple APs.

TCP data from the sender that may arrive to AP_1 right after the station has already moved to the next AP. Then, these packets will be buffered by AP_1 until the station connects again to it. As we can see, the length of the connectivity and non-connectivity periods may result in TCP packets observing a RTT artificially larger than the actual end-to-end delay.

Motivated by the aforementioned problem, the main contributions of our work [35] are:

1. an in-depth analysis of the effect of TDMA wireless multi-AP access on TCP. We perform numerous experiments using our wireless multi-AP access implementation introduced in [36] over commodity hardware.

2. an analytical model of the problem that accurately fits the experimental results.

3. a low cost resource allocation algorithm, named *min-max disconnection time*, that minimizes the impact of TDMA access on TCP by distributing the station connection to the APs on short periods. We evaluate our solution via extensive simulations and we show that is very close to the theoretical upper bound.

1.4 Channel Allocation in WLANs

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 to increased amounts of interference and contention among co-channel access points (APs) [2]. 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 consider the efficient allocation of frequency bandwidth (i.e., the available channels) to the APs and their clients (i.e., the AP cells) [49], [81], [68], [59], [69]. 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 our contribution in this thesis ([13], [11]), we propose LAC, a Load-Aware Channel assignment scheme that discovers the channel assignment that approximately *maximizes the AP through-*

put, i.e., the sum of throughputs achieved by all its affiliated clients. LAC employs the *airtime cost* metric [8][1]. 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 regards 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 regards 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 [7] we have analyzed the relationship of the airtime metric with the uplink channel long-term throughput. In this thesis 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 [72]. 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, bi-directional 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 [49] 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.

[1]The *airtime cost metric* was proposed in [43] as a routing metric (RM-AODV) and in [8], [7] for optimal user association in 802.11 networks.

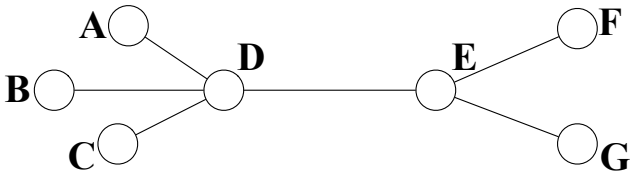


Figure 1.2: Load-aware channel selection.

1.5 Channel Selection in Multi-Radio Mesh Networks

Wireless mesh networking has been touted as the new technology that can support ubiquitous end-to-end connectivity. In wireless mesh networks, information has to be routed via multiple wireless hops before it can reach the destination [66, 63]. 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. We ask the question: *How can we allocate frequencies in a mesh network, in order to maximize the total network throughput, in a distributed manner?*

In order to efficiently allocate the set of available channels to nodes, the load at each individual link needs to be taken into account. Here, the load at the mesh access and backhaul levels could be represented in various ways [8, 49], potentially involving the number of neighbor transmitters, the traffic demand, the amount of traffic flowing through each node, etc., as we discuss later. Previous studies on frequency selection, however, do not consider the *end-to-end load distribution* across entire routes; they consider the sub-problems of either the access level, between clients and access points (APs) [49, 81], or the backbone portion of the network [69, 79, 5]. We argue that frequency selection algorithms should prioritize the assignment of low-interference channels at highly-loaded mesh links, both at the access and the backhaul levels. As a simple example, consider the connectivity graph of Fig. 1.2, where nodes A, B and C generate equal amounts of traffic towards node E, while the same channel is initially used by all links. In this scenario, the DE link facilitates the aggregated traffic, destined to node E. Thus, DE should be assigned a frequency such

that the aggregate traffic towards E is efficiently forwarded, i.e., the bottleneck situation with DE, due to potentially low SINR or high contention levels, is avoided to the extent possible. Note here that the frequency selection outcome will likely affect the decision of load-aware routing protocols, such as RM-AODV (Radio Metric Ad Hoc On-Demand Distance Vector Routing) [43]. Hence, both the frequency selection and load-aware routing functionalities are inter-dependent and must be considered in conjunction.

In this thesis [12], we propose ARACHNE, a load and routing aware channel selection protocol for wireless mesh networks. ARACHNE performs end-to-end channel selection along a route, by adopting a variation of a load characterization metric [8]. Given that the load-aware routing choices are affected by the frequency selection policy, ARACHNE combines frequency selection and route selection under the same unified framework. To the best of our knowledge, this is the first work that presents a frequency selection protocol across entire routes, involving both the access and the backhaul levels in conjunction with the load-aware routing of information between end-hosts. We evaluate ARACHNE through trace-driven simulations, using the OPNET [72] simulation platform. We observe that ARACHNE outperforms other channel allocation mechanisms for mesh networks in terms of overall network throughput, average delay and dropped data.

Chapter 2

State of the Art

2.1 Association Schemes

In IEEE 802.11 [42] standard, the association procedure consists of three phases. In the first phase, the un-associated STAs (e.g., STAs just turned their power on) scan the medium and listen to the beacons of the available APs (*passive scanning*). The STAs use the information broadcasted by the APs (in their beacon frames or *Probe Response* frames) to make their association decisions. Similarly, a STA may choose a more active approach and send *Probe Request* frames, to which available APs respond by *Probe Response* frames (*active scanning*). During the second phase, the STA determines the AP that is the most appropriate to associate with. Finally, in the third phase, STA sends an *Association Request* frame to the selected AP (Association phase). The AP responds with an *Association Response* frame, and an authentication process starts. If the authentication is successful, the STA becomes part of the network and is able to communicate with other STAs. In the standard, the information that is used during the second phase is defined as the received signal strength of the management frames transmitted by the APs, and the STA associates with the AP that has the highest Received Signal Strength Report Indicator (RSSI).

RSSI is not an appropriate decision metric for user association for several reasons. High RSSI values cannot univocally indicate high throughput. This is because, RSSI not only depends on the

distance from the APs, but also on the transmission powers of the APs. Also, the traffic between the STAs and the APs is usually bi-directional, but RSSI is an indicator for the downlink but not for the uplink channel conditions. Moreover, since the wireless channel is a shared medium, throughput depends on the population of the cell served by the AP. Consequently, an AP may become overloaded if large number of stations are associated with it.

IEEE 802.11 standard defines RSSI-based association policy for WLANs. It is well known that this association policy can lead to inefficient use of the network resources [6], [17], [18]. Therefore, there has been increasing interest in designing more efficient STA association policies. In [18], the authors study a new STA association policy that ensures network-wide max-min fair bandwidth allocation to the users. The work in [49] presents self-configuring algorithms that provide improved client association and fair resource sharing in the wireless network. The authors in [64] propose a new handoff mechanism that is based on monitoring of the wireless links. In [6], there is a detailed analysis of the 802.11 handoff process, and the authors optimize the probe phase of the association process in order to reduce the probe latency. The system presented in [17] ensures fairness and QoS provisioning in WLANs with multiple APs. The authors compare their system with 802.11 and prove that 802.11 cannot support both fairness and QoS guarantees. In [53] the problem of optimal user association to the available APs is formulated as a utility maximization problem. The technique that is presented in [78] reduces the cost of discovering new APs by synchronizing the management packets transmitted in the network. In [85], neighbor and non-overlap graphs are utilized in order to reduce the probing latency. The "multi-homing" scenario is introduced in [83], where the traffic is split among the available APs. In [83], the throughput is maximized by constructing a fluid model of user population that is multi-homed by the available APs in the network. In [21], the authors have designed and implemented a new handoff scheme which exploits multiple radios in wireless devices. The system presented in [16] handles user service requests by readjusting the load across all APs. The proposed mechanism is named network-directed roaming, since if an AP cannot meet the needs of a client, the network suggests the client another location in the network where the request can be satisfied. In [4], the authors propose new protocols applied in a new architecture

called SMesh. This architecture supports fast handoff introducing collaboration between the APs in the network. In [75] the authors propose a new pre-authentication mechanism that speeds up the handoff process in wireless mesh networks. Lastly, the authors in [55] and [56], present a dual-association approach in wireless mesh networks, where the APs for unicast traffic and the APs for broadcast traffic are independently chosen by exploiting overlapping coverage. In this way they optimize the overall network load.

The last four approaches [4] -[56] are applied in wireless mesh networks and so they are close related to our work. The first two of them try to speed up the handoff process introducing collaboration between the APs and pre-authentication. It is clear that they don't work in the direction of improving the performance of the association decision of the STAs in the network. Our work optimizes the association decision, taking into account the channel quality and the load in the network. In the last two approaches the authors improve the network load using traffic categorization and dual association on top of the RSSI-based user association process. Our approach is totally different while we support single association and we incorporate the channel quality in the association process, replacing in this way the RSSI-based association procedure. We must mention that in order to apply most of the previous approaches we must introduce significant modifications in the way that 802.11 works and so these approaches are not compliant with the 802.11 standards. Contrarily, our work is fully compliant with the 802.11 standards.

2.2 Handoff Schemes

IEEE 802.11 defines association/handoff procedures based on *RSSRI (Received Signal Strength Report Indicator)* measurements. The unassociated STAs or the STAs that are trying to reassociate with a new AP, initialize a scanning process to find the available APs that are placed nearby. During this scanning process, the STAs sequentially switch to the available operational frequencies in order to probe the APs and receive their information. They measure the *RSSRI* values of each AP and associate with the AP that has the highest *RSSRI* value (the strongest received signal). The

authentication process follows.

Several studies have proven that the *RSSRI*-based association/handoff mechanism can lead to poor network performance while the networks resources are not utilized efficiently [17], [6]. Therefore, the research community focuses on designing new association/handoff methodologies that will provide better resource utilization in the network. In our previous work [8] we have introduced new dynamic association and reassociation procedures that use the notion of the *"airtime cost"* in making association/handoff decisions. This metric reflects the uplink/downlink channel conditions and the traffic load in the network. The cross-layer extension of this mechanism takes under consideration the routing-based information from the mesh backbone. Consequently, the STAs are based on this information to optimize their association/handoff decision.

In [18], the authors study a new STA association policy that guarantees network-wide max-min fair bandwidth allocation in the network. The system presented in [17] ensures fairness and QoS provisioning in WLANs with multiple APs. The work in [49] proposes an improved client association and a fair resource sharing policy in 802.11 wireless networks. In [51], the authors propose an association scheme that takes into account the channel conditions (the channel information is implicitly provided by 802.11h [44] specifications). In [53] the problem of optimal user association to the available APs is formulated as a utility maximization problem. The work in [83] proposes a new mechanism where the traffic is split among the available APs in the network and the throughput is maximized by constructing a fluid model of user population that is multi-homed by the available APs in the network.

The papers mentioned above study optimal STA association mechanisms in the network. On the other hand, a lot of attention has been given in reducing the delays introduced during the association/handoff procedure. The authors in [6] describe in detail the main factors that cause those delays:

- **Probe or scanning delay**: During the first step in the association/handoff procedure that is determined by 802.11 a STA have to scan for available APs: a) passively, by listening to their beacon frames or b) actively, by probing the APs. These are time consuming procedures since

the STA must scan all the available channels (12 for 802.11a) in order to find active APs. Furthermore, the STA has to follow the beacon intervals for data synchronization reasons. Scanning delay constitutes a major portion of the handoff delay.

- **Association/Handoff delay**: When a STA associates with an AP, it has to exchange *association frames* with this AP. Similarly, when a STA moves from an AP to a new AP, it has to exchange *reassociation frames* with the new AP.

- **Authentication delay**: A STA has to exchange *authentication frames* in order to be authenticated by the new AP.

The following approaches attempt to reduce those delays and they are closely related to our work in this thesis: The authors in [6] propose a technique to eliminate the probe phase delay of the association process. The work in [86] proposes a selective scanning algorithm and a caching mechanism in order to reduce the delay introduced by the scanning phase. Selective scanning uses a channel mask and therefore the STAs scan a small subset of the available channels (using this channel mask). In particular, when a STA scans APs, a new channel mask is built based on the current scanning status. In the next handoff, during the scanning process, this channel mask will be used. Consequently, only a well-selected subset of channels will be scanned. In [85], the authors formulate the association problem using neighbor and non-overlap graphs. In [21], multiple radios are used in order to implement more effective/fast handoff mechanisms. Management frame synchronization is the basic part in the proposed mechanism presented in [78] while monitoring of the wireless communication links is the basic component of the proposed handoff mechanism in [64]. In [67], the authors present a proactive association scheme based on a distributed cache structure that speeds up the association procedure. Another approach that reduces the handoff delay is proposed in [91]. In this work the channel scanning is performed proactively and smart triggers reduce service disruption time in the system. The authors in [4] present a new mesh network architecture called SMesh. In this architecture they provide fast handoff procedures. In [34], the authors design client-driven handoff techniques that support vehicular mobility in multi-hop wire-

less mesh networks. In their work they use channel quality measurements in the handoff decisions and they employ mechanisms to control handoff frequency. An interesting approach called Cooperative Roaming (CR) is proposed in [31]. This work is very relevant to our work, while the authors introduce cooperation in order to perform layer 2 handoff, layer 3 handoff and authentication. In their approach the STAs subscribe to multicast groups in order to spread useful information in the network. Our thesis focuses especially on mesh networking deployments, where a large number of clients must be supported and the provided QoS should be high. In these highly congested environments multicast communication is inefficient. Consequently in our work we follow a different approach in which we utilize 802.11k measurement techniques that are adaptively applied in mesh deployments and can be applied in WLANs too. Finally, in [73] there is an interesting study of different fast handoff mechanisms.

Our work in this thesis eliminates the delays in the first part of the handoff procedures (scanning and probing delays). It is worth mentioning that in our 802.11k compliant *client-based* framework the STAs "govern" the handoff procedures. This differentiates our contribution from other approaches in literature (like in [4]) where the APs are the responsible entities for the execution of the association/handoff procedures.

2.3 Multiple Association Schemes

The need for resource allocation schemes in 802.11 has been extensively studied in the literature [33]. Many of the proposed schemes either rely on non-standard 802.11 features [29], or completely develop an entire new MAC protocol [84]. Both approaches may be undesirable, and so we avoid them. Given that, the resource allocation scheme that more closely relates to ours is [80]. There, the authors study the problem of absence of application-specific resource allocation schemes in 802.11. As a solution, they design and implement an overlay MAC layer (OML), to divide the time into slots of equal size, and then use a distributed algorithm to allocate them across the competing nodes, where each competing node receives a number of slots proportional to its weight function.

However, the authors let as an open issue the understanding of the increased delay for TCP traffic in presence of the slot mechanism [80].

Although overlay solutions are easy to be implemented, they are often sub-optimal and difficult to scale because of the overlapping and duplication of similar functionalities at different layers (e.g. in the driver and in the card firmware). The VirtualWiFi project [24] proposes an architecture that abstracts a single WLAN card to appear as multiple virtual cards to the user. Each card instance adopts standard PHY/MAC protocols, but can be separately configured at the driver level. the possibility to tune each card on a different orthogonal channel, in order to improve the overall available capacity.

An interesting application was the idea of connecting to multiple APs through a single radio interface. The authors rely on the Power Save (PS) mode feature of the WLAN standard to switch among different Wi-Fi nodes (in AP mode and/or Ad-hoc mode) in a time-division fashion. A station can inform the current Wi-Fi node that it is going into PS mode — so that it can buffer packets directed to it — and switch radio-frequency to other Wi-Fi nodes, only to come back to the original node before the PS period expires.

FatVAP [47] studies the problem of ADSL bandwidth aggregation via wireless connectivity. The authors introduce a scheduler to select the percentage of connection time on each AP to maximize the aggregate throughput at each station. The solution leverages on the fact that the high speed wireless card at the station needs to be connected on each AP for a short period of time in order to collect all the pending data. FatVAP, however, does not explore TCP latency-related problems.

Finally, Juggler [70] proposes an architecture similar to one in [47] and focuses on the support for a seamless hand-off between WLAN APs.

2.4 Channel Allocation Policies for WLANs

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 [32] 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 [68]. Similarly, Leith and Clifford [58] 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, Mhatre et al. in [49] 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, i.e., 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 [49].

Moreover, the work in [69], by Mishra et al., belongs to a set of studies that propose a distributed channel hopping mechanism. The mechanism in [69], MaxChop, provides higher levels of fairness among users. Channel hoping, 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 [90]; this is prohibitive in terms of incurred overhead.

Lee et al. [54] 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 [81] 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 [68], [59]. Mishra et al. [68] 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 et al. [59] present a formulation of the channel assignment problem for 802.11 WLANs, which is then proven to be NP complete. Then, the design and analyze a heuristic algorithm that attempts to minimize the effective channel utilization for the bottleneck APs.

Efficient channel selection is essential in 802.11 mesh deployments too, for minimizing contention and interference among co-channel 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. [5] study the joint channel allocation and routing problem, assuming that traffic demands and network topology are known. They present a LP formulation of the problem and they propose a centralized algorithm that maximize the aggregate throughput.

Raniwala et al. propose in [79] 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. [77] propose a measurement-based centralized approach to provide effi-

cient channel allocation for radios. They perform channel-to-interface assignment based on channel reuse possibilities which in turn depend on interference.

The previous three 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 in this thesis embraces the aforementioned parameters under a common frequency selection framework, since there are no approaches in 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, select the channel that will provide approximately maximum long-term throughput in the network.

2.5 Channel Selection Policies for Mesh Networks

In wireless mesh networks there is an important interest in the research community in designing new channel selection policies. However, the ”mesh” technology is quite new and it is obvious that these studies are limited. Kauffmann et al. in [49] propose a distributed frequency selection algorithm that minimizes the global interference in the network. However, their algorithm considers WLANs only. Similarly, Rozner et al. in [81] propose a channel assignment scheme for WLANs, taking into account traffic demands. The MaxChop mechanism [69] provides high levels of fairness among users using channel hopping. However, it requires tight synchronization between AP and clients, while it is difficult to implement efficiently with off-the-shelf hardware. In [5] the authors study the joint channel allocation and routing problem, assuming that traffic demands and network topology are known. They propose a centralized algorithm that maximizes the aggregate throughput. Raniwala et al. [79] propose a tree-based mesh architecture, Hyacinth, where the local channel load information exchange facilitates the channel selection. Hyacinth tries to address the joint problem of channel assignment and routing in wireless mesh networks. The latter two

approaches, [5, 79], however, assume the availability of a global network view.

Our work (ARACHNE) differentiates from these approaches by providing efficient end-to-end channel selection in a distributed manner (access level and mesh backhaul). In ARACHNE there is no need of synchronized channel access. Moreover our protocol does not employ any tree-based architecture, which in some cases cannot represent the actual network topology and its dynamics. Finally, our approach is fully compliant with 802.11s wireless mesh networks and it can be implemented on top of the existing IEEE 802.11 standards.

Chapter 3

Cross-Layer Association in 802.11-based Wireless Networks

In this chapter we present a new association framework for optimal user association and network performance. Section 3.1 briefly discusses the main problems in the implementations of the current 802.11 association scheme. In Section 3.2, we describe our proposed user association and handover method based on the estimation of the channel quality. In section 3.3 we consider the end-to-end QoS-aware cross-layer association scheme for wireless mesh networks. Section 3.4 presents a hybrid association mechanism. In sections 3.5 and 3.6, we present the simulation and the experimental evaluation of the proposed association schemes.

3.1 Association Schemes

3.1.1 New Metrics for User Association

In this section, we investigate important parameters that need to be taken into account for associating STAs to available APs in an efficient manner. It is clear that the selfish objective of the STA is to associate with the AP that provides the maximum average throughput. Therefore, the STA needs to first estimate the average throughput it can obtain from the available APs. For this purpose, we

consider two scenarios that may be observed in real life. In the first scenario, the wireless network is coordinated and the APs are assigned channels so that no two adjacent AP interfere with each other. In the second scenario, we consider completely uncoordinated network working in the worst possible way, i.e., all APs in the network operate at the same channel. We explore the important parameters in these two scenarios and the lessons learned in this section are used to design the association schemes proposed in the latter sections.

APs operating on orthogonal channels

In this section, we will discuss the efficacy of using the *airtime metric* for making association decisions [43]. The airtime metric is currently proposed in the 802.11s wireless mesh networking standard as a default routing metric. Briefly, the airtime metric shows the current load of the channel. Formally, the airtime metric of station $i \in U_a$, where U_a is the set of stations associated with AP a, is given as:

$$C_a^i = \left[O_{ca} + O_p + \frac{B_t}{r^i} \right] \frac{1}{1 - e_{pt}^i}. \tag{3.1}$$

In (3.1), 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. Some representative values for these constants (for 802.11b networks) are: $O_{ca} + O_p = 1.25$ms and $B_t = 8224$bits. The input parameters r^i and e_{pt}^i are the transmission rate in Mbs, and the frame error rate for the test frame size B_t, respectively. The rate r^i depends on the local implementation of rate adaptation, and represents the rate at which the mesh point would transmit a frame of standard size B_t based on the current conditions. The estimation of e_{pt}^i is also a local implementation choice, and it is intended to estimate e_{pt}^i for transmissions of standard size frames B_t at the current transition rate r^i. Note that the airtime metric reflects the radio resources used by STA i.

In order to understand the reasoning behind the airtime metric, we should refer to the recent work investigating the calculation of the average throughput of the 802.11 based WLANs [20], [38], [40], [52]. It has been observed that when there are several flows with different physical trans-

mission rates, then the throughput of all flows is bounded by the slowest transmission rate [19]. In order to explain this anomaly, further studies have been conducted, and the average uplink throughput in a single cell environment is calculated under saturation and decoupling approximations in [20] and [52]. 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 lengths are equal to L. As derived in [52], the total average network throughput $\theta(\beta)$ is given as:

$$\theta(\beta) = \frac{n\beta(1-\beta)^{n-1}L}{1+\sum_{i=1}^{n}\beta(1-\beta)^{n-1}(\frac{L}{C_i}+T_0)+(1-(1-\beta)^n-n\beta(1-\beta)^{n-1})\,T_c} \tag{3.2}$$

where β is the attempt rate (probability) in the equilibrium, C_i is the physical transmission rate of node i, T_0 is the fixed overhead with packet transmission and T_c is the fixed overhead for an RTS collision. 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:

$$G(\gamma) = \frac{\sum_{k=0}^{K}\gamma^k}{\sum_{k=0}^{K}\gamma^k b_k}, \tag{3.3}$$

where K is the maximum number of attempts allowed under 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 $\Gamma(\beta) = 1-(1-\beta)^{n-1}$ due to the decoupling approximation. The equilibrium behavior of the system is governed by the solution of the fixed point equation $\gamma = \Gamma(G(\gamma))$. If this equation is solved it yields the collision probability from which the attempt rate in the equilibrium β can be determined from (3.3).

Note that $\frac{L}{\theta(\beta)}$, is the average delay per packet in the equilibrium, which is given as:

$$\rho(n) = \frac{L}{\theta(\beta)} = \frac{1}{n\beta(1-\beta)^{n-1}} + (T_0 - T_c) + \frac{1-(1-\beta)^n}{n\beta(1-\beta)^{n-1}} T_c + \frac{1}{n}\sum_{i=1}^{n} \frac{L}{C_i} \tag{3.4}$$

The first three terms in (3.4) represent the delay due to channel contention and protocol overheads, and the last term represents the average transmission time of an L length packet by a node in the cell.

Comparing the definition of the airtime metric C_a^i and $\rho(n)$, we can see that the sum of O_{ca} and O_p correspond to the approximation of the first three terms in (3.4) with a constant value. Clearly, the transmission of RTS, CTS, DATA and ACK frames may also get corrupted not only due to collisions but also due to channel errors. The authors in [38] extended the analysis in [20] to account for wireless bit errors. The goodput expression in [38] assumes that the wireless channel can be modeled by an appropriate Gilbert model with known transition probabilities. In wireless networks with dynamically changing conditions, such an assumption is not practical. Therefore, in our method, the current frame error rate, e_{pt} is measured by the users and AP. For each packet attempted to be transmitted, it would be in error due to channel errors with probability e_{pt}. Clearly, the average number of attempts until successful transmission would be $1/(1-e_{pt})$. For each attempt on the average $\rho(n)$ amount of time is experienced. The product of $1/(1-e_{pt})$ and $\rho(n)$ gives the airtime metric used in our algorithm.

When the overhead is calculated as described in (3.4), we observe that it varies with the number of nodes in the cell. The overhead with respect to the number of nodes is given in Figure 3.1, when $K = 7$, $b_0 = 16$, $b_k = 2^k b_0$, the slot length is $20\mu s$, $T_0 = 52$ and $T_c = 17$ slots. The overhead varies between 59 and 66 slots, and thus, it is clear that there is weak dependency between the number of users n and the observed overhead. Therefore, in our algorithms the overhead is taken as a constant to avoid additional computational complexity.

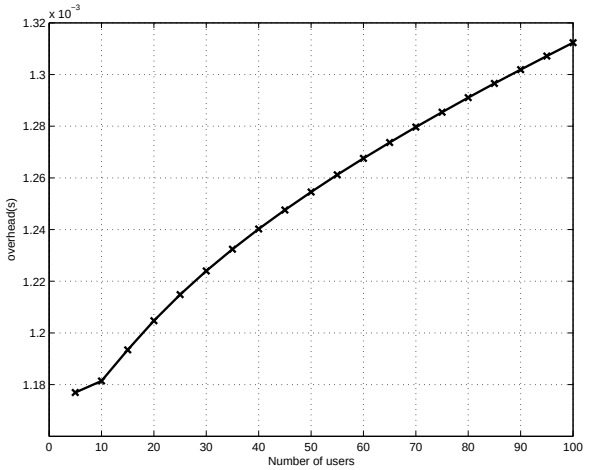


Figure 3.1: The contention and protocol overhead.

APs operating on the same channel

Aforementioned discussion presupposes that all APs within the carrier sense range of each other are working on independent (orthogonal) wireless channels, so that a transmission in one cell do not affect transmission in the other cell. Consequently, the average throughput depends both on the number of users in the cell and the quality of the wireless link between the user and AP.

However, in many practical cases, APs operate in the same channel. In such a case, the STA that is in the carrier sense range of more than one APs has to remain silent during the transmissions of other users in each of these cells. Moreover, if all nodes in the network cannot overhear each other (i.e., all nodes are within the same carrier sense region), the basic Distributed Coordination Function (DCF) cannot prevent collisions from *hidden nodes*. It is well known that the presence of hidden nodes in the network reduces the network throughput significantly [38]. There have been a few attempts for analyzing the throughput in 802.11 wireless LANs in such a setting [38, 40]. For the sake of simplicity of the analysis, it is assumed that the wireless channel is perfect and there is no bit errors. We adopt the approximate calculation of STA throughput given in [40] for our purposes. It has been shown that the average throughput of STA B transmitting to AP A can be

given as:

$$\theta'_A(\beta) = \frac{\beta(1-\beta)^{n_A-1} L\, P_1\, P_2}{T_{avg}}, \tag{3.5}$$

where P_1 and P_2 are the probabilities that there exists no hidden node problem during RTS-CTS and DATA transmissions respectively, and n_A is the number of nodes in the carrier sensing region of AP A. Also, T_{avg} is the average time spent during collision due to another node (hidden or not) within the carrier sense range of B transmitting, successful transmission of a node other than B, idle period and successful transmission of B. It is clear that P_1, P_2 and T_{avg} depends on the topology of the network, and it is practically very difficult if not impossible to determine them in a real network setting.

We assume that if a STA can associate with more than one AP, all the stations in these cells can hear each other's transmissions. Therefore, from (3.5), the average throughput of the STA is the same regardless of the AP it associates with. This assumption helps simplify the association algorithms, but clearly it is not satisfied for most real cases. In our numerical studies, we study the performance of the algorithms when there are hidden nodes, and show that the performance of the algorithms is not hampered by this assumption.

Consequently, the only important parameter that affects the association decision is the quality of the link. If a STA has better quality link to one of the APs, then associating with that AP reduces the number of retransmissions due to bit errors, and thus, increases the average throughput of the STA.

In Figure 3.2, we can have an intuitive view of the previous discussion. Since 802.11 uses a shared medium, what is really important for a STA is not the load of the associated AP but the load of the STAs in its coverage area. For example, STA3 is affected by the load of AP1 and AP2 no matter with which AP it is associated. Obviously in this scenario an association mechanism that is based on the load [8] will force the STA to associate with AP1. Nevertheless, STA3 will not have the chance to transmit to AP1, no matter how "light" the load of AP1 is. The key point in the

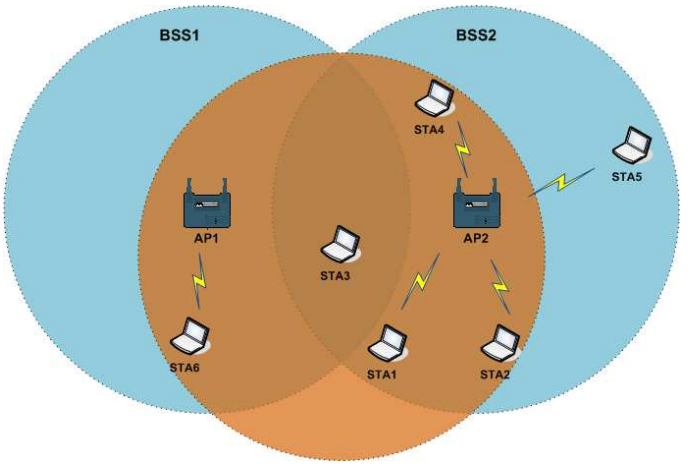


Figure 3.2: An association scenario (APs operate on the same communication channel).

association of STA3 in order to improve its performance (and of the overall network performance) is to take into account its own transmission rate and the packet error rate, i.e., the quality of the channel between STA3 and the AP that will be associated with.

3.2 Channel Quality based User Association

Based on our observations in Section 3.1, we now develop two different association algorithms. The first algorithm is more suitable for networks where there is no coordination and APs operate at the same channel.

The QoS perceived by a client depends mainly on the uplink and downlink rates of the client. Meanwhile, uplink and downlink rates depend on the interference levels observed at the AP and the client locations as well as the transmission power levels. The interference occurs due to the ongoing transmissions in adjacent cells as well as external radio devices (such as radar, HyperLan, appliances, etc). An association decision that takes into account all these factors can clearly provide better user satisfaction.

The IEEE 802.11h [44] protocol provides new features that can be used to effectively control the user association and handoff in wireless LANs. The mechanisms introduced in 802.11h provide new procedures for the stations, and the APs to become more aware of the current channel conditions.

Table 3.1: RSSI classes

RSSI class	**Power Observed at the Antenna (dBm)**	**Tolerance (dBm)**
0	Power $<$ -87	+5
1	-87 $<$ Power $\leq$ -82	$\pm$5
2	-82 $<$ Power $\leq$ -77	$\pm$5
3	-77 $<$ Power $\leq$ -72	$\pm$5
4	-72 $<$ Power $\leq$ -67	$\pm$5
5	-67 $<$ Power $\leq$ -62	$\pm$5
6	-62 $<$ Power $\leq$ -57	$\pm$5
7	-57 $<$ Power	-5

The new information available to the stations and APs include the link degradation factor, estimated link margin, maximum allowable transmission level for the stations, and the received signal strength indication (RSSI) measurements.

In WLANs, effective user rate is a function of the average of the uplink and downlink rates. Therefore, we propose a new association procedure that maximizes the bi-directional throughput between the STA and the AP. Our scheme determines the AP that maximizes the transmission rates in **both** directions by considering the link quality in the uplink and the downlink. In our scheme, every STA estimates the signal-to-interference-ratio (SINR) for the uplink and the downlink for every candidate APs. The estimation procedure is given as follows:

For the downlink:

In current implementations, a STA already have information of the received signal strength of the broadcast packets it has received. Operating under the framework of 802.11h, the STA can also perform a "RSSI measurement" to determine an histogram of the RSSI densities as shown in Table 3.1. Using this table, the STA can calculate the mean interference in its neighborhood. Thus, the STA estimates downlink SINR by:

$$SINR^d_{n,k} = \frac{S^r_{n,k}}{I^k_{est}}, \tag{3.6}$$

where $SINR^d_{n,k}$ is the estimated SINR for the downlink between AP n and STA k, $S^r_{n,k}$ is the signal strength observed by the receiver (STA k), and I^k_{est} is the mean estimated interference in the

neighborhood of the STA k. Both the STA and AP periodically performs RSSI measurements and populates the tables such as the one given in Table 3.1. The average of the histogram given by this table gives the estimated interference.

For the uplink:

For the uplink, a STA performs similar estimations. However, for the uplink the STA does not have all the information it needs to calculate (3.6). Therefore, some small but important modifications are required in the procedure. In particular, the STA requires the strength of the received signal at the AP and the table with the RSSI densities measured by the AP in order to estimate the SINR in the uplink.

The received signal strength of an uplink packet in the AP is estimated as the transmission strength of the packet multiplied by the link degradation factor which depends on the physical medium. The link degradation can be calculated (we assume that the link degradation factor is the same for the uplink and the downlink) using the 802.11h framework. According to 802.11h, every AP broadcasts via its beacons the **Local Maximum Transmit Power Level**. This is the maximum limit for the transmission power of every STA belongs to the specific cell. Thus, STA uses this power for the estimation of the SINR in the uplink (assuming that it transmits using the maximum allowable power in the specific cell). On the other hand, the AP now includes in its beacon the power level used in the transmission of the beacon. The STA with the knowledge of the transmission power and the reception strength of the specific packet can calculate the link degradation factor.

Finally, the STA needs the interference RSSI density table measured by the AP in order to estimate the mean interference on the uplink. This information is available in the AP and it can be broadcasted easily to the interested STAs. This information is carried in the beacon frame as a new Information Element. Note that the length of the specific table is only a few bytes, so it does not increase the overhead significantly.

Therefore, the STA estimates the uplink SINR by:

$$SINR^{u}_{n,k} = \frac{\left(C * S^{t}_{k,n}\right)}{I^{n}_{est}} \tag{3.7}$$

where $SINR^u_{n,k}$ is the estimated SINR for the uplink between AP n and STA k, C is the propagation loss of the transmission power, $S^t_{k,n}$ is the estimated transmission power from STA k to AP n, and I^n_{est} is the mean estimated interference in the neighborhood of AP n.

Given the uplink and downlink SINR, the STA can calculate the Packet Error Probability (PER) by using the methods described in [76]. We denote by $1 - p_u$ and $1 - p_d$ the PER for the uplink and the downlink channels, respectively. We characterize the channels as lossy channels. Therefore, every transmitted packet is successfully received with probability p_u and p_d respectively. We estimate the mean number of frames required for a successful packet transmission as follows:

$$\begin{aligned} E(\#frames) &= \sum_{k=1}^{\infty} k(1 - p_u p_d)^{k-1} p_u p_d \\ &= \frac{p_u p_d}{1 - p_u p_d} = \frac{1}{\frac{1}{p_u p_d} - 1} \end{aligned} \tag{3.8}$$

In order to minimize the number of frames required for a successful transmission we should maximize the product $p_u p_d$. Thus, the STA selects the AP that provides the maximum $p_u p_d$.

3.3 Airtime-Metric based User Association

In this section, we propose an association algorithm which takes into account the loads of the APs. This association algorithm is more suitable for coordinated networks where the adjacent APs operate at different channels and do not interfere with each other. The algorithm uses the airtime metric to measure the current uplink and downlink load of the APs. Also, the algorithm takes into account the routing cost in wireless mesh networks. Finally, we propose a load-balancing mechanism used jointly with our proposed association algorithm that helps increase the network efficacy.

3.3.1 End-to-End QoS-Aware Cross-Layer Association Scheme for 802.11-based Wireless Mesh Networks

We propose a load-aware association scheme, where the association decision is based on a new metric. This new metric is called *airtime metric*, and it takes into account many important parameters necessary in making a good association decision. We define the load on the "uplink" channel of a particular AP a by aggregating the radio resources used by all STAs, C_a^i, $i \in U_a$ as:

$$C_a^{up} = \left[O_{ca} + O_p + B_t(\overline{1/r^{up}})\right] \frac{1}{1 - \overline{e_{pt}^{up}}} \left|U_a\right|, \tag{3.9}$$

where $\overline{e_{pt}^{up}}$, $\overline{r^{up}}$, and $|U_a|$ are the average uplink error probability, average uplink transmission rate and the number of STAs associated with AP a respectively. The average error probability and the average transmission rate on the uplink can be computed by the AP by observing the past communications with their associated stations. The AP uses a time window in order to compute these average values. In general, the radio resources are used for both the uplink and the downlink. The load on the downlink is similarly calculated, but this time e_{pt} is different for each downlink associated for every station, $i \in U_a$:

$$C_a^{down} = [O_{ca} + O_p] \sum_{j \in U_a} \frac{1}{1 - e_{pt}^j} + B_t \sum_{j \in U_a} \frac{1}{r^j(1 - e_{pt}^j)}. \tag{3.10}$$

In our implementation, C_a^{down} is calculated by the AP a according to the APs' and its associated STAs' measurements. All APs announce their relevant measurements, and downlink airtime loads, C_a^{down}, in their beacon frames. A new STA chooses to associate with the AP for which $C_a^{down} + C_a^{up}$ is the minimum, since that AP is the one with the least load from the STA's perspective. The main steps of our airtime association scheme are as follows:

- **STEP 1**: The STA calculates the frame error rate e_{pt} based on previous measurements (e.g., by measuring the percentage of the dropped packets in a time window).

- **STEP 2**: APs broadcast in their beacon or probe response frames, the number of users currently associated, the average error rate and the average transmission rate on the uplink.
- **STEP 3**: The STA calculates C_a^{up} for each AP a that is candidate for the association.
- **STEP 4**: The STA receives C_a^{down} from each AP a. C_a^{down} is calculated with respect to the stations "already" associated with AP a. Downlink airtime cost is calculated by the AP because the STAs are not aware of the downlink transmission rates and error rates in each cell. In other words a STA is not aware of the other STAs' downlink channel conditions in each cell.
- **STEP 5**: Finally the STA selects for association the AP with the minimum $C_a^{up} + C_a^{down}$.

Note that the airtime metric not only considers the uplink and downlink interferences through the inclusion of the transmission rates, but also considers the current number of users associated with the APs. Each additional station regardless of how fast it can transmit/receive to/from AP induces a cost to the system by the constant durations to access the channel. The above metric provides a meaningful way to include the effects of the population of the cells.

The aforementioned scheme introduces an association metric which considers the radio resources used both at the uplink and the downlink. Although this association scheme provides significant improvements in user throughput as demonstrated in the forthcoming section, it is not adequate for wireless mesh networks. In a typical 802.11-based wireless mesh network, it is expected that there will be a large population of MAPs (Mesh Access Points) and MPs (Mesh Points) connected to each other, and the level of the end-to-end QoS provided to the users is the most important performance criterion. Among several APs that a STA can associate with, it should choose the one that provides the best end-to-end QoS. In the following we present an association scheme that incorporates the total airtime cost of the path from the source to the destination in the association decision.

The upcoming 802.11s standard introduces the airtime link cost as the default routing metric. The default routing protocol selects the path with the minimum cumulative airtime cost in order to

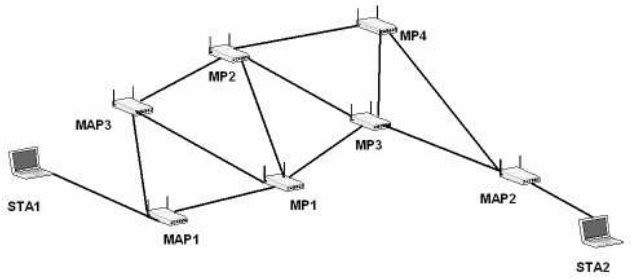


Figure 3.3: A wireless mesh network.

provide the highest data rate to the end-user. In our approach, we combine the routing airtime cost (calculated by the mesh routing protocol) with the association airtime cost.

Our approach is better motivated with an example. In Figure 3.3, we give a sample wireless mesh network, where STA1 is initially associated with MAP1, and STA2 is associated with MAP2. The data STA1 sends to STA2 is routed through MAP1, MP1, MP3, and MAP2, where this route has the minimum "one way" cumulative airtime cost to STA2. Assume that STA1 has the option of associating with MAP1 or MAP3. During the re-association phase, STA1 reconsiders its association. Following the approach that we described, STA1 receives from MAP3 and MAP1 the necessary information to calculate the association airtime costs for each MAP. According to the airtime association scheme, STA1 associates with the MAP that provides the minimum association airtime cost. However, according to our QoS-aware cross-layer approach the association decision of STA1 is affected by the respective routes between MAP1-MAP2 and MAP3-MAP2. During the association phase, if MAP1 and MAP3 inform STA1 about the expected total "one way" routing airtime cost to MAP2, then STA1 can use this information to determine the MAP with the better end-to-end QoS. Moreover, the link asymmetry and the variability of the network conditions can affect the provided end-to-end routing QoS. Therefore, STA1 is periodically informed about the routing airtime costs during the data transfer in order to initiate a re-association when needed.

The Radio Metric-Ad Hoc On Demand Distance Vector (RM-AODV) protocol has been proposed as the default routing protocol in the context of 802.11s mesh networks. In AODV, a Route

Request (RREQ) and Route Reply (RREP) mechanism is utilized for establishing routes between the nodes. Each MAP or MP in the network calculates the airtime metric representing how much it is loaded. This metric is appended to the RREQ and RREP packets. Then, by using the typical AODV protocol, an end-to-end path with the minimum total airtime cost is selected.

Note that although the routing in the mesh network is between the MAPs, data packets are transferred between the STAs associated with the respective MAPs. Therefore, before RM-AODV is executed, the MAPs that the end-users are connected to should be determined. Local Association Base Advertisement (LABA) is a mechanism that informs the entire mesh network about the STAs associated with each of the MAPs [43]. MAPs periodically broadcast LABA frames, which consist of the MAC addresses of the STAs that are associated with the MAPs. Consequently, the MAPs and MPs in the network are aware of the MAP to which each STA is associated with. LABA protocol is used together with the RM-AODV routing protocol, so that the paths are found in an expedited fashion.

In our approach, the association decision of a station is based on the association airtime cost, and the routing airtime cost as calculated during the execution of the routing protocol. The station combines the association airtime cost and the routing airtime cost to choose the most appropriate MAP for the association, e.g., the one with the minimum total airtime cost.

During the scanning phase, the STA sends a pre-association message to the candidate MAPs. This message contains the intention of the station to associate with a MAP, and the address of the receiver STA (or multiple possible receivers) it intends to exchange data with. This message can be incorporated as a separate field in the Probe Request frame. The MAPs that are in the transmission range of the station receive this message. In case their routing tables do not contain any route to the receiver, a route request is initiated. MAPs advertise their minimum “one way” routing airtime cost in an extra field in the Probe Response frame. The total cost reflects the end-to-end QoS that can be provided by the network in case that the station associates with a specific MAP. The total cost is a

weighted sum of the association airtime cost and the routing airtime cost:

$$TC_i^{rcv} = (AC_i^{up} + AC_i^{down})w_1 + RC_i^{rcv}w_2, \tag{3.11}$$

where TC_i^{rcv} is the total weighted cost calculated for MAP i, AC_i^{up}, AC_i^{down} are the association airtime costs for the uplink and downlink respectively (discussed before), RC_i^{rcv} is the routing airtime cost for the path from MAP i to the receiver rcv and w_1, w_2 are the weights. Our simulation results in Section 3.5 are given for $w_1 = 0.6$ and $w_2 = 0.4$, since these values are observed to maximize the throughput of the simulated scenarios. The main steps of our end-to-end QoS-aware association scheme are as follows:

- **STEP 1**: The STA sends a Probe Request frame which contains the addresses of the intended receivers, and calculates the frame error rate e_{pt} based on the previous measurements.
- **STEP 2**: The MAPs broadcast in their Probe Response frames the number of clients currently associated, the routing airtime cost, and AC_i^{down}. AC_i^{down} takes only into account the STAs already associated with MAP i.
- **STEP 3**: The STA calculates AC_i^{up} for each candidate MAP i.
- **STEP 4**: The STA calculates TC_i^{rcv} and selects the MAP with the minimum total airtime cost.

Each STA that is associated with a MAP periodically repeats the aforementioned steps. This presupposes that the information that is broadcasted by the beacon/probe frames is updated dynamically. Thus, during a transmission, a station can initiate a dynamic re-association process if $TC_j^{rcv} < TC_i^{rcv}$, where i is the MAP that the station is currently associated with and j is the new candidate MAP.

It is obvious that the airtime-metric based association scheme is a greedy scheme, where each STA chooses to associate with the AP from which it expects to receive the best performance. In

this setting, the users are non-cooperative and behave selfishly to optimize their own performance. When a new STA associates with an arbitrary AP S, those users already connected to the same AP may experience performance degradation. Thus, this new association may trigger re-associations in the network, since those STAs associated with AP S may now find some other AP with lower airtime cost than AP S. In order to avoid frequent re-associations in the system we have introduced 5% tolerance during the computation of the airtime cost and the re-association process. In other words, when a STA that is associated with an AP "overhears" a new AP with lower airtime cost, it has to check if the new AP will significantly improve its performance. The STA must not initiate a re-association process in case that the new AP will insignificantly improve its performance (lower than 5% improvement). Lastly, it is important to understand whether such a system ever achieves an equilibrium within a finite time. In [30], we investigate the convergence of airtime cost based association scheme by using a novel game theoretical model. In so called "user association game", the user's utility obtained from an AP depends on the number and transmission rates of other associated users. We prove that this game has a Nash equilibrium solution, where no user attempts to change its association given the decisions of other users, and the equilibrium is reached within finite number of steps. Therefore, while this equilibrium is reached there are no re-associations present in our system (if there is no change in the network conditions).

3.3.2 A Sophisticated Weight Selection Mechanism: Network Load Balancing

Our experiments showed that the proposed cross-layer association scheme still falls short of balancing the network load effectively as the network load increases. Therefore, we require an additional mechanism working together with our cross-layer association scheme to achieve the load balancing in the network.

The airtime cost for mesh networks is calculated as given in (3.11). In (3.11), the weights w_1 and w_2 are used to assign appropriate importance to the association and routing costs. The main

idea of our weight selection mechanism is to update the weights of the association and routing airtime costs "on-the-fly" to avoid any overloaded APs.

We now explain the weight selection mechanism in detail. Note that some of the APs in the network can provide routes with low cumulative airtime costs to popular destinations such as mesh nodes with Wide Area Network (WAN) connections. According to our cross-layer airtime-metric based association algorithm those APs are preferred by the STAs, and may become overloaded. In such cases, the weight of the association cost should be increased to deviate some of the STAs to the APs that might have higher routing costs but lower association costs. This idea is similar to cell breathing strategies proposed previously in the literature [18]. The network conditions vary dynamically in time according to the user traffic loads, mobility of the STAs, etc. Therefore, the weight selection mechanism adaptively determines the weights according the network conditions.

In order to update the weights effectively, we first define a balancing index b to measure the network conditions. A similar index was previously used as a fairness measure in [25]. The balancing index b reflects the load conditions in a neighborhood where there are n APs:

$$b = \frac{(\sum_{i=1}^{n} AC_i)^2}{n \sum_{i=1}^{n} AC_i^2}, \tag{3.12}$$

where $AC_i = C_i^{up} + C_i^{down}$ for each AP i. We assume that APs periodically broadcast their AC value to the whole network. This broadcasting can be performed by using the LABA protocol that is already proposed under 802.11s standard. Recall that LABA protocol is used to inform the entire mesh network about the STAs associated with the mesh APs. Also, in our proposed cross-layer association mechanism, we extended the messages conveyed by the LABA protocol by incorporating the airtime costs of APs. Consequently, the mesh APs are aware of the channel conditions of other mesh APs in the network.

Consider an example scenario when there are two neighboring APs. For this case, the load

balancing index is computed as:

$$b = \frac{(AC_1 + AC_2)^2}{2(AC_1^2 + AC_2^2)} = \begin{cases} 1,\ AC_1 = AC_2\ (balanced) \\ \frac{1}{2},\ AC_1 = 0\ or\ AC_2 = 0 \\ 1 > b > \frac{1}{2},\ otherwise \end{cases} \tag{3.13}$$

Note that when the balancing index is equal to 1, $b = 1$, the loads on the APs are the same. In contrast, when the balancing index is equal to $1/2$, then all STAs are associated with only one of the APs and thus, the network is completely unbalanced. Therefore, the balancing index takes values between 1 and $1/2$. We can generalize this observation when there are n neighboring APs. In a neighborhood that contains n APs when b is close to 1 the communication load is balanced and when b is close to $1/n$ then the network is unbalanced.

Clearly, we would like to operate the network with balancing index as close to 1 as possible. However, it is not always possible to achieve the most load balanced network operation in a dynamically changing system. Therefore, we introduce a threshold $1 > T \geq 1/n$, that represents a lower bound on the balancing index corresponding to the balanced network operation. In particular, when $b < T$, the STAs increase w_1 in order to make an association decision based mainly on the association airtime cost (AC). As far as the calculation of threshold T is concerned, the index values of the APs are periodically broadcasted in the network (they are incorporated in the LABA messages). Therefore, each AP calculates its load balancing threshold T as the average of these index values. It is obvious that this threshold oscillates according to the conditions in the network. Figure 3.4 depicts the execution of the heuristic algorithm that performs the load balancing in the mesh network based on the balancing index that we have described before.

During the execution of our heuristic algorithm the mesh APs periodically receive the airtime cost values from the neighboring mesh APs and compute the balancing index b. The mesh APs broadcast this index along with the threshold value in their beacon frames and their associated STAs can "hear" them, extract the carried information and compare b with the threshold. In case $b < T$ there is an unbalanced network operation in this neighborhood, and the STAs increase the

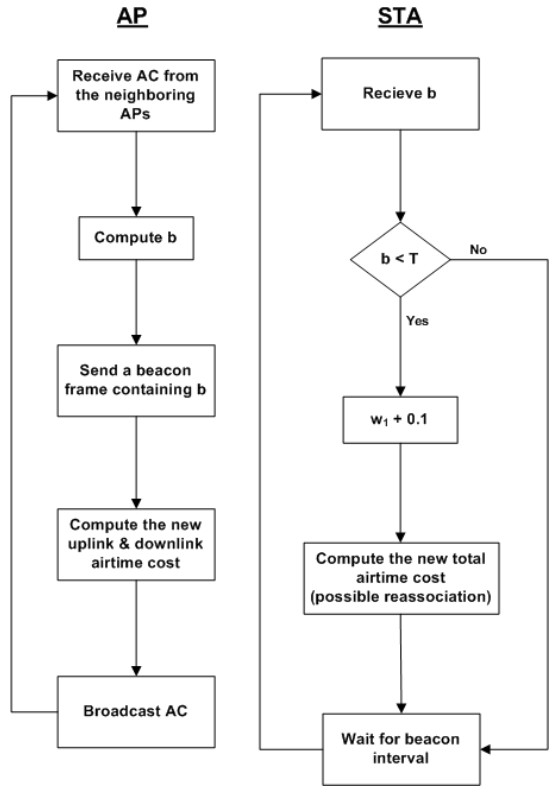


Figure 3.4: The heuristic load balancing algorithm.

value of w_1 computing a new total airtime cost. If there is another AP with lower total cost, the STA re-associates itself with that AP.

3.4 A Hybrid Association Scheme

As we have seen in the previous section, when a STA is in the coverage range of more than one AP operating on the same channel (a common situation in real networks), the STA is generally punished since it has to remain silent during the communication of the neighboring APs with their associated STAs. Consequently, in these conditions the best approach for the STA would be to associate with the AP that has the channel conditions that would allow the STA (each time it gets the right) to transmit with a high rate and with minimum transmission errors. Therefore, neither of the association schemes discussed before can provide sufficiently good performance alone under all network conditions.

In this section, we propose a hybrid association scheme that combines the *channel quality* and *cross-layer airtime metric* based association mechanisms in order to maximize the performance under all network conditions. In hybrid association scheme, we use the cross-layer airtime-metric based mechanism as the default association procedure. However, as the network conditions change in time, we switch from this default mechanism to the channel quality based association mechanism. We describe the basic steps of the proposed approach:

- **STEP 1**: A STA that has just turned on executes the airtime-metric based association mechanism in order to find an AP to be associated with.

- **STEP 2**: The STA periodically estimates the available bandwidth based on the channel occupancy in the shared medium. Thus, during a period T_{est} the STA calculates the channel occupancy ratio as $Ch_r = \frac{T_{channel-busy}}{T_{est}}$. In order to bound this ratio between 0 and 1 we calculate the channel occupancy ratio at time t as $Ch_r^t = Ch_r^{t-1}a + Ch_r^{crnt}(1-a)$, where Ch_r^{t-1} is the last calculated ratio, $a \in [0,1]$ and Ch_r^{crnt} is the calculated ratio during the current T_{est} period. Finally, the STA estimates its available bandwidth as $B_{est}^t = B_{802.11}(1-Ch_r^t)$, where $B_{802.11}$ is the available channel bandwidth in 802.11 networks (eg. 11Mbps).

- **STEP 3**: The STA compares the estimated channel bandwidth with a predefined threshold Th_{mode}. In case that $B_{est} < Th_{mode}$ the STA changes mode and executes channel-quality based association mechanism in order to improve it's performance (the previous can happen reversely). In our simulation studies we have seen that the optimal threshold is $Th_{mode} = \frac{35}{100}B_{802.11}$. We note that we have introduced a $\pm 5\%$ tolerance in order to avoid continuous mode changing situations.

Especially, in highly dynamic networks (dynamic channel allocation is applied, we have mobile STAs, etc.) the STAs should dynamically adapt their behavior in order to improve their performance. Thus, the STAs should execute periodically the previous mechanism in order to change between the airtime-metric based and channel quality based schemes.

3.5 Simulation-based Evaluation

For the evaluation of the proposed protocols, we have implemented our association mechanisms in OPNET [72]. OPNET is a powerful simulation environment and its Wireless Module integrates full protocol stack modeling capability, including MAC, routing, higher layer protocols and applications with the ability to model all aspects of wireless transmissions. The physical layer behavior is simulated in a realistic way through a 13-stage pipeline execution. Using the Transceiver Pipeline, link properties (transmission power, path loss, interference, noise, delay, bit error rate) can be controlled. These link properties are modeled by controlling the simulation settings at each pipeline stage.

We have implemented our association mechanisms using the basic procedures defined by the IEEE 802.11 standard. We have modified the beacon and probe frames in order to incorporate the information elements that our system needs, e.g., transmission error rate, airtime cost, number of associated nodes, etc. We have not changed the packet synchronization procedure, and the small modifications we added for our algorithms, do not affect the overall performance of the 802.11 network. We have completely built all the procedures of our proposed mechanisms into the simulation model. Our approach of "building on-top of the current 802.11 standard" is quite important in order to achieve backward compatibility.

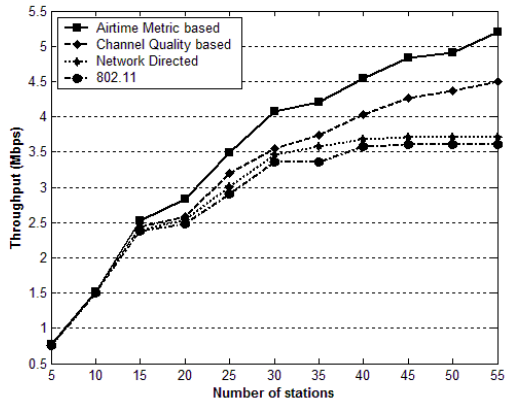
3.5.1 Performance of the Association Schemes

We simulate a multi-cell 802.11 network that consists of four different overlapping cells (4 APs). The STAs are distributed uniformly *at random* in the network. We have chosen this scenario because in simple topologies like this we can control our system parameters and efficiently evaluate the performance. The STAs generate data at a rate of 1024 kbps. Rate adaptation is supported in the simulation model, so the rate of each transmission varies over time according to the channel conditions. We vary the network load by varying the number of active source/destination pairs in the network. For each source and destination pair, the ingress and egress nodes are chosen randomly.

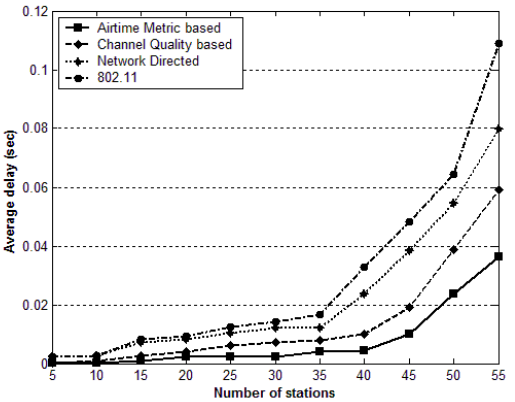
There is no synchronization between stations, so the time that each station is ready to transmit also varies. Therefore, the interference changes dynamically over the duration of the simulation. The beacon interval that is used in our simulation study is 100ms (default interval value in 802.11) and the information that is incorporated in the beacon frames is updated dynamically based on the channel conditions (when the error rate in the channel changes, etc.). Due to the channel variability the re-association frequency is dynamic too and there is no certain re-association frequency. The re-association delay (the delay of each re-association process) that is observed in our simulation study is approximately 50msec. We compare the performance of our association mechanism with that of the association procedure recommended by the 802.11 standard, and with that of the network-directed approach proposed in [16].

We first study the behavior of our mechanisms when the APs operate on orthogonal channels. We measure the network throughput, end-to-end delay and packet drop rate with respect to increasing network load. We execute several different experiments, where the STAs are randomly distributed in the network and the number of the associated STAs in the network increases from 5 to 55. The experiments last for 500 seconds, and in the subsequent figures, we report the average performance of our system.

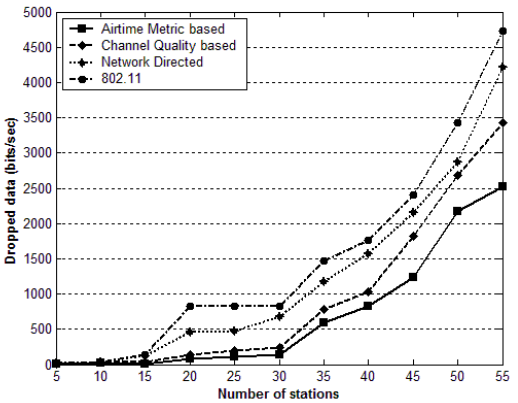
Figure 3.5(a) depicts the average throughput of our mechanisms while the number of the associated stations increases. It is clear that our mechanisms can achieve higher throughput values than the 802.11 recommended mechanism. Under low load conditions the increase in the throughput provided our proposed mechanisms is limited; however, when the network load is high our airtime-metric based protocol achieves approximately 44% increase in the throughput. An interesting outcome is that, as it can be seen in Figure 3.5(a), the maximum 802.11 network throughput is achieved when we have 45 STAs in the network. Meanwhile, the behavior of the network-directed approach is similar to 802.11. In network-directed association mechanism, the STAs are associated with the available APs based on the demands of each STA and the communication load in the network. Unfortunately, the channel conditions for the uplink and downlink communications are not considered in this approach. On the contrary, our proposed schemes maximize the throughput when



(a) Average throughput.



(b) Average transmission delay.



(c) Average dropped data.

Figure 3.5: Simulation results when APs operate on orthogonal channels.

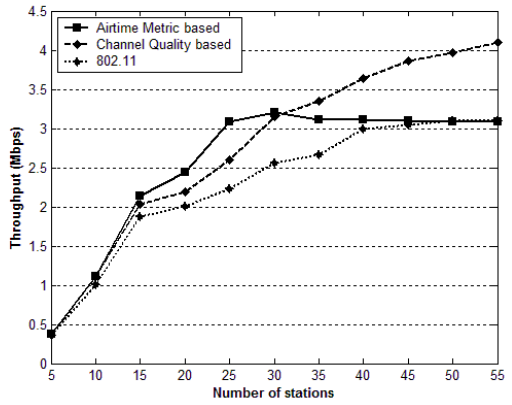
there are 55 associated STAs. Our mechanisms "expand" the network capabilities by applying a sophisticated policy for the association and re-association. The uplink and downlink channel information helps in improving the static association procedure that is used by 802.11, and dynamic STA association and re-association can control in an effective way the communication interference that is present in the network.

Figure 3.5(b) depicts the average packet transmission delay in the network. We can see that the average transmission delay of 802.11 and the network-directed approach is quite low when the network is lightly loaded. When the load in the network increases the average transmission delay gets higher as well. Meanwhile, our proposed airtime-metric based mechanism performs a kind of load

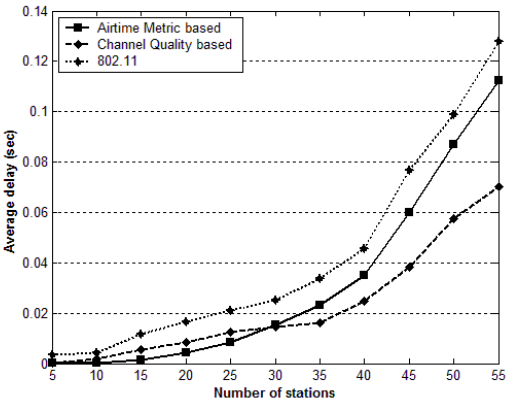
balancing in the network by associating the STAs in an intelligent manner. Dynamic re-associations play an important role in this experiment. The performance of our channel quality based mechanism is quite similar. The RSSI-based association policy used by traditional 802.11 cannot offload some of the STAs, and thus, cannot balance the load in the network, since STAs keep communicating with the AP that has the highest RSSI. Thus, the APs with the strongest signal can become overloaded, and cannot serve the associated STAs in an efficient way. Meanwhile, the network-directed roaming provides a load balancing in the network but it does not take into account the channel conditions. On the other hand, our system monitors the network load and the uplink/downlink channel conditions. In our experiments, we verified that the dynamic re-association introduces a small delay (approximately 50msec) in the communications. However, when the communication delays are observed, we demonstrate (in Figure 3.5(b)) that these small delays do not adversely affect the performance of the network, and on the contrary, can reduce the overall transmission delay. Besides, the study in [30] proves that our mechanism converges in finite time to a stable point and then an equilibrium is achieved. After this point there are no re-associations in our system if there is no change in the network conditions. More details about the convergence of our system can be obtained in [30].

In Figure 3.5(c), we observe the amount of data dropped during the communications due to channel errors and contention. Our proposed mechanisms achieve a lower data dropping rate compared to the traditional 802.11 and the network-directed roaming. The sophisticated association policy that is introduced in our airtime-metric based mechanism captures the uplink/downlink characteristics and the load conditions of each AP. Consequently, we can improve the data dropping rate in the network when the association decision of the STAs is dynamically controlled.

Next, we study the performance of our mechanisms while neighboring APs in the network operate on the same channel. Figures 3.6(a) and 3.6(b) depict the average throughput and the average transmission delay achieved by our mechanisms. As depicted in the figures, under low load, the performance of our mechanisms is the same with the previous experiments. Under high load, the performance of the load based mechanisms decreases. We have discussed the reason why this is happening in earlier section and the simulation results confirm the output of our analytical

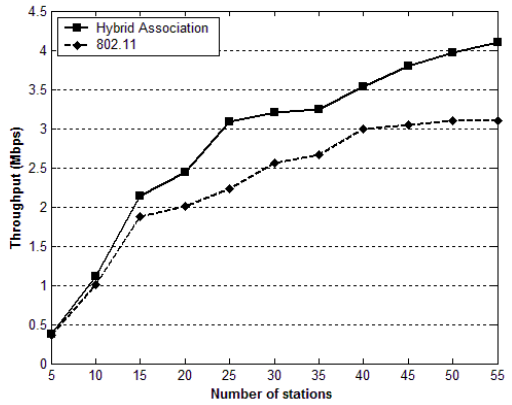


(a) Average throughput.

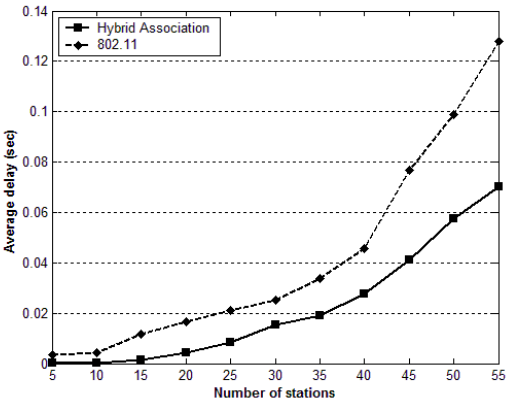


(b) Average delay.

Figure 3.6: Simulation results when APs operate on the same channel.



(a) Average throughput.



(b) Average delay.

Figure 3.7: Simulation results of the "hybrid" association scheme.

study. Actually, under very high load the performance of the airtime-metric based mechanism approaches 802.11 performance.

In Figures 3.7(a) and 3.7(b) we investigate the performance of our hybrid association mechanism while the APs in the network operate on the same channel. Since the association mode changes dynamically depending on the network conditions, the hybrid scheme uses the association mechanism that performs best in each case and thus achieves the best performance between the two mechanisms. In particular, we observe in our simulations that hybrid scheme applies the airtime-metric based association mechanism in low load conditions, and the channel quality based

association mechanism in high load conditions. Moreover, simulation results indicate that the association mode changing process does not adversely affect the overall network performance. Our hybrid association scheme estimates the available bandwidth in the background without hampering the regular network operation. The STAs passively observe the channel occupancy and use the predefined threshold in order to adapt their behavior. Thus, there is no additional communication overhead introduced by the hybrid scheme.

The previous measurements validate our claim that our association mechanisms have better performance than the 802.11 based association procedure and the network-directed approach proposed in [16]. We argue that throughput, transmission delay and data dropping rate are the representative measurements that reflect the system performance under different operational conditions. In the following sub-section we describe the simulation model and results of the end-to-end load based QoS-aware association mechanism.

3.5.2 End-to-End QoS-Aware Cross-Layer Association Mechanism Performance

We keep the protocol modifications (in beacon-probe frames, etc.) that are present in the implementation of the previous mechanisms. In our QoS-aware cross-layer association mechanism, we need to obtain information from the routing layer. Thus, we have modified the AODV routing protocol, which is currently implemented in OPNET. We have introduced the airtime cost as a routing decision metric and we have implemented in this way a link quality aware AODV. In addition, we have implemented a cross-layer interface, which supports the passing of the information from one layer to another.

In our effort to simulate a 802.11-based wireless mesh network in the OPNET simulation environment, we made good use of the wireless routers and their functionality. Note that the wireless routers are part of the backhaul network. We modify the node model of the wireless routers in order to apply the link quality aware AODV routing protocol. The peripheral wireless routers can

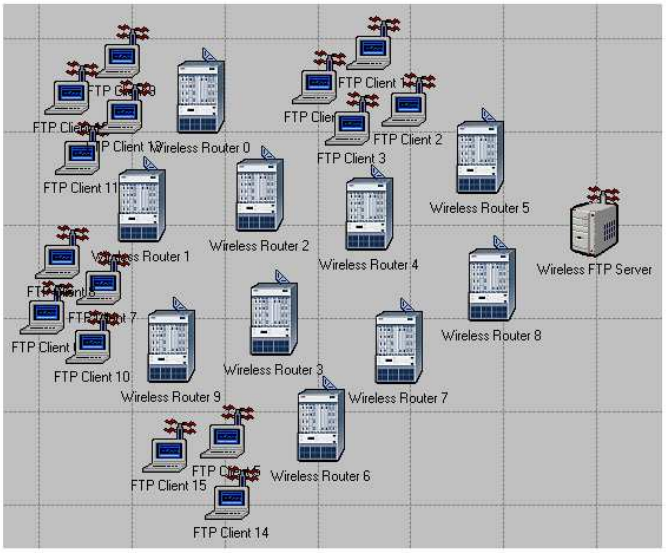
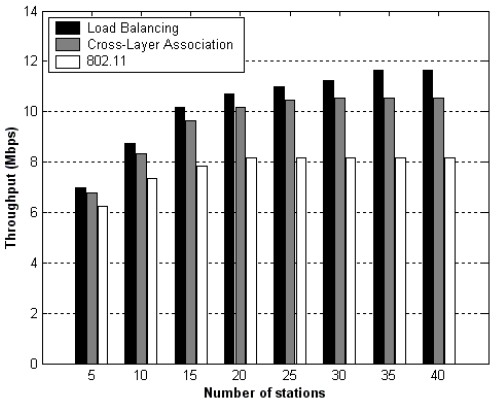


Figure 3.8: The topology of the simulated mesh network.

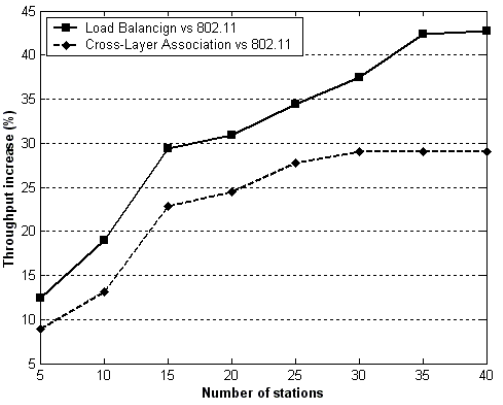
serve as APs as well. In other words, the deployed network that we simulated is a 802.11-based mesh network equipped with a sophisticated cross-layer association mechanism, where the STAs are associated with the available peripheral APs and their data is routed via the link quality aware AODV routing protocol in the backhaul. The STAs are again distributed uniformly at random in the network. For the communication between the wireless routers in the backhaul network, we use the physical model of IEEE 802.11a OFDM physical layer, which supports physical rate of 12 Mbps. The STAs that are associated with the available peripheral APs and generate data at a rate of 1024 kbps. The topology of the simulated wireless mesh network is depicted in Figure 3.8.

We first consider FTP traffic in our wireless mesh network, where there is an FTP server connected to the wireless backhaul and the end users can download/upload different files from/to the server. In the first experiment, we keep the file size fixed (500 Kb) and we continually increase the number of the FTP clients. In the second experiment, we vary the size of the uploaded/downloaded files while keeping the number of the associated FTP clients fixed, i.e., 15 FTP clients.

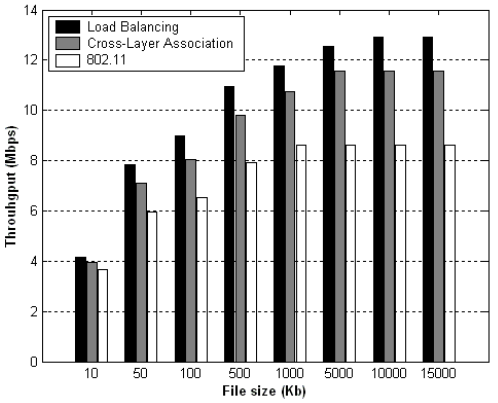
Figure 3.9(a) depicts the variation of the network throughput while we increase the number of the FTP clients from 5 to 40. The results of our system are compared with the corresponding results of the RSSI-based 802.11 association scheme. It is clear that our system attains higher throughput.



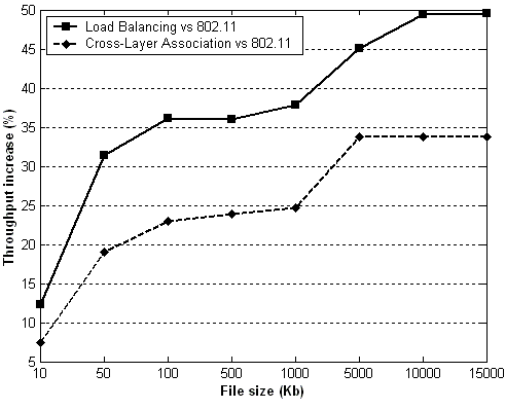
(a) Average throughput vs number of stations.



(b) Average throughput increase.



(c) Average throughput vs file size.



(d) Average throughput increase.

Figure 3.9: FTP simulation results with end-to-end QoS-aware cross-layer association scheme.

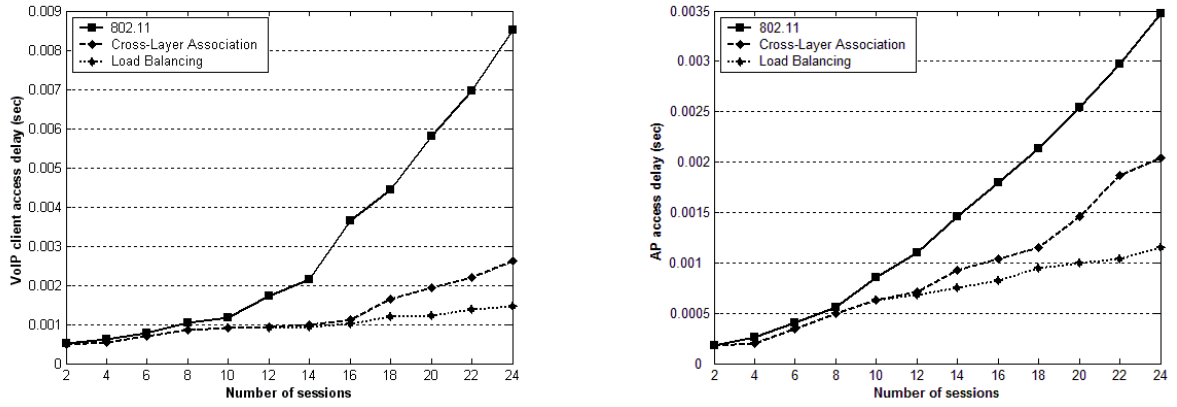
An important remark is that the maximum throughput with the 802.11 is achieved when there approximately 28 FTP clients. Meanwhile, in our proposed system the throughput keeps increasing even beyond 28 clients. The load balancing mechanism keeps increasing the network throughput and achieves its maximum when there are 40 FTP clients. Our sophisticated association mechanism and its combination with the link quality aware routing process in the backhaul can "expand" the network capabilities that remain unexploited by 802.11. The pure 802.11 association policy is not aware of the backhaul characteristics and performs static associations. Practically, this means that in RSSI-based association mechanism there is no concern about the communication conditions that are present as the packet leaves the AP interface and is routed through the backhaul network. A user that is aware of the uplink/downlink channel conditions and the quality of the path in the backhaul

network, can optimally make an association decision. In addition the load balancing mechanism uses the same information to provide a "communication breathe" to the overloaded cells by measuring the balancing index and adapting the weights that are used in our cross-layer association mechanism. Figure 3.9(b) depicts the throughput improvement that our system introduces compared to 802.11. The maximum throughput increase that is achieved by the cross-layer association mechanism is 28% and maximum throughput increase that is achieved by the load balancing mechanism is approximately 43%.

In Figure 3.9(c), we observe the results of the second experiment when the number of clients are kept constant and the file size is increased from 10 Kb to 15 Mb. The throughput increase is illustrated in Figure 3.9(d). Under high network load, the maximum throughput increase is 33% with the cross-layer association scheme, and is 49% when we apply the load balancing mechanism.

Our cross-layer association mechanism is QoS-aware, and it would be interesting to study its performance when we support QoS sensitive applications in our wireless mesh network. We choose to simulate VoIP application, which is a time sensitive application. In our experiments there are several VoIP clients that are associated with the available peripheral mesh APs. The simulation results are performed by varying the number of the VoIP sessions that are supported in parallel. We measure the client access delay, AP access delay, end-to-end delays and the data dropping rate in our simulations. We compare these results with the corresponding results of the RSSI-based 802.11 association scheme. The client access delay of a VoIP packet is the time from when the packet is generated until it leaves the client interface. The AP access delay of a VoIP packet is the time between its arrival to the AP until it is either successfully transmitted over the wireless mesh network or dropped at the head of the queue because it has exhausted the retry limit for retransmissions. From an end-to-end viewpoint, it is essential for the local delay to be small so that the overall end-to-end delay of a VoIP stream can be bounded tightly to achieve good quality of service.

In Figure 3.10(a) we observe the average local client access delay in the network while the number of the sessions, that are supported in parallel, is increased. It is obvious that the VoIP clients



(a) Average client access delay.

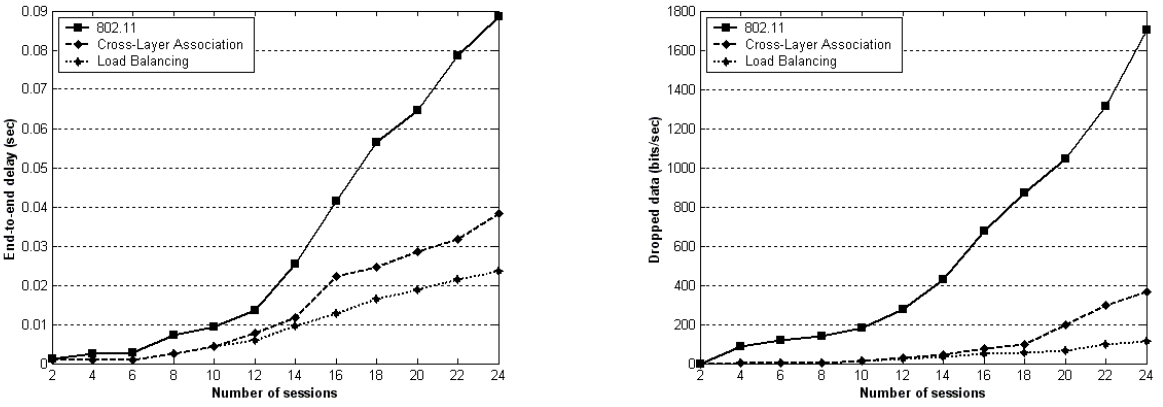
(b) Average AP access delay.

Figure 3.10: Average VoIP delays.

that adopt our association policy can transmit their VoIP packets to their associated APs quite faster than those that follow the traditional 802.11 association policy. This is because in our system the load in the network is balanced. The clients are associated with the available APs in a way that minimizes the channel access delay. We avoid in this way overloaded APs that cannot support fast channel access to their clients. The main difference in the performance of these two approaches is more observable when we support more than 8 sessions in parallel. Our load balancing mechanism minimizes the channel access delay while it provides a "cell breathing" to the overloaded cells and keeps the client access delay in low level while the traditional 802.11 operation overloads the network and the client access delay is continually increased.

Figure 3.10(b) depicts the average local AP access delay. The simulation results and the outcome of the comparison with 802.11 is similar to the client access delay measurements. In high load conditions ($\geq$10 sessions) we can observe a significant difference in the AP access delays between the two mechanisms. The main reason that the 802.11 AP access delay is getting very high is that the overloaded APs have a lot of traffic to forward to the backhaul network. Thus, the VoIP packets are waiting for a long time to be served in the AP queue.

In Figure 3.11(a), we can see the average end-to-end delay in the transmission of the VoIP packets from the sender to the receiver. The main factors that affect the end-to-end delay is the



(a) Average end-to-end delay. (b) Average dropped data.

Figure 3.11: Average end-to-end delay and dropped data in VoIP.

client access delay, the AP access delay and the backhaul routing delay. We have described the characteristics of the first two kind of delays. The routing delay is introduced by the routing protocol that is executed at the backhaul network. The modified AODV protocol is aware of the link quality conditions between the backhaul nodes and performs effectively the routing process by introducing low routing end-to-end delays. The average end-to-end delay of our system is lower that the average 802.11 end-to-end delay especially in high load conditions. We argue that the most interesting result is depicted in Figure 3.11(a): The pure 802.11 operation can support at most 14 sessions in parallel while the proposed load balancing mechanism has the capability to support 24 sessions in parallel. Therefore we gain approximately 70 % network performance improvement. The network capabilities are expanded by the use of the sophisticated load balancing mechanism.

Lastly, we measure the average dropped data in the VoIP communication due to channel errors and contention. The comparison of our system and the 802.11 performance is depicted in Figure 3.11(b). It is observed that our proposed protocol keeps the packet dropping rate at a very low level, while the traditional 802.11 has high level of packet drop.

3.6 Testbed Evaluation

In this section we present the experimental evaluation of the proposed cross-layer user association mechanism. We evaluate our mechanisms in a wireless tesbet deployed at the University of Thessaly. In the forthcoming subsections we describe the testbed deployment, we give some details about the implementation of the proposed mechanisms and we present the evaluation results of the association scheme.

3.6.1 UTH Wireless Testbed

Architecture

The UTH Wireless Testbed has been designed to operate as a static wireless mesh network; mobility can also be facilitated by connecting laptop computers and other mobile devices and using the wireless interface(s) of those. The testbed is deployed across the 5 floors and the rooftop of the Computer and Communications campus building, in downtown Volos, Greece.

We are using ORBIT nodes in our testbed. Each node has 2 wireline (Ethernet) interfaces and is connected to a central server, through a wired Ethernet back-haul infrastructure. One of the Ethernet interfaces is used for the network boot as well as for accessing the OS of the node and logging experiments. The second interface is connected to a Chassis manager, through which we are able to remotely power on/off the node. In conjunction to our NFS mounting strategy, this has many advantages; the main advantage is that all configurations and testbed updates are conducted centrally on the server. A simple, remote node reboot command is sufficient for all nodes to load the most-updated kernels, modules, drivers and applications. Hence, although the ORBIT nodes come with a local hard disk, we are not currently using it for OS boot. On the contrary, we have configured the nodes to retrieve their kernels, and mount their root file system directly from the central server (using the wired back-haul). This means that every researcher can maintain his/her own, independent experimental setup, including kernel, drivers, implementations, measurement logs, and every other potential component of the OS distribution. Note here that it

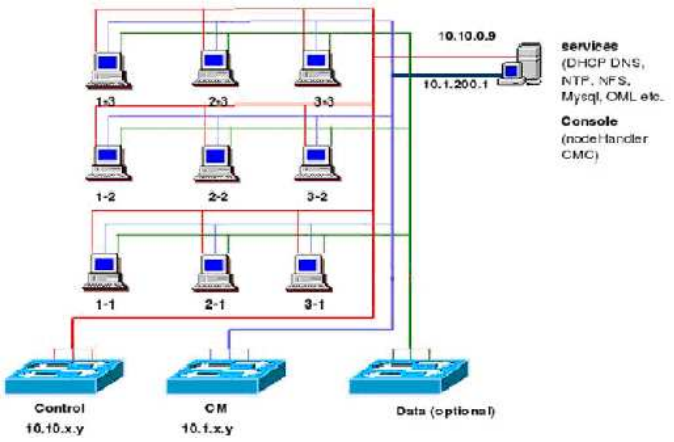


Figure 3.12: Testbed architecture.

isn't even necessary to run Linux; any operating system capable of mounting their root off NFS will work.

Hardware, Configuration and Management Details

The testbed consists of wireless nodes that are based on commercial WiFi wireless cards as per the the IEEE 802.11 standard. The architecture of the testbed is similar to the ORBIT testbed in Rutgers University. The wireless nodes are connected via a wired gigabit Ethernet and the network is managed by three servers (figure 3.12). The three servers are described below:

Services - It is used to host various services including DHCP, DNS, NTP, TFTP, PXE, Frisbee, NFS, mysql, OML and Apache. We have different aliases for the management host to segregate the services that it hosts. This machine or port shall be connected with the Control port of the nodes.

Console - It is used to run experiments using the management software. Console is also connected with the Control port of the nodes. It may share one Ethernet port with Services. A better way is setting up a console in one machine exclusively and let it be accessible by experimenters using ssh or XDMCP.

CMC - It is the control and monitoring manager for all CM elements of the nodes. It is connected with the CM port of the nodes and can NOT share the same Ethernet port with Services and Console.

Figure 3.13: Wireless Node Architecture.

Wireless Nodes

Each wireless node in the testbed consists of an 1 GHz VIA C3 processor, 512 MB of RAM, 40 GB of local disk, three Ethernet ports, two 802.11 a/b/g wireless cards and a Chassis Manager (CM) to control the node. The basic architecture of a node is depicted in Fig 3.13. The three Ethernet ports in a node are used as follows:

Control port - The Ethernet port between the USB ports of the node. It is a Rtl-8169 Gigabit Ethernet port, which is used to load and control the wireless node and to collect measurements.

Data port - The Ethernet port above the USB ports. It is a VT6102 Rhine-II 100/10baseT Ethernet port, which is used for wired data communication between the nodes.

CM port - The 10BaseT Ethernet port on the Chassis Manager (CM) Card, which is used to control the on/off switching of the nodes. It communicates with the management application, which controls the nodes in experimentation that is called Gridservice.

The wireless testbed is designed in a way that it can setup an infrastructure network. The backbone of the network can be either wired or wireless. For the implementation of the wired backbone among the wireless stations, we can use the data plane of the testbed (data port on each node). For the implementation of the wireless backbone, we can use in parallel both the wireless cards of the node. One card will be setup in infrastructure and one in the ad-hoc mode. In this way

we can create a hybrid network that consists of a wireless backbone (cards in ad-hoc mode) that forwards the packets to the final destinations in a centralized manner (cards in infrastructure mode). In other words, we can setup an infrastructure network where the distribution system among the access points is a wireless backbone.

3.6.2 Evaluating the Cross-Layer User Association Mechanism in UTH Testbed

We have used MadWifi driver in order to implement our scheme. MadWifi is one of the most advanced WLAN drivers available for Linux today. It is stable and has an established user-base. The driver itself is open source but depends on the proprietary Hardware Abstraction Layer (HAL) that is available in binary form only. The current stable release is v0.9.4. MadWifi has a well commented code and a large community of users and developers use it and hence it is thoroughly investigated. MadWifi is looked as the best open source driver for wireless cards for Linux as of now. It is constantly updated patched and researched by the MadWifi community.

The MadWifi driver implements most of the 802.11 MAC functionalities and therefore it is easy to modify its code in order to change parameters, or implement new features. In particular, in our protocol implementation we have changed the RSSI-based association functionality that is implemented in MadWifi and we have introduced our airtime-based association mechanism. Besides, we have extended the management frames (beacon frames) in order to carry useful information about the operational parameters of the APs in the network. The main modifications that we have introduced in the driver are:

1. Every AP must measure and broadcast periodically the cumulative airtime cost in both directions (uplink and downlink): **a.** Each AP measures the transmission rate and the packet error rate (based on the transmission of the data frames) at each downlink communication and then computes the cumulative airtime cost for its downlink. As far as the computation of the packet error rate is concerned, we capture the percentage of the dropped data frames in

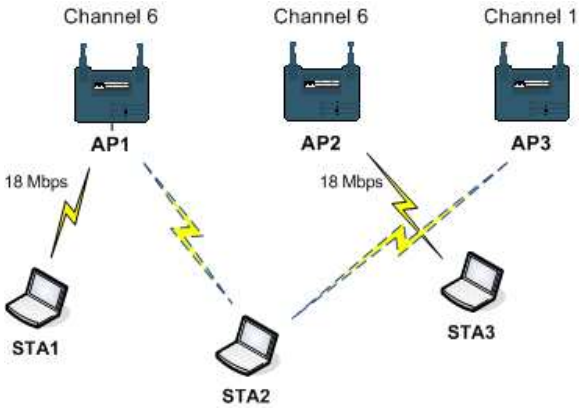


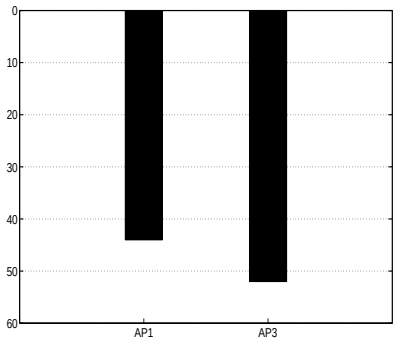
Figure 3.14: Topology of the first experiment.

a time window. **b.** Each associated STA captures the percentage of the dropped data frames in order to compute the packet error rate in its uplink. Then, the STAs compute their uplink airtime cost and piggy-back its value in their data frames. This is a practical way to inform the AP about the quality of the uplink communications. **c.** Each AP computes the cumulative airtime cost in both directions based on the previous measurements. **d.** Each AP periodically broadcasts the cumulative airtime cost (in its beacon frames). In order to incorporate the aforementioned value in the beacon frame we have overwritten some of its fields

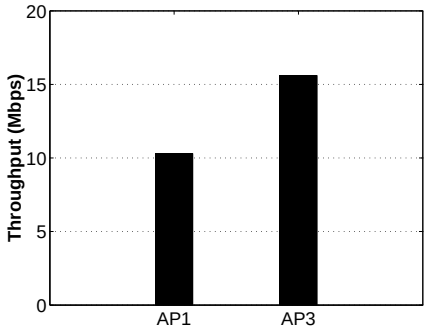
2. Each STA that tries to find an AP to be associated with, initiates a scanning procedure. During this procedure it receives the transmitted beacon frames and captures the cumulative airtime cost of the candidate APs for association. Then, the STA decides to be associated with the AP with the minimum airtime cost.

Simple Scenarios

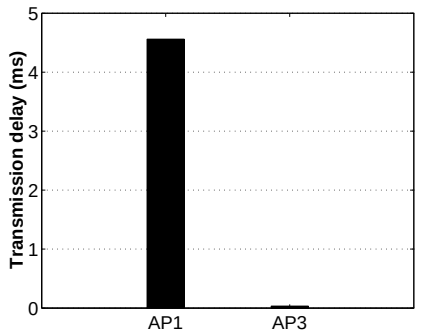
In our scenario-based experimental methodology we first support different simple network topologies where we apply the implemented association scheme. In these topologies we vary the operational parameters (communication load, locations, etc.) in order to highlight the contribution of the proposed association mechanism. Then, we compare the cross-layer association mechanism to the pure 802.11 operation (RSSI-based association).



(a) RSSI.



(b) Average throughput.



(c) Average transmission delay.

Figure 3.15: Performance results for the first experiment.

In the first experiment (3.14) we deploy three APs. Two of them face co-channel interference since they operate on the same frequency. The third AP operates on orthogonal frequency. AP1 and AP2 have one associated client and they send downlink traffic with rate 18Mbps. We must mention here that we exclude AP2 from the association process. We have placed AP2 in the network in order to vary the operational conditions in the network (co-channel interference). STA2 is closer to AP1 and therefore the received RSSI from this AP is higher than the received RSSI from AP3. STA2 chooses for association AP1 (highest RSSI) in case that it operates in pure 802.11 mode. Contrarily, our airtime-based user association mechanism captures the co-channel interference effects (increased transmission error rate, etc) and chooses for association AP3. Figure 3.15 depicts the performance of STA2 under the aforementioned two schemes. As we can see STA2 achieves higher throughput values under the airtime scheme (associated with AP3) and lower transmission delays (52% improvement).

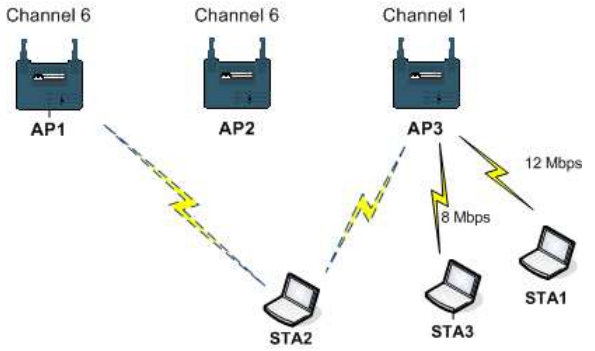
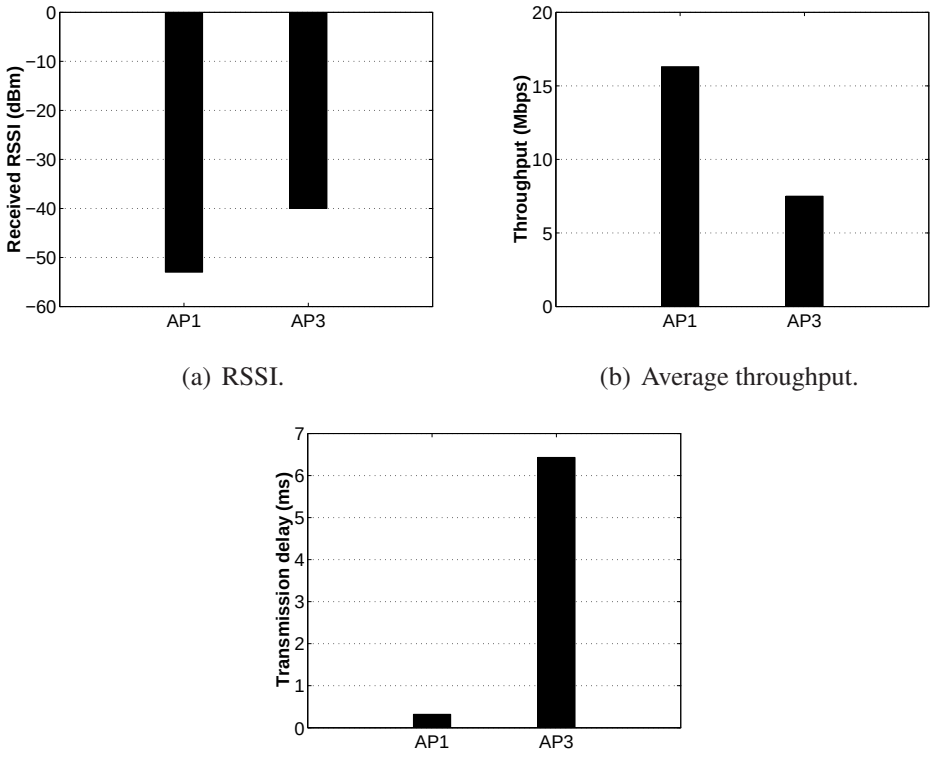


Figure 3.16: Topology of the second experiment.



(a) RSSI.

(b) Average throughput.

(c) Average transmission delay.

Figure 3.17: Performance results for the second experiment.

In the second experiment (3.16), STA2 is placed near to AP3 and therefore the received RSSI from this AP is higher than the received RSSI from AP1. Besides, AP3 is overloaded since it must serve STA1 and STA3. Pure 802.11 operation forces the STA2 to be associated with AP3. The cross-layer association mechanism makes a sophisticated decision taking into account the overloaded situation in AP3 and forces STA2 to be associated with AP1. In figure 3.17 we can observer

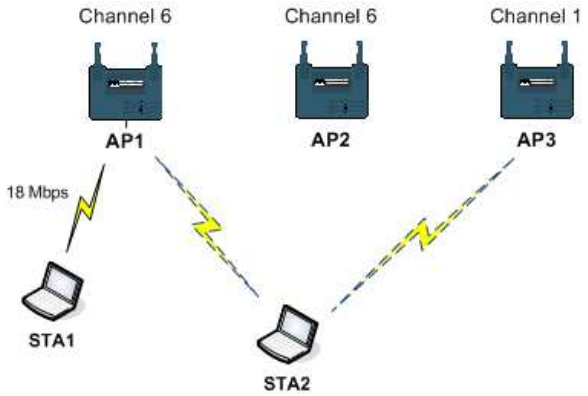
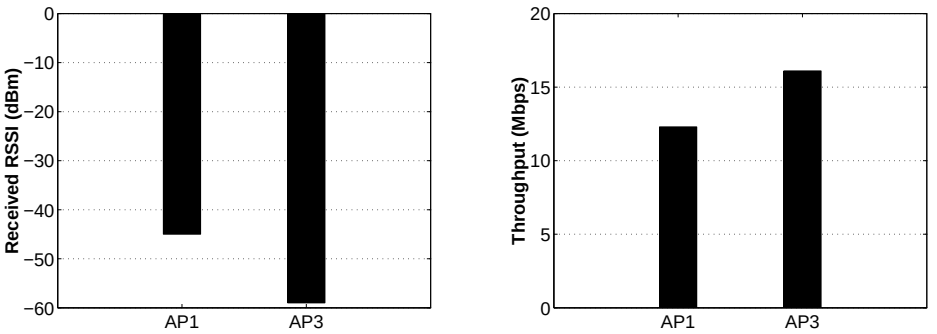
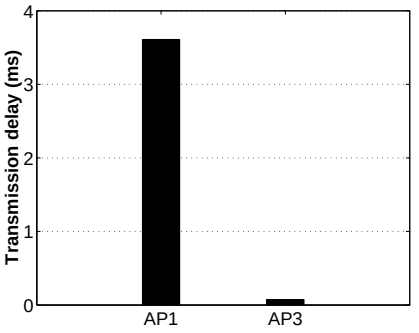


Figure 3.18: Topology of the third experiment.



(a) RSSI.

(b) Average throughput.



(c) Average transmission delay.

Figure 3.19: Performance results for the third experiment.

the performance of the network under the two schemes. Our association mechanism achieves the best performance (118% improvement).

In the third experiment (3.18) we keep the same network topology and we now place two stations in the system (they send uplink traffic). STA1 is associated with AP1 and the other two APs

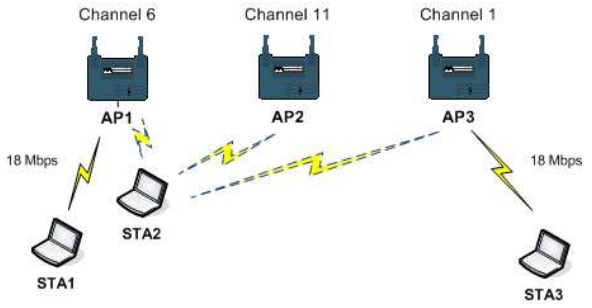
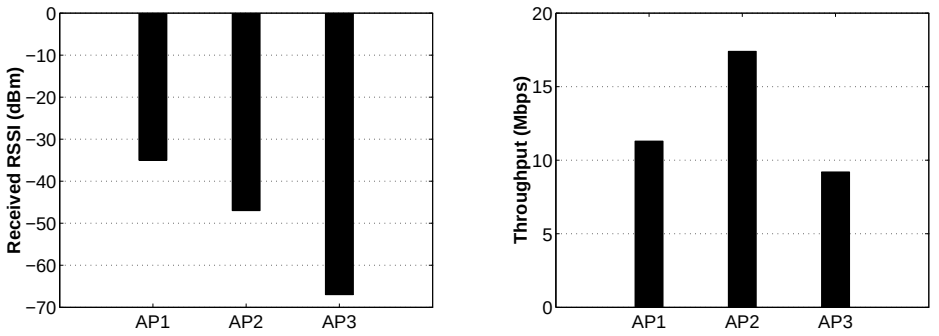
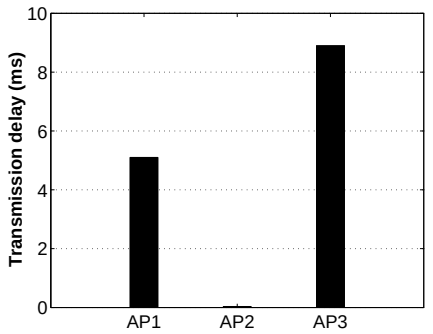


Figure 3.20: Topology of the fourth experiment.



(a) RSSI.

(b) Average throughput.



(c) Average transmission delay.

Figure 3.21: Performance results for the fourth experiment.

have no associated clients. STA2 receives strong signal (RSSI) from AP1 and therefore it associates with AP1 (based on 802.11). This static association policy does not provide a balanced network operation since there are three APs available in the network and STA2 is forced to be associated with the most loaded AP (AP1). A load-aware association policy (like our cross layer association policy) would consider that in an overload situation like this the transmission rate would drop and the packet error rate would increase. Consequently, our load-aware association mechanism selects AP3

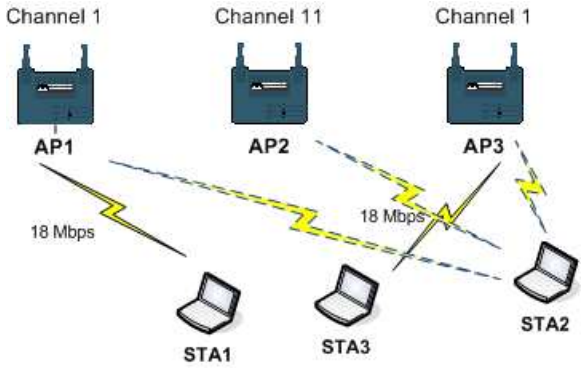
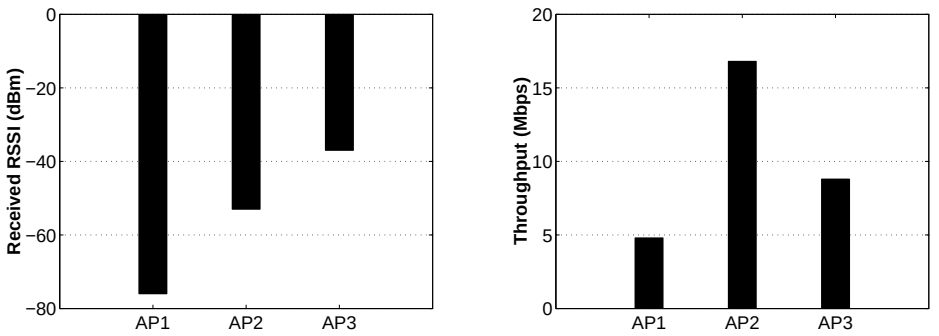
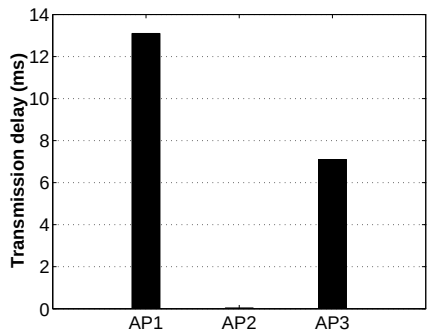


Figure 3.22: Topology of the fifth experiment.



(a) RSSI.

(b) Average throughput.



(c) Average transmission delay.

Figure 3.23: Performance results for the fifth experiment.

for association. The network performance is significantly improved under our scheme as depicted in figure 3.19 (31% improvement).

In the forth scenario (3.20) we vary the experimental parameters as far as the availability of the APs that can be associated with clients is concerned. In other words, STA2 is free to choose between AP1, AP2 and AP3 that operate on orthogonal frequencies (there is no co-channel interference

in the system). It is obvious that the RSSI-based association policy prefers AP1 for association. Our airtime-based user association mechanism captures that AP1 and AP3 have already associated clients that must be ”served” by them and STA2 must share the available bandwidth (in case that it will be associated with one of them). The best association decision under our scheme is AP2. Figure 3.21 depicts that our association policy can provide a balanced network operation while the 802.11 does not exploit effectively the network resources (54% improvement).

In the last experiment (3.22) we study a more complicated network scenario where AP1 and AP3 operate on the same frequencies. We could claim that there is not co-channel interference since AP1 is out of the transmission range of AP2 and therefore there is no performance degradation. Taking a closer look in the experiment, we can observe that STA1 is located in the transmission range of STA3 and therefore the two cells do face co-channel interference. For example, when STA1 transmits to AP1 forces STA3 to be silent until this transmission will be over (based on CSMA/CA). Our mechanism is able to ”predict” this suboptimal network operation and therefore forces STA2 to associate with AP2 (91% improvement in figure 3.23).

Complex Scenarios

In order to study the behavior and the scalability of the proposed association mechanism in a more realistic environment we have applied our system in the UTH wireless testbed. The topology of the testbed is depicted in figure 3.24. The wireless network is deployed in the 4rth, 5th and rooftop floor of the building.The testbed setup that we use in our experiments include 5 APs and 14 clients. As far as the environmental conditions are concerned, the temperature is average and the humidity is high (these conditions affect the channel quality). The walls are supported by thick metallic skeletons, and many of them are made of brick. This degrades the signal strength on a sub-set of the links where no direct line of sight exists. The positions of the APs in the network were selected after a set of measurements and placed uniformly to ensure maximal coverage.

Most of our experiments were performed with IEEE 802.11a (in order to avoid interference from the neighboring networks). We must mention here that the wireless nodes are equipped with

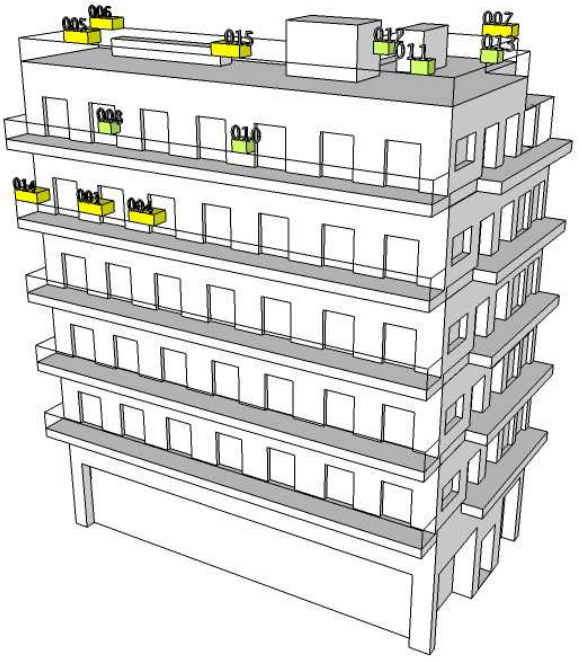
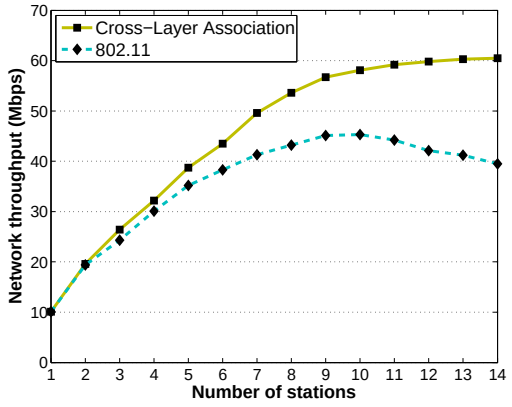


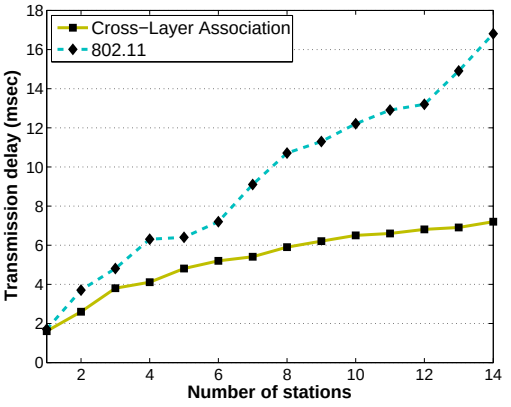
Figure 3.24: Topology of the UTH wireless testbed.

two wireless interfaces that can be used independently in our experiments. All nodes by default set their transmission power to the maximum (20 dBm). Each client sends/receives fully saturated UDP traffic for two hours, to/from its AP(we run each experiment several times in order to get accurate results). We use the iperf bandwidth measurement tool to generate traffic in the network and measure the performance. During each experiment, a central testbed server periodically stores information concerned to the performance of the network.

In the first experiment we set nodes 1, 6, 7, 10, 11 as the APs in the network and the rest of the nodes act as clients. The channels that are used by the APs are selected in a way that there is no co-channel interference effects. First of all, we turn on the APs in the network and we keep the clients turned off. Then, we start turning on the clients and measure the network performance while the number of the clients in the network increases. We apply both the airtime-based association policy (modified MadWifi driver) and the pure 802.11 protocol (original MadWifi driver). We compare the network performance that is achieved under these policies. An important observation in this experiment is that the APs 1 and 11 “attract” a lot of clients (5 clients each) in case that the



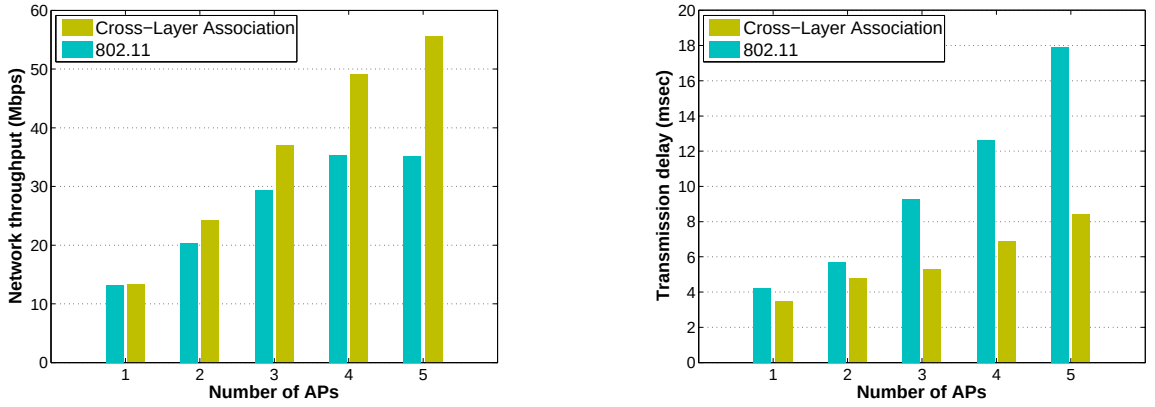
(a) Total network throughput Vs. Number of clients.



(b) Average transmission delay Vs. Number of clients.

Figure 3.25: Network performance in the first experiment.

RSSI-based association policy is applied. This is true due to their favorable location. An important consequence of the aforementioned situation is the overloaded performance of these two APs. APs 6, 7, 11 serve the rest of the clients in the network. It is obvious that under these operational characteristics the bandwidth is significantly wasted in the network. Figure 3.25(a) depicts the performance variation of the network while the number of the clients increases. Our cross-layer association policy achieves similar performance to 802.11 when the load in the network is low. Contrarily, when the number of the clients is getting high and the load in the network increases the suboptimal operation of 802.11 drops the network performance. The airtime mechanism captures the overload effect in APs 1 and 11 (based on the transmission rate and the packet dropping), and provides load balancing by forcing some clients to be associated with the neighboring APs. The performance of the network is improved up to 52%. Another important observation is that the total network throughput achieved by 802.11 is maximized when 10 clients are supported in the network and after that point the performance drops due to the overloading. However, our association policy keeps improving the total network throughput even if 14 clients are supported in the network. Figure 3.25(b) depicts the average transmission delay when both policies are applied. Our association mechanism keeps the average transmission delay in low levels and improves the performance of 802.11 by 59%.

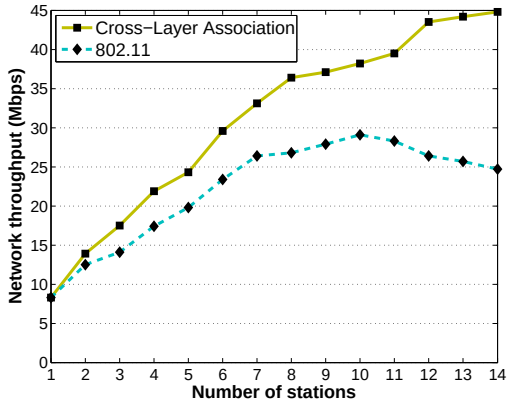


(a) Total network throughput Vs. Number of APs. (b) Average transmission delay Vs. Number of APs.

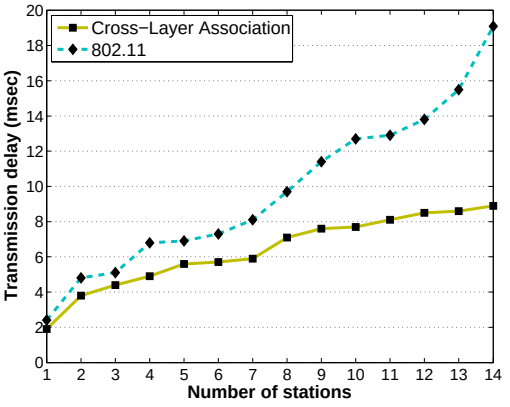
Figure 3.26: Network performance in the second experiment.

In the second experiment we keep the same topology and the network configuration. In order to measure the scalability of the proposed association mechanism, we pick randomly one AP and 3 clients at a time and we turn them on. This process continues till all the APs and the clients are turned on in the network. Figure 3.26(a) shows the total network throughput and figure 3.26(b) shows the average transmission delay while the number of the APs increases. As we can see our mechanism scales much better and improves the total network throughput achieved by 802.11, by 61%. An important observation here is that while the number of the deployed APs in the network increases, the clients act statically. In other words, the clients keep their associations with the old APs and a possible re-association is significantly delayed. Our mechanism introduces dynamic re-associations when new light loaded APs with better channel quality are deployed in the network, providing in this way balanced network operation.

In the third experiment we opt to introduce co-channel interference and high contention in the network. The APs 6, 10 and 11 operate on the same frequency and the rest APs operate on randomly picked frequencies. This scenario is close to real network deployments where most of the APs that are deployed operate on default frequencies or the users chose randomly channels for their APs without taking into account the interference. As we have seen from the previous experiments, the AP 11 is overloaded when the 802.11-based association procedure is applied. In



(a) Total network throughput Vs. Number of clients.



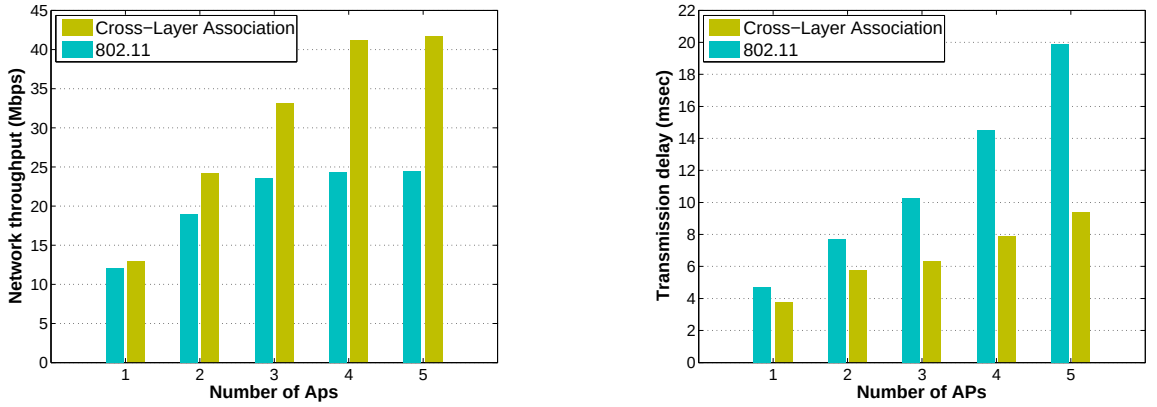
(b) Average transmission delay Vs. Number of clients.

Figure 3.27: Network performance in the third experiment.

the current experiment the suboptimal network performance of 802.11 is getting even worse since the AP 11 must respect the transmissions of APs 6 and 10 (must be silent when these APs are active) since they operate on the same channel. Figures 3.27(a) and 3.27(b) show the comparison between the cross-layer association mechanism and the 802.11. Our association policy captures the performance degradation that is introduced due to the co-channel interference and the increased contention levels, and forces the clients to be associated with the rest APs. Unfortunately, the RSSI-based association policy is not capable to capture these conditions and keeps associating the clients with the closest AP (the AP with the higher RSSI). The performance of 802.11 is improved by 84%.

In the fourth experiment we keep the previous configuration and we measure the performance of the network while the number of the deployed APs varies. In figures 3.28(a) and 3.28(b) we can observe that the performance improvement that is introduced by our mechanism is higher compare to the previous experiment (close to 75%). In addition, the total network throughput is stabilized when we have more than 3 APs deployed in the network. Our mechanism applies a sophisticated association policy expanding the network capabilities and maximizing the total network throughput in presence of 5 APs in the network. In particular, the dynamic re-associations that are present under the cross-layer association mechanism provide a "cell breathing" to the overloaded cells.

Continuing our experimental evaluation, we vary "artificially" the interference level in the net-



(a) Total network throughput Vs. Number of APs. (b) Average transmission delay Vs. Number of APs.

Figure 3.28: Network performance in the fourth experiment.

work. In this experiment we opt to observe the behavior of the proposed association policy when some of the APs in the network face huge amounts of interference/contention and therefore, they are enable to serve their clients. We introduce malicious clients, called jammers, that produce huge amounts of traffic in a specific channel trying to achieve denial of service in the specific part of the network. Our implementation of a constant jammer is based on a card configuration that sends broadcast packets as fast as possible. By setting the CCA threshold to 0 dBm, we force the WiFi card to ignore all 802.11 signals during carrier sensing (packets arrive at the jammers circuitry with powers much less than 0 dBm, even if the distances between the jammer and the legitimate transceivers are very small). The jammer transmits broadcast UDP traffic. This ensures that its packets are transmitted back-to-back and that the jammer does not wait for any ACK messages (the back-off functionality is disabled in 802.11 for broadcast traffic). In particular, we use the clients 5, 13, 14 as jammers, that are close to the APs 1, 6 and 11. Figure 3.29 depicts the effect of the jammers in the network performance. The throughput degradation with 802.11 is very impressive (close to 73%), especially when all the jammers are active in the network. As we have previously mentioned the jammed APs 1 and 11 serve a lot of clients. These clients face now a denial of service attack and their performance is significantly affected. Our association mechanism captures the huge amounts of interference/contention (measuring the huge packet dropping and the transmission

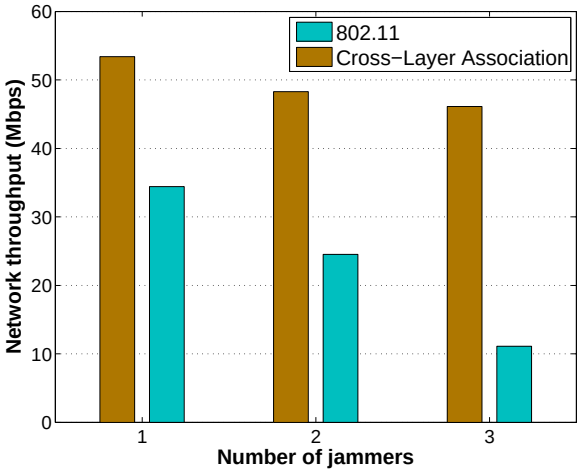


Figure 3.29: Total network throughput Vs. Number of jammers.

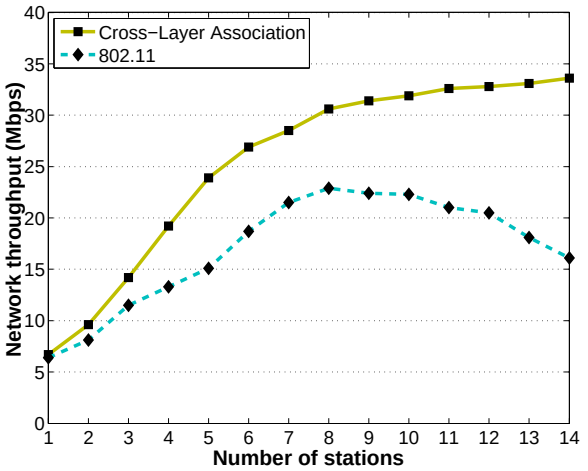


Figure 3.30: Network performance with 802.11g (Total network throughput Vs. Number of clients).

delays) and force the jammed clients to be associated with the "healthy" APs. In this way we limit the throughput degradation (close to 18%).

In the last experiment we use 802.11g. The main focus of this experiment is to approach the operational conditions of the real 802.11 wireless network deployments. We select randomly the channels that are used by the APs in our network, without taking into account in this decision the interference from the neighboring networks. Due to the limited number of orthogonal channels in the 2.4 GHz band, the contention and the interference are high. This effect is getting even worse

while most of the users set their APs in a default channel. We performed a scanning procedure in the neighborhood: 18 APs are active (outside the testbed) and 10 of them use channel 6 (default channel). Figure 3.30 depicts the total network throughput while the number of the clients varies. The cross-layer association mechanism keeps improving the network throughput by re-associating the clients that face huge interference levels with APs that are free of interference and contention, while the 802.11 is incapable to achieve high throughput values. The performance improvement that is introduced by our mechanism is quite impressive (up to 112%).

Chapter 4

Cooperative Handoff in 802.11 Wireless Networks

In this chapter we present a cooperative handoff framework that can be applied in both *WLANs* and *wireless mesh networks*, and speeds up the basic handoff procedure. Section 4.1 presents in detail our 802.11k compliant cooperative handoff framework. In section 4.2, we describe the evaluation results of the proposed mechanisms.

4.1 A cooperative handoff framework

In this section, we present a 802.11k compliant framework for cooperative handoff. The main contribution of this scheme is the provisioning of fast handoff procedures that take full advantage of the cooperation between STAs and APs in the network. The underlying association/handoff decision protocol can utilize the capabilities of this framework and improve its performance. The proposed framework focuses on wireless mesh networks where the APs communicate through a wireless backbone network, but it can be applied in multi-cell wireless networks (WLANs) where the inter-APs communication can be supported through their wired connections.

4.1.1 IEEE 802.11k framework

IEEE 802.11k [45] is a *Radio Resource Management* standard that provides measurement information for APs and STAs in the network. In particular, 802.11k determines *Radio Measurement* mechanisms that enable STAs/APs to observe and gather data about the radio link performance and the radio environment. There are special *Radio Measurement* periods where the STAs/APs execute these procedures in order to get informed about the communication conditions in their neighborhood. During those *Radio Measurement* periods the STAs/APs switch to a control channel in order to communicate and share information. Our cooperative framework exploits the capabilities of the 802.11k-based mechanisms and provides efficient handoff procedures. Below we describe two mechanisms that are utilized in our framework:

- **Beacon Report**: A STA can receive a *beacon report* from the neighboring STAs in order to be aware of the communication conditions in its neighborhood. The STA can operate in an active way and broadcast a *beacon request* to the neighboring STAs. Afterwards the STA waits for a specific period (measurement period) in order to receive *beacons* from the neighboring STAs. In addition, a STA can operate in a passive way by listening to *beacons* that neighboring STAs send during the measurement periods. *Beacon* in its pure form carries information about the operating APs in the neighborhood, their communication channels, BSSID, etc. We must mention that 802.11k specifies measurement periods but it doesn't define the way to adjust their duration and how frequent they are initiated. Figure 4.1 depicts the general format of the *measurement report* defined in *802.11k* standard [45], which contains the *beacon report* (inside the Measurement Report field). *Beacon report* is depicted in figure 4.2. More information about the details of the fields that are present in the *beacon report* can be obtained in [45].

- **Neighbor Report**: In this request/response mechanism a STA/AP can request information about the neighboring APs. *Neighbor report* supports communication and information exchange between APs in the network (this is not supported in *beacon report*). According to

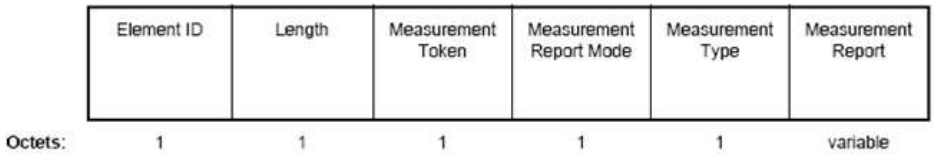


Figure 4.1: Measurement report element.

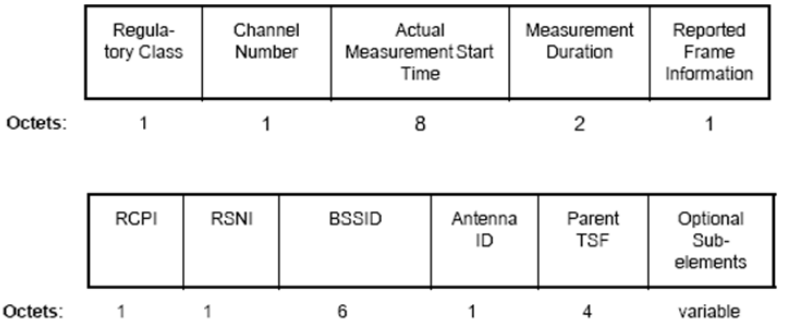


Figure 4.2: Beacon report.

802.11k a STA/AP can initiate a *neighbor report* process and send a *neighbor request* to the neighboring APs. The APs that "hear" this request react by sending a *neighbor report* that contains information stored in their MIB (Management Information Base). In addition, the APs can behave in a passive way during a *neighbor report* process. In other words during the measurement period all the APs in the network broadcast *neighbor reports* that contain information stored in their MIB. Therefore, an AP can "hear" the reports of its neighboring APs without initiating a request/response procedure. Figure 4.3 depicts the *neighbor report* element as defined in *802.11k* [45].

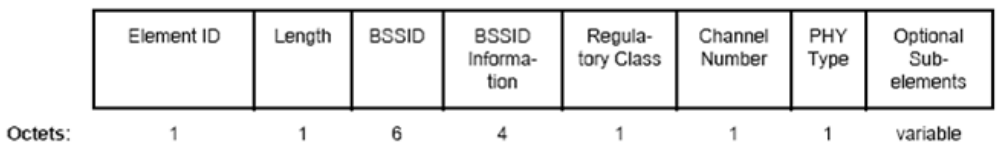


Figure 4.3: Neighbor report element.

4.1.2 Proposed framework

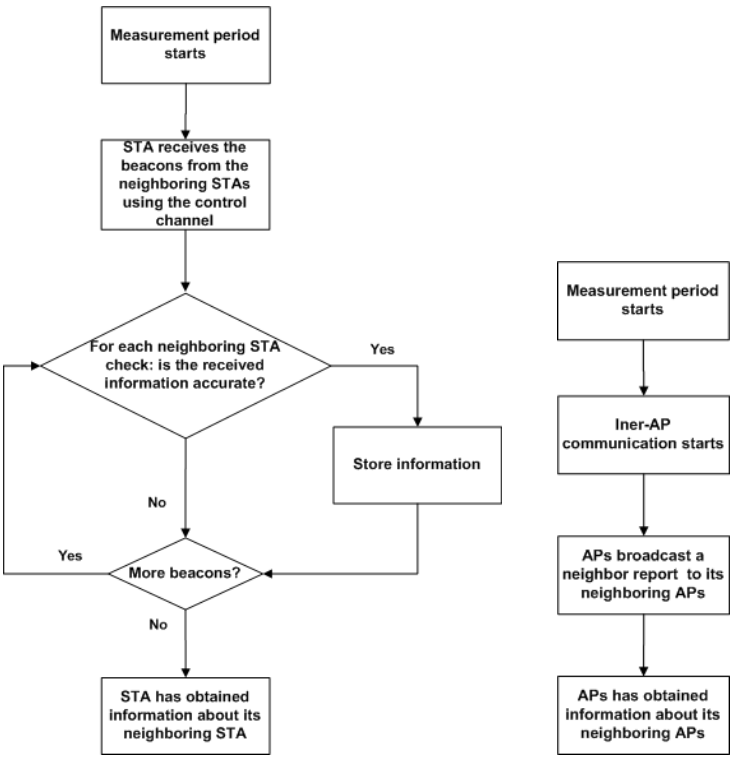
In our framework we support information sharing between the STAs and the APs in the network, based on the aforementioned mechanisms that are defined in 802.11k. The first component in our framework is the ad-hoc cooperative procedure that STAs use in order to share information with their neighboring STAs. The second component is the cooperation between the APs in the network, where inter-AP communication is supported and the APs share information with their neighbors. The previous two procedures are totally independent and they are executed during the periodic measurement periods. Therefore, at the end of each measurement period the STAs and the APs are aware of the operational conditions of their neighboring STAs/APs. In case that a STA is searching for a new AP, it initiates a cooperative handoff procedure where the information that has been obtained during the last measurement periods is used.

The flow diagrams in figure 4.4 depict the main steps of the information sharing procedures. We now give more details about the ad-hoc cooperative information sharing depicted in figure 4.4(a) and the cooperation between the APs depicted in 4.4(b):

Ad-hoc cooperative information sharing

- **STEP 1**: STA switches to the control channel and "hears" the *beacons* that the neighboring STAs send during the measurement period. The STAs choose a random interval and broadcast a *beacon* when this interval expires. *Beacon* collisions are avoided by using this random interval mechanism. The length of the measurement period depends on the number of the STAs that are present in the network. During this measurement period a STA must acquire a uniform distribution of received *beacons* and minimize the collisions. The mechanism that defines the optimal measurement period is out of the scope of his paper.

- **STEP 2**: STA receives the *beacons* that the neighboring STAs send (during one measurement period). We divide the handoff related information that the *beacons* carry into two categories: a) "Objective" information: MAC address of the APs, their operational frequencies, etc. and



(a) Ad-hoc cooperative information sharing.

(b) Cooperation between the APs.

Figure 4.4: Cooperative information sharing during the measurement periods.

b) "Subjective" information: communication load of the APs, channel conditions, error rate, transmission rate, etc. We call this information as "subjective" because each STA in the network experiences its own communication conditions and therefore it can provide a "subjective" view of the network in its proximity. We must mention here that the aforementioned information is stored into the basic fields of the *beacon* frame, depicted in figure 4.2. Additionally, several fields can be appended in the *Optional Subelements* super field. In this way the *beacon* frame can be extended in order to carry extra information about the operational environment.

- **STEP 3**: For each received *beacon*, the STA checks the accuracy of the "subjective" information that is carried.

- **STEP 4**: STA stores only the "accurate information", in the way accuracy is defined in the

following discussion.

Cooperative information sharing between the APs

- **STEP 1**: APs choose a random interval and broadcast a *neighbor report* when this interval expires. *Neighbor report* collisions are avoided by using the random interval mechanism. The measurement period should be adjusted based on the number of the APs that are present in the system, in order to eliminate the collisions.

- **STEP 2**: APs passively "hear" the *neighbor reports* that the neighboring APs send. The *neighbor reports* carry "objective" information in its information fields (figure 4.3).

- **STEP 3**: APs store the received information in order to be able to respond to a possible information request by a STA.

Accuracy of the "subjective" information

We claim that the "subjective" information that is carried in the beacon frames is accurate and therefore can be used by the STA that initiated the cooperative handoff procedure when the neighboring STAs are nearby. In other words, we support that "subjective" information can be fully adopted in case that the STAs are close to each other and therefore share similar communication conditions with each of the available APs. An easy way to estimate the location/distance of the neighboring STAs is to measure the *RSSI (Received Signal Strength Indicator)* value of the transmitted signal. In order to estimate the distance from the *RSSI* value we use free space propagation model (line of sight) for simplicity reasons. In indoor environments this model is not precise but is still capable to approximate the STAs location. In free space propagation the *RSSI* is determined as:

$$P_r(d) = P_0 - 20\log_{10}(\frac{4\pi d}{l})\ \textit{dBm} \tag{4.1}$$

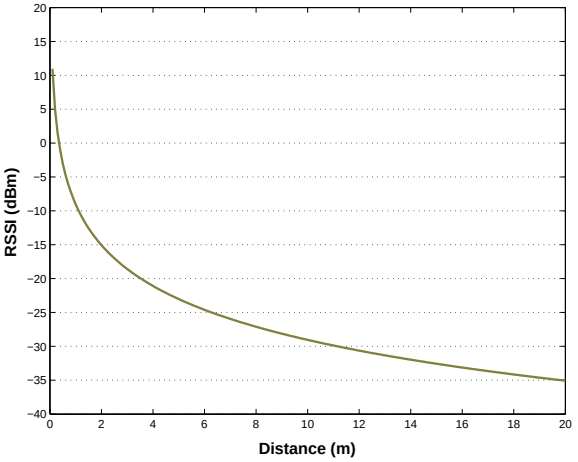


Figure 4.5: RSSI vs distance (free propagation).

where $P_0 = 30dBm$ (theoretically the maximum transmission power in 802.11), and $l = \frac{3*10^8 m/s}{2.4GHz}$. Figure 4.5 depicts the relationship between *RSSI* and the distance of the STA that transmits the measured signal. In order to measure the information accuracy, we determine an *RSSI* threshold T_{RSSI}. Besides, we can deal with the *RSSI* fluctuations that occur in real-time deployments, by measuring the mean *RSSI* value of the signal transmitted by a STA (we use a short window to calculate the mean *RSSI* value). We assume here that the STAs/APs use the same transmission power and there is no power control in the system (pure 802.11 operation). This assumption arises since we use a constant threshold T_{RSSI} in our system. However, this is not necessary because we can include the transmission power into the transmitted packet and therefore the threshold T_{RSSI} can be adapted accordingly. Furthermore, we claim that the received information is accurate in case that the mean *RSSI* value of the transmitted signal is higher than the predefined T_{RSSI}. In particular, *RSSI* helps us estimating how far the STAs/APs that transmit are and T_{RSSI} gives us the ability to receive accurate information from the STAs/APs that are close (and therefore it is possible that they face the same channel conditions). In our experiments (simulation environment) we have seen that the higher T_{RSSI} values we obtain the more accurate this information is. T_{RSSI} depends on the conditions of each system. Therefore, the system manager must adjust the threshold value according to the operational conditions (indoor or outdoor environment).

We must mention here that it is difficult to predict the radio propagation especially in indoor en-

vironments, due to propagation effects (scattering, diffraction, reflection, etc.) and the variability of the environment [74]. Consequently, the accuracy of the *RSSI*-based distance estimation may vary in these environments. In our framework we have used the simple approach based on the received signal, in order to provide a baseline of the framework. Since we do not focus on the way we will choose the criteria for the approximation of the nodes "locality", the simple algorithm of using *RSSI* provide a lightweight system solution. Handoff is a time critical procedure and therefore, it must be executed seamlessly and avoiding the effects of additional delays. The accuracy of the *RSSI*-based distance estimator can be improved in case that we use more sophisticated techniques [15, 46].

The communication between the APs is totally "orthogonal" to the communication between the STAs. In particular, in multi-cell WLANs the APs communicate through their wired connections and in wireless mesh networks the APs use the wireless backhaul to communicate. Especially in wireless mesh networks the APs can be equipped with a second interface for the backhaul communication (based on the network architecture) or use separate channels. Therefore, we can claim that the cooperative information sharing between the APs is performed independently and in parallel with the ad-hoc cooperative information sharing during the measurement periods.

The main part of our framework is the cooperative handoff mechanism that uses the information obtained from the previous procedures and provides seamless handoffs in the network. The flow diagram in figure 4.6 depicts the basic steps that are executed during a cooperative handoff procedure. We describe in detail the main steps of this mechanism:

Cooperative handoff

- **STEP 1**: STA realizes that it must find a new AP (based on the underlying association/handoff decision protocol) and initiates a handoff procedure. So, it sends a *neighbor report request* to the AP (old AP) that is currently associated with. The *neighbor report request* can be imported to the *probe request* frame that the STA sends in order to probe an AP and receive useful information (in 802.11-based scanning procedure).

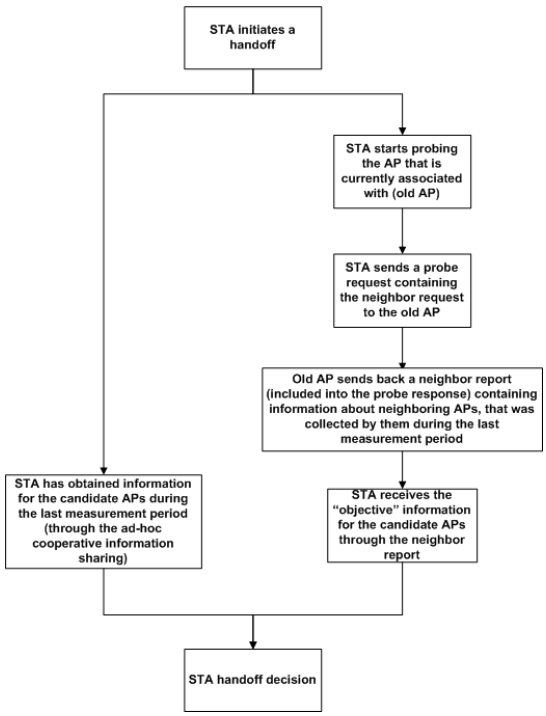


Figure 4.6: Cooperative handoff procedure.

- **STEP 2**: Old AP sends back a *merged neighbor report* to the STA. The *merged neighbor report* contains information about its neighboring APs, that has been obtained during the last measurement period. In particular, the *merged neighbor report* use several information fields that are part of the *Optional Subelements* super field (figure 4.3) and carry information for each neighboring AP. The *merged neighbor report* can be incorporated into the *probe response* frame that the AP sends back to the STA during the 802.11-based scanning process. *Neighbor report* contains similar information to *beacon report*. The main difference here is that the *neighbor report* contains additional information about "objective" characteristics of the new APs (that the STA receives through the old AP).

- **STEP 3**: STA comes up with a handoff decision based on the underlying association/handoff decision protocol that is applied in the network using: a) The information obtained during the STEP 2, and b) The information for the neighboring APs that the STA has obtained through the ad-hoc cooperative information sharing procedure, that was executed during the

last measurement period. We must make clear here that in our framework every STA that initiates a handoff procedure uses both types of information (a and b) to come up with a handoff decision.

An important observation here is that our cooperative handoff mechanism gathers handoff information during a *probe request* (the *neighbor report request* is incorporated into the *probe request*) and a *probe response* (the *merged neighbor report* is incorporated into the *probe response*) exchange between the STA and the AP. The traditional 802.11-based scanning process wastes approximately the same time in scanning just one channel, since each STA must keep listening to a channel for a constant time in order to hear all the beacons that are transmitted by the neighboring APs and then scan the next channel. Therefore, our mechanism is much faster in gathering the information that the STAs need and the added overhead is quite small (less than an 802.11-based one-channel scanning). In addition, the communication between the APs can be independently executed (during the measurement periods) from a handoff procedure. In this way the information from the neighboring APs (to the old AP) will be immediately available to the STA, when a cooperative handoff procedure is executed.

The ad-hoc cooperative information sharing plays an important role in our framework since there are situations where the old AP cannot be aware of the operational conditions of all the candidate APs for association. In a mesh environment the APs communicate over a wireless backhaul network and a candidate AP could be placed out of the transmission range of the old AP. Besides, in multi-cell environments a candidate AP could lose connection with the old AP or it could belong to another subnetwork where the communication with the old AP is impossible. For example in figure 4.7 we assume that STA3 is currently associated with AP1 and it initiates a handoff process. AP1 (old AP) cannot be aware of the operational conditions of AP2 (using the neighbor report mechanism) because AP2 is located out of the transmission range of AP1. In this case the STA3 receives this information from STA4 and STA5, through the ad-hoc cooperative procedures. Furthermore, we use ad-hoc cooperation in order to obtain "subjective" information (uplink channel conditions, etc.). This information cannot be obtained using inter-AP cooperation (neighbor report) because

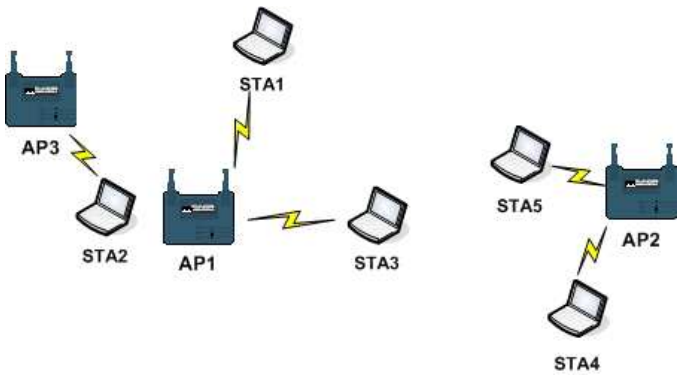


Figure 4.7: Special case: Cooperative Handoff.

the APs are not aware of these operational parameters.

If the STA decides that the "subjective" information is accurate, then it has all the information it needs to proceed with the handoff decision. In the opposite situation, since the STA considers the "subjective" information as inaccurate, it has to find a way to figure out the channel conditions between itself and the active APs in the neighborhood. In the existing approach, the STA could start scanning the available channels and get measurements about the neighboring APs. In our scheme the STA is aware of the available APs and the channels they currently use, by exploiting the "objective" information it has obtained. Thus, instead of scanning all the available channels, it directly "jumps" to the active channels, saving in this way significant time and decreasing the scanning delay.

Another issue that arises in our cooperative handoff framework is the possible greedy behavior of the STAs that share information about the active APs in the network. In other words, one or more STAs can misbehave in the system and send fake information to their neighboring STAs. In this way our cooperative handoff framework does not perform effectively since it does not have the correct information. Our scheme assumes that a trusted information exchange has been established in the network. The issue of the trustworthy among the stations is out of the scope of our work and it can be achieved using authentication techniques.

Before closing this section we must note that in our cooperative framework we use a separate control channel for information exchange. An interesting approach would be to equip the STAs with

a second communication interface for information exchange. In other words, we could keep the first interface for data communication and the second for channel scanning and control information sharing. This approach would gain in performance since we would avoid control channel switching delays. However, this is not a realistic scenario while most end user devices are not equipped today with a second interface (cost reasons, etc.). This is the main reason that leads us to choose control channel communication in our framework. Nevertheless, this could be an additional option in our framework.

4.2 System evaluation

We have implemented our cooperative handoff framework using OPNET [72]. Our mechanisms were built on top of the IEEE 802.11 standard in order to achieve backward compatibility. We have modified the main control frames (beacon, probe frames) in order to simulate the basic measurement mechanisms that are introduced by 802.11k and incorporate the appropriate information in them. The light modifications that we have introduced in the basic functionality of the IEEE 802.11 standard do not affect the performance of the network. In our simulation study we compare our framework to the scheme proposed in [86] and to 802.11. The work in [86] proposes a selective scanning algorithm and a caching mechanism in order to reduce the delay introduced by the scanning phase.

As far as the overhead and the communication cost are concerned, it is true that our cooperative mechanisms introduce an overhead in the performance of the network since now the STAs/APs have to switch to the control channel (in a periodic basis) in order to gather handoff information from the neighbors. Besides, several control frames must be transmitted during the periodic 802.11k-based measurement periods in the network. However, our framework does not introduce higher overheads and communication costs as compared to 802.11k. As we have mentioned, our scheme is built on top of the main mechanisms determined by the 802.11k standard and it is fully compliant with it. More information about the performance of the 802.11k standard can be obtained in [61].

Our simulation study takes into account the communication costs and the extra delays that are present in our framework, during the execution of our mechanisms. The simulation results declare that our cooperative handoff framework gains in performance as compared to other schemes. The main reason for this improvement is that in our framework we avoid unavailing channel scanning. Besides, the information sharing that is introduced between the STAs/APs during the measurement periods provides seamless handoffs in the network, avoiding in this way large delays and traffic interruptions. In more detail, the overhead that our mechanisms add is approximately similar to the overhead added by the one channel scanning procedure which is significantly smaller than the original overhead (in 802.11-based handoff procedure), that is equal to this time multiplied by the number of the channels that are scanned (more details will be given later in this section). Therefore, the main outcome of this work is that the number of the scanned channels is significantly reduced (compared to 802.11 channel scanning).

As described before, 802.11k introduces mechanisms for information exchange during a period called measurement period. In our scheme STAs use these mechanisms in order to collect information related to the available APs in their neighborhood. The duration of the measurement period as well as how frequent the period is initiated, is not defined by the standard. In order to study how the measurement period affects the performance of our mechanism and the overhead that is introduced, we run several experiments on a multi-cell wireless network of 5 partially overlapped cells and 65 STAs (we give more details about the simulation environment in the next subsection). Figure 4.8 depicts the average transmission delay (average delay of all transmissions in the system) in the system as the measurement period (x axis) and the measurement intervals (y axis) change. As we can see in this figure the more often the measurements are taken place, the more accurate is the information that is exchanged. However, the overhead increases due to frequent information exchange in the network and the average transmission delay is getting higher. The average transmission delay is increased too, when the frequency of the measurements is increased (measurement interval). Our system is not able to obtain "up to date" information during a cooperative handoff procedure and therefore, the performance of the handoff mechanism decreases. Additionally, large

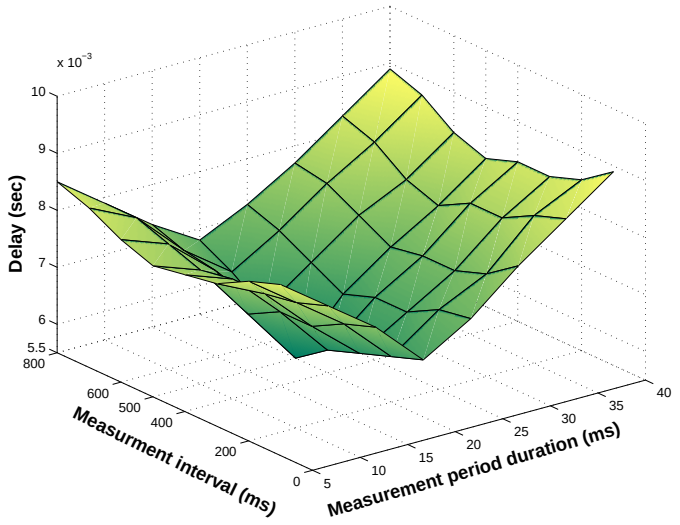


Figure 4.8: Optimal interval values for the measurement periods (STAs and APs follow these intervals).

measurement periods increase significantly the overhead too. On the other hand, when we use very small measurement periods, our mechanism does not "have the time" to take into account the "up to date" information that is carried in the control frames. Consequently, the average transmission delay increases. In figure 4.8 we can observe that the optimal system operation (minimum transmission delay) is achieved when the measurement period lasts for 20*ms* and it is initiated every 500*ms* (we use these values in our simulation study). We must mention here that the aforementioned values resulted from our simulation study. The duration of the measurement period and its periodicity is a system designer decision. Therefore, the system designer must adapt the measurement period to the properties of the system.

Figure 4.9 depicts the accuracy of the RSSI based distance estimation used in our system. We observe that the estimated distance is close enough to the real distance of STAs/APs that transmit.

4.2.1 The multi-cell scenario

We first study a multi-cell 802.11g network that consists of five partially overlapping cells. In such simple topologies we can control the parameters of our system and therefore we can have a clear view of the performance of the proposed protocols. The STAs are uniformly distributed (at random)

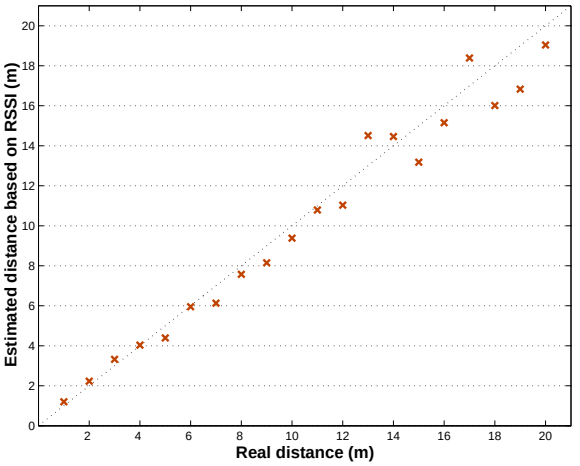
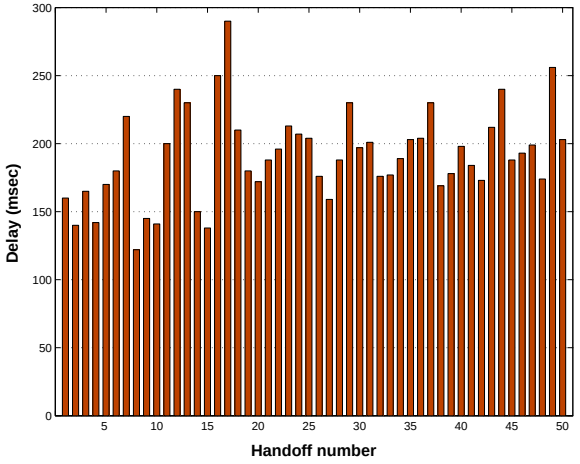


Figure 4.9: RSSI based distance estimation accuracy.

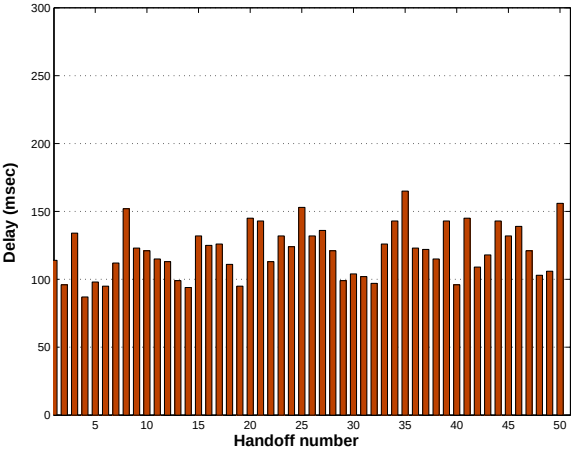
in the network and their data frames are transmitted at 1024kbps (we consider CBR traffic). We vary the number of source/destination pairs in order to vary the overall load. The source and destination nodes are chosen randomly among the nodes in the network. We compare the performance of the basic 802.11-based handoff mechanism to the performance of our 802.11k compliant cooperative handoff framework as the communication interference changes during the network operation. In order to effectively evaluate the performance of our framework we consider two cases: a) The communication load is represented by the number of STAs that are associated with an AP and b) The communication load is represented by the airtime metric introduced in our previous work [8] (the measured communication load in a) and b) is used as described in our cooperative procedures). In particular, the airtime cost of STA $i \in U_a$, where U_a is the set of STAs associated with AP a, is:

$$C_a^i = \left[O_{ca} + O_p + \frac{B_t}{r^i} \right] \frac{1}{1 - e_{pt}^i} \tag{4.2}$$

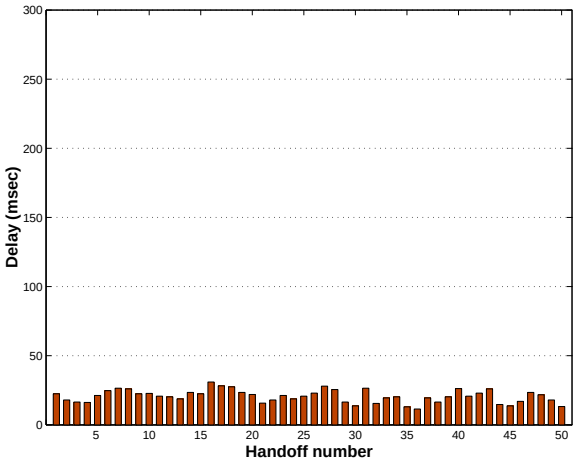
where O_{ca} is the channel access overhead, O_p is the protocol overhead and B_t is the number of bits in the test frame. Some representative values (in 802.11g networks) for these constants are: $O_{ca} = 335\mu$s, $O_p = 364\mu$s and $B_t = 8224$bits. The input parameters r^i and e_{pt} are the bit rate in Mbs, and the frame error rate for the test frame size B_t, respectively. More information about this metric and the underlying association/handoff decision mechanism can be obtained in [8]. It



(a) 802.11 performance.



(b) Selective scanning performance.



(c) Cooperative framework performance.

Figure 4.10: Handoff delays with stationary STAs.

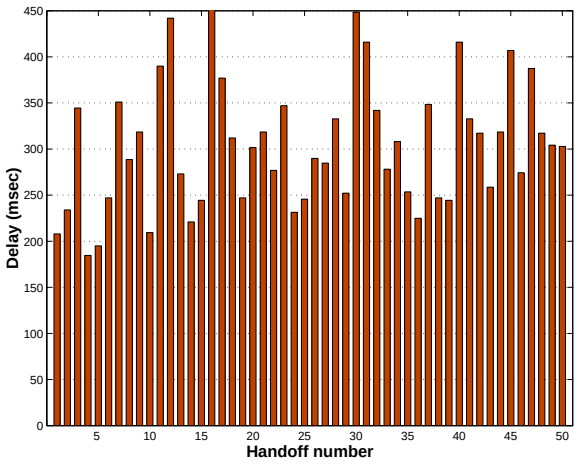
is clear that in the second case we take into account channel quality information (error rate and transmission rate), which are qualitative measurements, contrary to the first case where we just take into account the number of the associated STAs.

In the first simulation scenario we support 65 STAs (uniformly distributed at random) in the multi-cell network. We measure the handoff delays in the system when our cooperative mechanism is applied in comparison to the selective scanning algorithm proposed in [86] and to 802.11. In particular, we measure the delay of each handoff that is present in our system (x axis represents the handoff number) and we calculate the average handoff delay values. In order to evaluate the per-

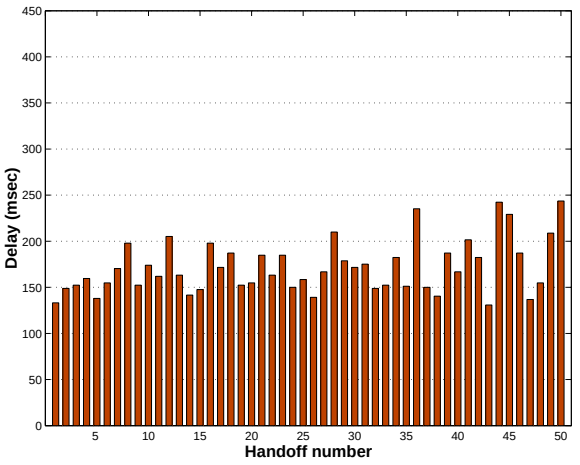
Table 4.1: Average Handoff Delays

	Stationary STAs	**Mobile STAs**
802.11	191 *ms*	303.39 *ms*
Selective Scanning	120.96 *ms*	171.6 *ms*
CoopHandoff	20.73 *ms*	22.69 *ms*
Selective Scanning Impr.	36.67%	43.44%
CoopHandoff Impr.	89.14%	92.52%

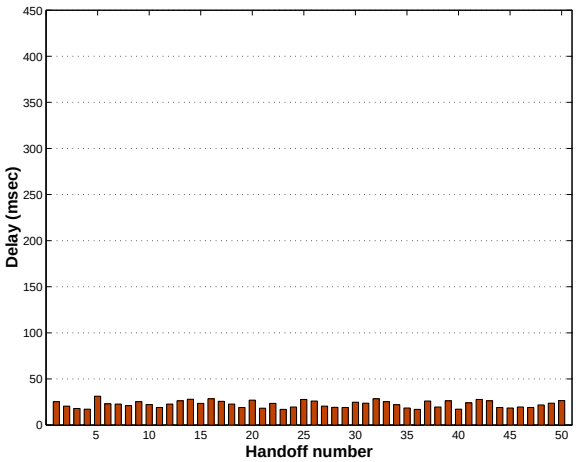
formance of our mechanisms we consider both stationary STAs and mobile STAs. We use random waypoint mobility model, where the velocity is chosen randomly between 1 and 20 *m/s*. Figures 4.10(a), 4.10(b) and 4.10(c) depict the handoff delays during the pure 802.11-based handoff mechanism execution, the selective scanning algorithm application and our scheme. In this scenario the STAs are stationary. In order to vary the channel conditions we add interference generating *jammers* that are periodically active in our system. When *jammers* are active, they continuously transmit jamming packets that cause interference. In this way we force the stationary STAs to handoff to a new AP, where interference is limited. Selective scanning improves the performance of the 802.11-based handoff mechanism using a channel mask, scanning in this way a small subset of the available channels. It is clear that our system achieves lower handoff delays due to the fact that pre-handoff information is obtained rapidly (without scanning). In figures 4.11(a), 4.11(c) and 4.11(b) we observe the handoff delays in a network that supports random STA mobility. The outcome is similar to the previous experiment. The proposed framework achieve quite lower handoff delays. Table 4.1 compares the average handoff delays between 802.11, the selective scanning algorithm and our cooperative framework. An important outcome is that our mechanisms improve the 802.11-based handoff delay by approximately 89% when we have stationary STAs and 92% when we support mobile STAs in our system. We allegate that this significant delay improvement will play an important role in the improvement of the end-to-end network performance. More details about this claim will be provided in the remaining section.



(a) 802.11 performance.



(b) Selective scanning performance.



(c) Cooperative framework performance.

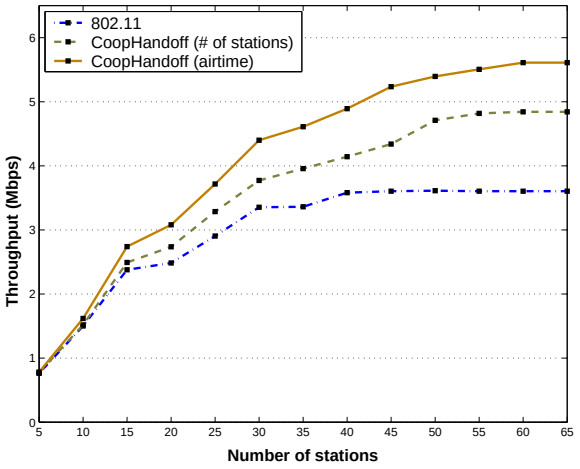
Figure 4.11: Handoff delays with mobile STAs.

During our second simulation scenario the number of the associated STAs in the network increases from 5 to 65 (STAs are uniformly placed in the network). We measure the network throughput, the average transmission delay and the data dropping. These measurements are representative and reflect the system performance under different operational conditions. In order to effectively evaluate the performance of our cooperative framework we consider two cases. The underline association decision mechanisms use: a) the number of STAs as the load metric and b) the airtime cost as the load metric. In particular, the association decision mechanisms avoid overloaded APs using these metrics (where the number of the associated STAs is large in the first case, and in the second case where the cumulative airtime cost in the cell is high).

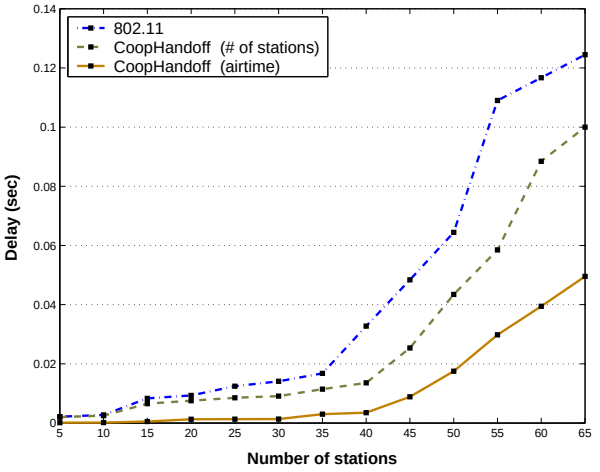
Figure 4.12(a) depicts the network throughput as the number of the associated STAs in the network increases. We compare the throughput values that are achieved during the execution of the basic 802.11-based handoff scheme and our cooperative framework. It is clear that the highest throughput values are achieved when we apply our cooperative handoff mechanisms since they speed up the handoff procedure. Airtime mechanism achieves the best performance because it takes into account channel quality information for both uplink and downlink communication and so it uses a more representative load metric than in the case we consider the number of STAs. In low load conditions we observe a quite small throughput improvement when we use the proposed mechanisms. In high load conditions, throughput increase is higher. The maximum throughput improvement that is achieved by our cooperative handoff mechanism is approximately 55% (when we have 65 associated STAs). It is important to notice that the 802.11 network throughput is stabilized when we have 45 associated STAs in the network. This means that after this point the provided QoS in the network is getting worse as the number of the STAs in the network increases. On the other hand, our cooperative framework expands the network capabilities and maximizes the network throughput in presence of 65 associated STAs in the network.

In figure 4.12(b) we observe the average transmission delay in the network. It is clear that in low load network operation the average transmission delay of 802.11 is quite small and close to the average delay that is achieved by our cooperative mechanisms. When the number of the associated STAs increases over 35 the average delay of 802.11 is getting extremely high. In contrary, our cooperative mechanisms provide an additional performance improvement to the airtime mechanism and keep the transmission delay in low level. The 802.11-based handoff policy is quite static and that causes some cells to be overloaded while the number of the associated STAs increases. Our approach provides fast dynamic reassociations/handoffs in order to keep a balanced network operation. The high 802.11 scanning delays are avoided as our cooperative mechanism “grants” the appropriate information to the STAs that are trying to reassociate with new APs.

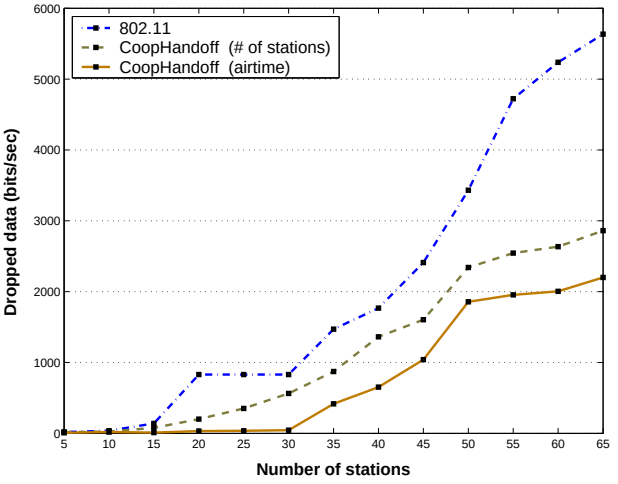
Figure 4.12(c) depicts the amount of packets dropped due to channel errors and collisions in the communication. As we can see, our mechanisms achieve lower number of dropped packets. The so-



(a) Average throughput.



(b) Average transmission delay.



(c) Average dropped data.

Figure 4.12: Simulation results for the multi-cell scenario.

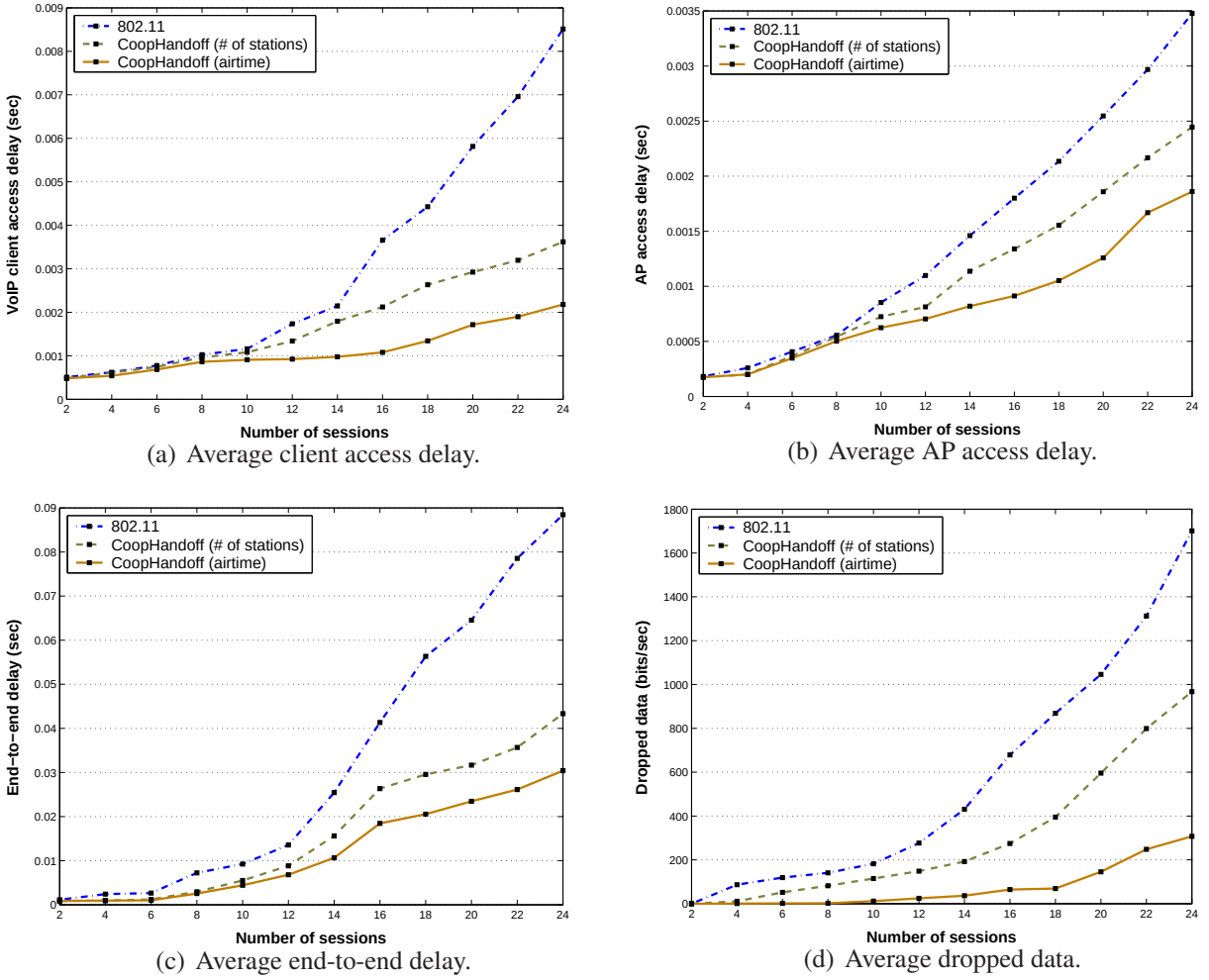
phisticated channel quality based association policies that are introduced by the airtime mechanism and our fast cooperative reassociation procedures provide a balanced network operation. STAs that face poor channel conditions and high number of dropped packets, perform fast handoffs in order to improve the network efficiency while the underling airtime association mechanism optimizes the STAs handoff decision.

4.2.2 The mesh network scenario

In order to measure the end-to-end network performance, we study the application of the proposed mechanisms in an 802.11-based wireless mesh network. We simulated a wireless mesh network in the OPNET simulation environment. The wireless routers that are provided by the OPNET wireless module are part of the backhaul network. The peripheral routers serve as APs as well. In our simulation we use 6 peripheral routers (mesh APs) and 4 backhaul routers (mesh Points). We implemented RM-AODV that is introduced by 802.11s [43] standard and we applied this routing protocol at the mesh backhaul (we can apply any QoS-aware routing protocol at the mesh backhaul in order to evaluate our framework). The STAs are uniformly distributed (at random) in the wireless mesh network. For the communication between the wireless routers in the backhaul network, we use the physical model of IEEE 802.11a OFDM physical layer. The supported physical rate is 12 Mbps. The STAs are associated with the available peripheral APs. We simulated a VoIP application in the 802.11-based wireless mesh network, which is a QoS sensitive application. In our simulations we uniformly placed several VoIP clients in the network. We run different simulation scenarios where we varied the number of the VoIP sessions that are supported in parallel.

First of all we measured the average local client access delay in the network. In practice, this delay reflects the time that the packet is generated until it leaves the client interface. The number of the sessions that are supported in parallel increases from 2 to 24. Figure 4.13(a) depicts the average VoIP client access delay. Our cooperative mechanism (with the airtime metric) achieves lower client access delays in the network. Consequently, our cooperative framework provides fast handoff procedures and keeps the client access delay in low level. The traditional 802.11 operation overloads the network and therefore increases significantly the access delay of the clients. In high load conditions the delay improvement that is introduced by our mechanism is very high.

Figure 4.13(b) depicts the average local AP access delay in the network. This delay is the time passed from the arrival of a VoIP packet at the AP until the moment that it is ether successfully transmitted over the wireless mesh network or dropped. As we see we get similar results to those



(a) Average client access delay.

(b) Average AP access delay.

(c) Average end-to-end delay.

(d) Average dropped data.

Figure 4.13: Average delays and dropped data in VoIP.

of the client access delay. In pure 802.11 the overloaded APs (in high load conditions) have a lot of traffic to forward to the mesh backhaul network. The main consequence is that the VoIP packets have to wait for a long time to be transmitted by the APs, introducing in this way high AP access delays.

In figure 4.13(c) we observe the average end-to-end delay in the VoIP packet transmission. The end-to-end delay is affected by the previous two kinds of delays that we have described and the routing delay that is introduced in the backhaul network. In our cooperative framework we achieve low end-to-end delays in the network. Especially in the airtime mechanism operation the delay improvement is very high. This improvement is true due to the fast VoIP clients/APs access

in the network and the fast handoff that is provided. We allegate that the most interesting result is depicted in figure 4.13(c), where the pure 802.11 operation can support at most 14 sessions in parallel while our cooperative framework supports 24 sessions. Therefore we have a network performance improvement of approximately 66%.

The last figure (figure 4.13(d)) depicts the dropped packets during the operation of the mesh network. Channel errors and packet collisions are the main reasons for this packet dropping. In 802.11 the number of dropped packets is high. Our proposed mechanisms decrease this number and manage to keep it low even in high load conditions.

Concluding this section we summarize the key achievements of our cooperative framework that are highlighted in our evaluation study:

- Adaptability of the measurement mechanisms defined in 802.11k.
- Lower handoff delays, compared to 802.11 and to selective scanning approach.
- Seamless mobility management in the network.
- Efficient scalability and network performance in wireless multi-cell environments.
- Support of QoS-sensitive applications in dynamic wireless mesh environments.

Chapter 5

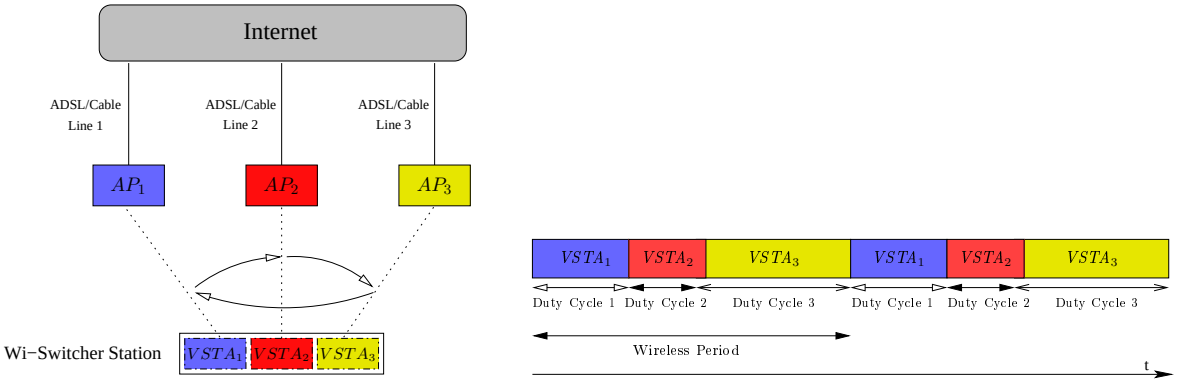
Multiple AP Association in 802.11 Wireless Networks: Optimizing TCP Performance

In this chapter we present an in-depth analysis of the effect of TDMA wireless multi-AP access on TCP and then we design a low cost resource allocation algorithm, named *min-max disconnection time*, that minimizes the impact of TDMA access on TCP. Section 5.1 presents our wireless multi-AP implementation used in the experimental tests. Then, section 5.2 investigates the performance degradation of TCP on the multi-AP TDMA scenario, and introduces an accurate analytical model. Section 5.3 validates the analytical model via experiments and simulations. The resource allocation algorithm is introduced and validated in section 5.4.

5.1 Connecting to Multiple APs with Off-The-Shelf Hardware

In this section, we briefly describe WiSwitcher [36], the experimental 802.11 station that will be used in the rest of our description as a base for the experimental tests.

Let us consider the scenario with one station in Fig. 5.1(a). The WiSwitcher station, configures 3 virtual stations $VSTA_1$, $VSTA_2$ and $VSTA_3$. Each of these *VSTAs* connects to one DSLAM (Digital



(a) Topology. (b) Relation between *duty cycle* and *wireless period*.

Figure 5.1: Time Division Access to Multiple APs

Table 5.1: Main Variables

AP_i	i-th AP
N	Number of AP backhauls
$T\ (ms)$	Wireless Period
$VSTA_i$	i-th Virtual STAtion, associated to AP_i
d_i	End-to-end wired delay
$f_i\ (\leq 1)$	Duty Cycle for the i-th virtual station
$g_i\ (\geq 1)$	Number of Slots for the i-th virtual station
G	Total number of slots
C_j	Disconnection cost for the j-th slot
SlotTime	Minimum slot size

Subscriber Line Access Multiplexer)[1] via its AP in range, independently of the AP radio-frequency (see figure 5.1(a)).

As we can see in Fig. 5.1(b), WiSwitcher assigns the control of the card to a $VSTA_i$ for a given percentage of time, called *duty cycle* f_i (with $\sum_i f_i = 1$). During this time, $VSTA_i$ transmits/receives frames over the AP_i backhaul while the other *VSTAs* (and the corresponding APs) can only buffer packets. We denote *wireless period* T as the amount of time to cycle through all the *VSTAs*. A summary of the main variables used in this section and the rest of this chapter is given in Table 5.1.

[1]The DSLAM is a network device, located in the telephony exchanges of the internet service providers, that connects multiple customer Digital Subscriber Lines to a high-speed Internet backbone line using a multiplexing technique.

5.1.1 MAC Protocol

WiSwitcher manages the multiple backhaul connections relying on the 802.11 PS mechanism. Particularly, referring to the example in Fig 5.1(b):

1. During the reserved *duty cycle*, *VSTA*$_1$ transmits and receives data according to the 802.11 DCF protocol. The other *VSTAs* are in PS mode, and hence they (and the corresponding APs) can only buffer packets.

2. When the *duty cycle* expires, *VSTA*$_1$ sends a frame to inform AP_1 that is going to PS mode and waits for its MAC ACK. According to the 802.11 protocol, AP_1 starts to buffer the packets directed to it.

3. WiSwitcher assigns the control of the card to *VSTA*$_2$ and switches to the AP_2 radio-frequency.

4. *VSTA*$_2$ sends a frame to announce that it can send/ receive traffic and waits for its MAC ACK.

5. The process continues until the station has cycled through all the *VSTA* (a *wireless period* T).

The reader is referred to [36] for an in-depth description of the MAC implementation.

5.1.2 Network Layer Functionalities

At the network layer, three functionalities are needed:

Scheduler: it assigns the *duty cycle* f_i to each $VSTA_i$. In this work, the *duty cycles* are fixed via user-space commands[2].

Load balancer: it splits the traffic at each WiSwitcher station across the different *VSTAs*. The procedure is done on a per-flow basis and assigns TCP flows to APs so that the total load sent to each AP is proportional to the *duty cycle* f_i. This minimizes problems as the initialization of a long-lived TCP connection on an AP with small *duty cycle*. The load balancer is similar to the one proposed in [47].

[2]We are currently implementing a scheduler for the management of this set of duty cycles.

Reverse-NAT: In order to guarantee transparency to higher layers, we implement a reverse-network address translation (NAT) module with two functions: i) assure that the packets leave the host with the correct source IP address (i.e. the one corresponding to the outgoing *VSTA*, as assigned by the AP) and ii) that the incoming packets are presented to the OS with the expected IP address, that in our implementation is a dummy IP address. Reverse-NAT modules were also present in [47] and [70].

5.2 TCP over TDMA

In this section, we first show an experimental test that enlightens the correlation between the end-to-end throughput and the delay added by the TDMA approach, and then we introduce an analytical model that characterizes the TCP RTT for a station connected to multiple APs. The importance of the model is that not only gives insight into the problem, but it will be also used as an efficient way to validate the resource allocation algorithm later defined in this chapter.

5.2.1 Example of TCP Throughput over TDMA transmissions

We experimentally tested the effect of the TDMA approach on a (long-lived) TCP session. In Fig. 5.2, we show the TCP throughput obtained by a station spending 50% (that is *duty cycles* of $f_1 = f_2 = 0.5$) of its time to one of the two APs, as a function of the end-to-end delay. For the test, we considered a wireless period T of 100 ms, that gives a connection of $f_1T = f_2T$= 50 ms on each AP. Each point is the average TCP throughput obtained over 5 independent tests of 300 seconds, with an experimental loss rate of around 0.4%.

We can see that the station gets a similar throughput at both 50 and 100 ms of end-to-end delay, caused by the similar RTT observed at 50 and 100 ms. Indeed, the packets arriving at the AP with 50 ms of end-to-end wired delay have to wait for an extra-buffering time at the AP, due to the disconnection of 50 ms. This hampers the throughput observed by the TCP traffic[3].

[3]Note that there are *valleys* in the throughput. These are caused by slight variations in the packet losses observed in

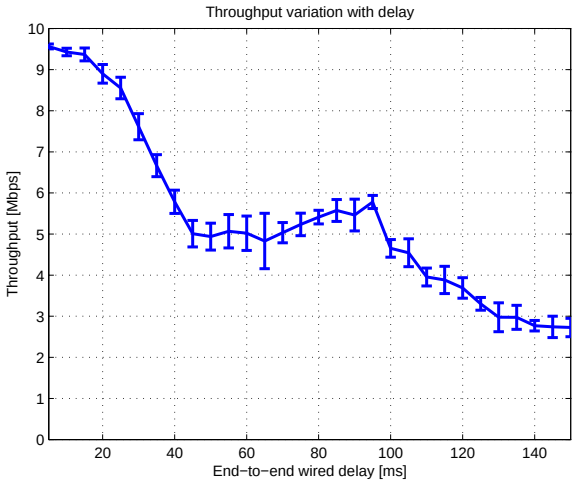


Figure 5.2: Experimental throughput connected 50% of time to one AP

5.2.2 Modeling the TCP RTT over TDMA transmissions

We can model the dependency of the TCP RTT on end-to-end delay and duty cycle by observing all the possible cases in which the TDMA technique affects the observed RTT. In what follows, we consider the uplink case (e.g. *VSTA* is sending data to a remote server), but it is straightforward to see that the RTT calculations are symmetric for both the uplink and downlink cases. We distinguish three conditions:

1. Consider the case of Fig. 5.3(1) in which the station sends the TCP data at time t_i, during its *duty cycle*. Also let us assume that the end-to-end delay d_i is such that the TCP ACK arrives from the server before the station disconnects from the AP. In that case it is easy to see that the observed RTT from TCP is indeed d_i.

2. Let us now consider the case of Fig. 5.3(2), where the station sends the TCP data at time t_i, during its active period, but the end-to-end delay d_i is such that the TCP ACK arrives from the server during the time reserved to other *VSTA*s. The AP_i will buffer the packet in its queue until the station reconnects again at time 0 of the next wireless period. In this case, the observed RTT for the TCP packets is $T - t_i$, where T is the wireless period. Note that, as

the experimental tests.

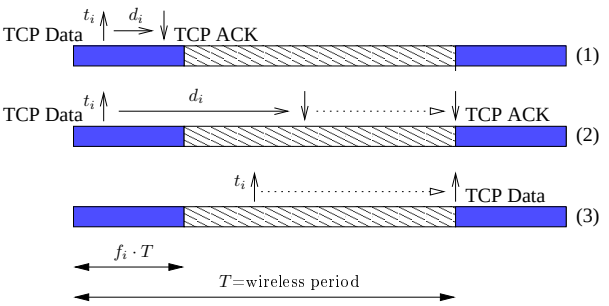


Figure 5.3: Model relation between TCP congestion control and duty cycle

long as $(1 - f_i)T > (d \; mod \; T)$, there will be always some packet that will wait in the AP_i downlink buffer because of the disconnection period.

3. Finally let us consider the case of Fig. 5.3(3), wherein the TCP data is buffered at the station at time t_i, during the sleeping period in the AP. However, we experimentally verified by monitoring the AP queues that case (3) does not occur in the TCP steady state, and no new TCP data is buffered during the sleeping period. The reason is that, in the TCP steady state case, new TCP data can only be sent when a TCP ACK is received from the server. But as we have seen, the TCP ACKs can only arrive to the station when it is connected to the AP.

Finally, in order to take into account that the station takes some time in processing and transmitting the TCP ACKs, as an approximation, we suppose that: i) the TCP ACKs arrive exponentially distributed over the *duty cycle*. ii) When $f_i T$ is very small, some of the TCP data scheduled during the connection period will inevitably be sent at the next connectivity period, due to buffering delay.

Mapping the modeled TCP RTT to Throughput

Although a wide variety of TCP algorithms are used on the Internet, the current most popular implementation is TCP Reno [88]. Then, in order to map the RTT estimation given by the model to throughput values, we used the Mathis TCP model [62], which is intended to predict TCP's end-to-end throughput as: $BW \leq \frac{MSS}{RTT} \cdot \frac{1}{\sqrt{p}}$, where RTT is the Round-Trip-Time observed by the station and MSS is the TCP Maximum Segment Size[4].

[4]This model applies to long lived connections over nearly all implementations of TCP Reno with SACK TCP. Note that, in order to use this model, the packet loss rate should be smaller than 2%, condition that was verified in all the

5.3 Evaluation

In this section we:

- validate the accuracy of the TCP RTT model presented in the previous section, comparing it with experimental results.
- show that high disconnection time severely affects the TCP performance.
- demonstrate that for, any value of *duty cycle*, the best strategy is to reduce the wireless period T as much as possible.
- show that the selection of the wireless period T must be done based on the smaller *duty cycle* of the station.

In what follows we discuss the details of the setup of the related tests.

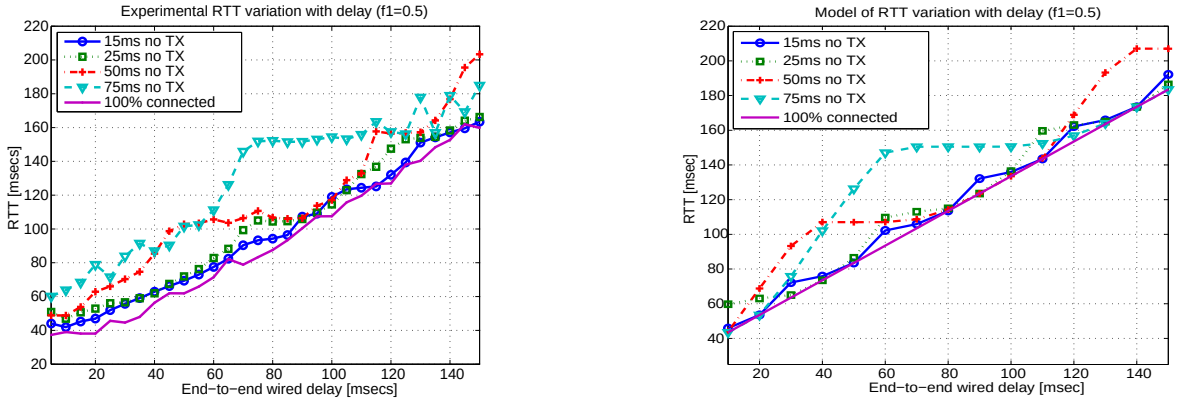
Experimental setup In each controlled test, we used laptops with Atheros-based chipsets running WiSwitcher and off-the-shelf APs (Linksys) with DD-WRT v24sp1 firmware. On the wireless station, automatic rate selection, wireless multimedia extensions, and the RTS/CTS mechanism were disabled. Any no-standard compliant features at the MAC level were also disabled. Our station used a H/W queue with best effort parameters.

For the transport layer, we used a Linux standard TCP Reno with SACK and delayed ACK options enabled. TCP parameters were monitored using a modified version of the TCP probe kernel module and the kernel patch Web100. For each test, we established one TCP connection over an AP backhaul[5] and collected statistics using the *iperf* tool. Regarding the wired connections, we emulated the AP backhaul links through the *tc* Linux traffic shaper, varying the delay using the *netem* tool.

For each experimental test, we established one TCP Reno connection over each AP, ran 5 independent tests of 300 secs and plotted the average values obtained. To achieve independent tests,

experimental tests in our work.

[5]Note that the link utilization can increase establishing more than one TCP connection over each AP[36], which is out-of-the-scope of our work.



(a) Experimental RTT

(b) Analytical RTT.

Figure 5.4: Downlink RTT for $f = 0.5$ to 1 AP.

the station was configured so that the TCP metrics were reset after each test. Finally, in order to minimize the variability of the results, we scheduled the tests to run during the night and weekends, when there were no people in the lab.

Simulation setup The simulations are performed using the model described in Sec. 5.2.2, implemented in MATLAB using as input the experimental values of MSS and TCP congestion signals rate per packet for the Mathis formula.

5.3.1 TCP RTT Model Validation

In this section we compare the RTT values achieved experimentally with the ones of the model. For brevity, we only show one scenario, but similar results have been achieved with several other scenarios.

Fig. 5.4(a) shows the RTT values obtained experimentally as a function of the end-to-end delay, when the WiSwitcher station connects 50% of its time to an AP, and observes a disconnection time of 15, 25, 50 and 75 ms. The plot shows that the increase of disconnection time affects significantly the observed TCP RTT. Fig. 5.4(b) shows the observed TCP RTT calculated using the model for the same scenario. We can see that the analytical model accurately predicts the RTT behavior. The observed differences are the result of the variable losses observed in the experiments and the

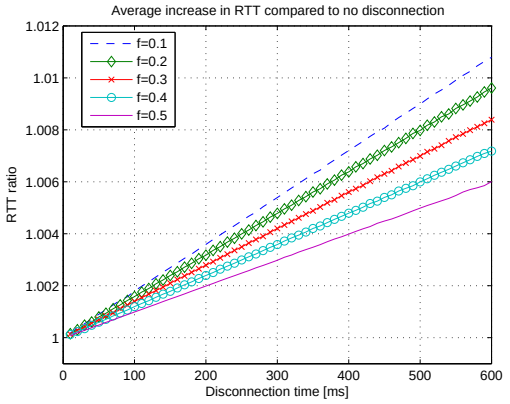


Figure 5.5: Correlation between duty cycle f_i and connection time f_iT. It is evident how a small f_iT gives benefits to the TCP RTT.

expected noise on the experimental environment.

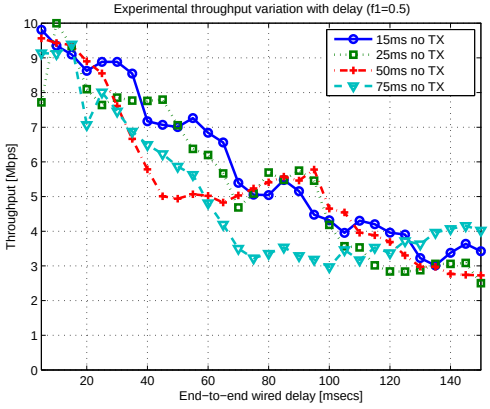
Interestingly, there are specific situations where smaller disconnection time results in higher RTT. As an example, consider the RTT in Fig. 5.4(b), for a disconnection time of 50 and 75 ms and an end-to-end wired delay of 125 ms. Here, the interplay between disconnection time and delay causes a higher TCP RTT for a disconnection time of 50 ms compared to the 75 ms case. This phenomenon gets less important at smaller disconnection times (see e.g. the RTT observed with a disconnection time of 15 and 25 ms).

However, in average, the RTT always decreases with the disconnection time. Fig. 5.5 shows the average[6] increase of RTT (labelled "RTT ratio" in the figure) for a station connected to an AP for a certain time f_iT, respect to a station connected only to 1 AP for its 100 $\%$ of time. In the figure, we depict the results for different values of *duty cycle* f_i and disconnection time $T - f_iT$ (ms). We observe that, for any value of f_i, the best strategy is to reduce the disconnection time.

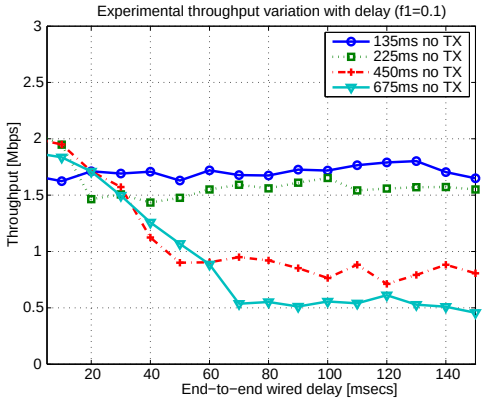
5.3.2 Impact on Throughput per TCP-flow

In this section we study the throughput observed by a TCP connection opened on one AP. Fig. 5.6(a) depicts the experimental throughput when the *duty cycle* of one AP is 50%. We observe that even

[6]For the average, we considered end-to-end delays ranging from 10ms to 5000 ms, calculated using the TCP RTT model described in previous section and plotted the average RTT in this range.



(a) Throughput with 50% of connection ($f_1 = 0.5$) to AP_1.



(b) Throughput with 10% of connection ($f_1 = 0.1$) to AP_1.

Figure 5.6: Throughput per TCP-flow with different *duty cycles* and disconnection times

for small delays, the throughput performance may be dramatically affected. As an example, when we operate with a disconnection time of 75 ms, we observe a quasi-constant throughput when the end-to-end wired delay spans from 75 to 150 ms. This is caused by the similar RTT observed at 75 and 150 ms of end-to-end wired delay.

Even more evident is the case where the *VSTA* is connected for the same amount of time — hence for a connection time of $f_1 T = \{15, 25, 50, 75\}$ ms — *but* for a smaller *duty cycle* to AP_1, e.g a 10% of its time. We can also see from Fig. 5.6(b) that the penalty in throughput is more severe as the disconnection time grows. For example, when the disconnection time is 675 ms, the average throughput is more than 3 times smaller than the throughput achieved when the disconnection time is 135 or 225 ms.

5.3.3 Minimizing the Wireless Period

Based on the analysis in the previous section, we can see that in order to reduce the impact of TDMA on the TCP throughput it is important to keep the disconnection time as small as possible. Since the disconnection time is equal to $T - f_i T = (1 - f_i)T$, this also implies that, for a fixed f_i, the wireless period T should be kept small. In order to confirm this intuition, Figure 5.7 shows the throughput achieved in experimental tests as a function of the percentage of time connected to one

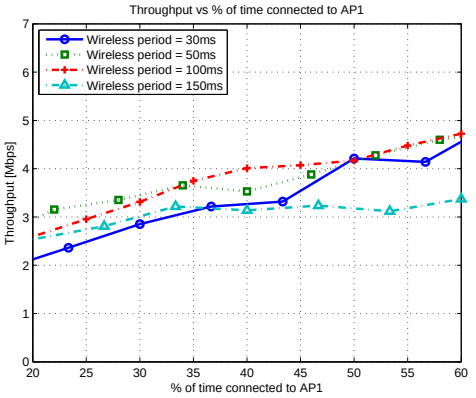


Figure 5.7: Experimental downlink throughput connected 50 % of time to one AP for an end-to-end delay of 100 ms

AP.

In the tests, we compare different wireless periods ($T = \{30, 50, 100, 150\}$ ms) and fixed the end-to-end delay to 100 ms. The results in the Figure show that similar performance are achieved with wireless periods of T=30 ms, T=50 ms, and T=100 ms while throughput can be severely affected choosing a wireless period of T=150 ms. Note also that, for T=30 ms, the station gets slightly less throughput for small *duty cycles*. In fact, there are two limiting factors: i) there is a time spent by the 802.11 card to switch AP (called *switching cost*) ii) the frequent AP switching introduces a packet loss rate of around 0.01-0.08%, caused by an inefficient management of the transmission queues at the driver level in off-the-shelf APs [36]. Since the packet loss rate and the switching cost affect more severely smaller *duty cycles* rather than bigger ones, we can conclude that the selection of the *wireless period* T should be based on the smallest *duty cycle* at the station.

5.4 Increasing the Aggregated Throughput

In this section we use the experience learned from the previous section to improve the aggregate throughput at the station by i) introducing the assignment of slots to each *VSTA*, and ii) allocating the slots via a distributed resource allocation algorithm. The objective is to minimize the disconnection time such as the TCP degradation is minimized.

5.4.1 Concept of Slotted Operation

Instead of connecting to each AP_i for a consecutive amount of time $f_i \cdot T$, we introduce the concept of slot assignment and give $g_i \geq 1$ slots to each *VSTA*$_i$. For this scope, according to the analysis in previous section, we first define *SlotTime* as the minimum amount of time allowed in the system at which the effect of the switching cost and the packet losses can be neglected on the connection with the smallest *duty cycle*[7].

Fig. 5.8(a) shows an example of how the upper-layer scheduler and the slot assignment jointly work. We calculate:

- the wireless period as: $T = \frac{\textit{SlotTime}}{\min_i \ f_i}$, that is, the procedure minimizes the wireless period T.
- the number of slots locally assigned to each *VSTA*$_i$ as: $g_i = \lfloor (f_i T)/\textit{SlotTime} \rfloor$. This gives a total number of slots of $G = \sum g_i$.
- the slot size per *VSTA*$_i$ as: $\textit{SlotTime}_i = \frac{f_i T}{g_i}$.

Note that this solution can be applied to the systems proposed in [47, 70] too.

Once selected T, $\{g_i\}$ and $\{\textit{SlotTime}_i\}$, we can construct a resource allocation algorithm that, given the set of duty cycles f_i provided by the upper-layer scheduler, assigns the set of slots to the *APs* that minimizes the overall disconnection time for all APs.

This requires the following definitions.

Disconnection Cost Let us define $\mathbf{S}_i = [S_i(1), S_i(2), \ldots S_i(g_i)]$ the vector that indicates the slot positions in the range $[1, G]$ for *VSTA*$_i$, with $S_i(g_i + 1) = S_i(1)$ and $S_i(l) \neq S_j(m)$ for any $i, j = 1, \ldots N$, with $i \neq j$, $l = 1, 2, \ldots g_i$ and $m = 1, 2, \ldots g_j$. Besides, we define the cost (slot duration) of each slot as the slot size of the *VSTA*$_i$ that uses the slot:

$$C_{S_i(l)} = \textit{SlotTime}_i \quad \forall i.$$

[7]Particularly, in our tests, we got 85% of the expected throughput — defined as the throughput that would be get without any cost of switching — with only 6 ms of connection time over a wireless period of 12 ms and 95% of the expected throughput with a connection time of 15 ms over 30 ms (see [36] for details).

In order to measure the disconnection cost of the *VSTA*$_i$ during two transmissions in the slots $S_i(l)$ and $S_i(l+1)$ we take into account the costs of the intermediate slots $C_{S_i(l)+1}, \ldots C_{S_i(l+1)-1}$. Therefore, we introduce the following cost function:

$$c_{i,l} = \sum_{j=S_i(l)+1}^{S_i(l+1)-1} C_j \quad l = 1, 2, \ldots g_i$$

Example: Let us suppose that N=3 and that the slots are allocated as follows: [*VSTA*$_1$ *VSTA*$_2$ *VSTA*$_3$ *VSTA*$_1$ *VSTA*$_2$ *VSTA*$_1$]. This gives: $S_1 = [1\ 4\ 6]$, $S_2 = [2\ 5]$ and $S_3 = [3]$. Furthermore, we suppose that *SlotTime*$_3$ = 10 ms, *SlotTime*$_1$ = 12 ms and *SlotTime*$_2$ = 15 ms. Then, we calculate the disconnection cost between $S_1(1)$=1 and $S_1(2)$=4 as $c_{1,1} = C_2 + C_3 = 15 + 10 = 25$ ms.

5.4.2 Resource allocation algorithm

We now present three different, fully decentralized, slot allocation mechanisms with different performance and computation costs that aim to minimize the impact that the TDMA multiple-AP access has on TCP[8].

Blind Resource Allocation. We have seen that using a TDMA approach increases the observed RTT of the TCP packets. We have also seen that this increase is exactly the disconnection time in the worst case, i.e., for *VSTA*$_i$, and given an allocation that produces a disconnection time of $\max_{l=1,2,\ldots g_i} c_{i,l}$, we would have

$$RTT_i = d_i + \max_{l=1,2,\ldots g_i} c_{i,l}.$$

The TCP throughput achieved by the above allocation can be approximated as

$$\frac{MSS}{[d_i + \max_l c_{i,l}] \cdot \sqrt{p_i}},$$

[8]We also tested an allocation mechanism with random assignment of the slots, using as a constraint that each slot is assigned to a given *VSTA*$_i$ with a probability equal to f_i. Although this random assignment may decrease the buffering time at the AP in certain configurations, we found that it generally increases the jitter observed by TCP, and then reduces the observed downlink throughput.

where MSS and p_i are the parameters of the Mathis model. It follows that, in order to minimize the throughput penalty caused by disconnection, we need solve the following problem:

$$\min \sum_{i}^{N} \left(\frac{MSS}{d_i \cdot \sqrt{p_i}} - \frac{MSS}{[d_i + \max_{l} c_{i,l}] \cdot \sqrt{p_i}} \right). \tag{5.1}$$

The slot assignment obtained from solving (5.1) depends on the correct estimation of the loss rates $\{p_i\}$ and end-to-end delays $\{d_i\}$. In a realistic deployment, an accurate prediction of these values may be not available. In the absence of any end-to-end delay information we can reformulate the problem simply as the maximization of the inverse of the maximum disconnection times as follows:

$$\begin{aligned} &\max_{S_i(l)} \sum_{i}^{N} 1/(\max_{l} c_{i,l}) \\ &s.t. \sum_{i=1}^{G} C_i = T \\ &\quad f_i T = g_i \textit{SlotTime}_i \quad \forall i \\ &\quad S_i(l) \in \{1,\ G\} \quad \forall i, l, \end{aligned} \tag{5.2}$$

where the variables C_i, *SlotTime*, f_i and g_i are defined in Table 5.1.

Min-Max Disconnection Time Allocation Algorithm. The blind resource allocation algorithm defined above can be prohibitively expensive. In order to reduce its complexity, we define a *min-max disconnection time* heuristic approach. This approximation algorithm considers that, in average, the TCP performance is more severely affected by the amount of time that each *VSTA*$_i$ is *not connected* to the corresponding AP_i. The algorithm operates as follows:

1. First allocate the slots to the *VSTA* with $max(g_i)$. In fact this is the *VSTA* with the highest demand in terms of throughput.

2. Next, the *VSTA*s with lower number of slots will be served one by one. At each step, the

selected i-th *VSTA*$_i$ analyzes *only* the slots not previously assigned and calculates the vector $\mathbf{S}_i$ to satisfy the condition: $\min \max_{l=1,2,...g_i} (S_i(l+1) - S_i(l))$, that is, it selects the g_i slots to minimize the maximum distance between each pair of consecutive slots assigned to the *VSTA*$_i$.

3. Finally, at the last step, the left set of slots are assigned to the *VSTA* with $min(g_i)$.

Note that the algorithm tries to minimize the disconnection time starting with the connections with higher *duty cycle*, which are the ones with higher expected throughput. Note also that the last connection to allocate is the one with the smallest *duty cycle*. For that one, the SlotTime is already chosen such as its performance is not affected.

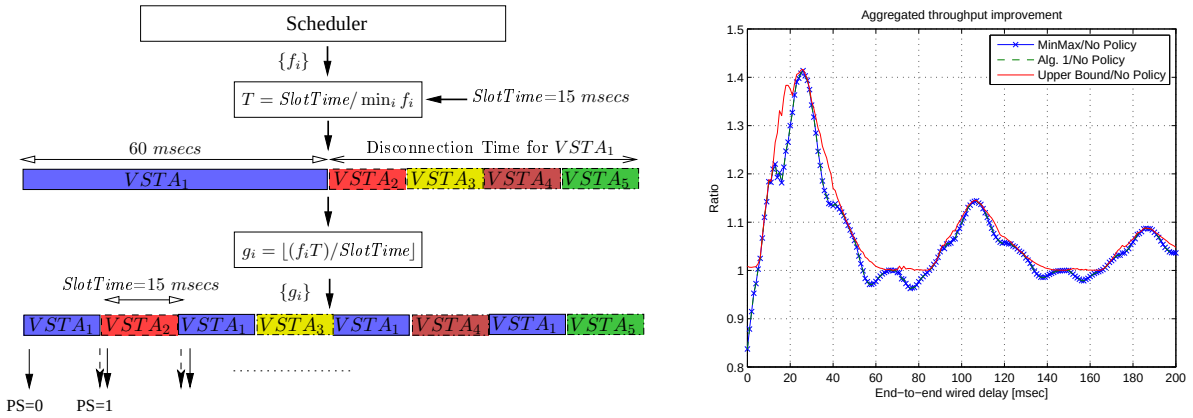
Upper-bound We also calculate the upper-bound for the TCP aggregate throughput: for each delay, we compute the TCP aggregate throughput for all the feasible solutions and select the one that achieves the maximum throughput. Note that this upper bound can not be calculated in practice and we include it for comparison purposes.

5.4.3 Evaluation results

We now evaluate the above algorithms via simulations in different scenarios. We considered one long-lived TCP flow for each *VSTA*$_i$ and we used a packet loss rate of 0.32%, measured experimentally by connecting to one AP in range with high signal-to-noise ratio, and measuring the average number of congestion signals per acknowledged packet[9].

Case 1: High number of APs. Fig. 5.8(a) shows a station is connected to 5 APs, with the scheduler sellecting the following duty cycles: f_1=0.5, f_2=0.125, f_3=0.125, f_4=0.125, f_5=0.125. The corresponding number of slots are g_1=4, g_2=1, g_3=1, g_4=1, g_5=1, and the total number of slots is G=8. In the model, we used a *SlotTime* of 15 ms, which gives a wireless period of 15 ms*8 slots

[9]We used the packet loss rate connecting to one AP, because current off-the-shelf APs implementations add at each AP a certain packet loss rate, that may limit the performance in experimental implementations. The reader is refered to [36] for more details. In this work, we do not consider these implementation issues, and assume that the limiting factor is the time spent by the 802.11 card to switch AP.



(a) Slot distribution

(b) Throughput improvement.

Figure 5.8: Min-max disconnection time allocation algorithm. Case 1.

= 120 ms. The algorithm minimizes the time of no-transmission/reception allocating firstly the slot to $VSTA_1$, then $VSTA_2$, $VSTA_3$, $VSTA_4$, and $VSTA_5$.

Fig. 5.8(b) depicts the throughput improvement versus the end-to-end delay, obtained comparing the proposed allocation algorithm (labelled "MinMax") with no resource allocation (labelled "No Policy"), that is, spending consecutive 60 ms on $VSTA_1$, and then 15 ms on $VSTA_2$, $VSTA_3$, $VSTA_4$, and $VSTA_5$, sequentially.

We observe that the *min-max disconnection time* allocation improves the throughput in all the cases, thanks to the reduction of the disconnection time. The min-max algorithm gains up to 1.5 times the throughput respect to the case without any resource allocation in the system. Quite interesting, for an end-to-end wired delay of $0 - 5$ ms, the *min-max disconnection time* algorithm has a slight lower aggregate throughput. This is because, with this very small delay, the higher number of AP switching increases the probability that the TCP packet needs to wait at the next connection period before being acknowledged.

Fig. 5.8(b) also depicts the throughput achieved by running the algorithm in (5.2) (labelled "Alg. 1"). We can see that the heuristic approach performs identically than the blind resource allocation.

Finally, we ran a test with all possible slot allocations. We verify for each delay the configuration that maximizes the equation 5.1 (labelled "Upper Bound"). We observe that, despite the high cost

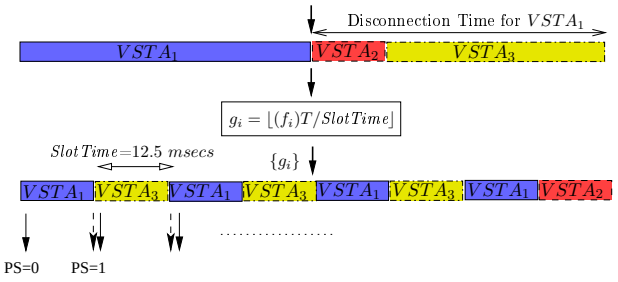
and the need for an optimal calculation of the end-to-end delay per connection and the packet loss rate, the upper bound algorithm only slightly increases the aggregate throughput observed by the station respect to the *min-max disconnection time* approach. The main reason behind this result is that the key parameter that affects the end-to-end TCP throughput is the disconnection time from the AP, which is taken into account in the *min-max disconnection time* approach, rather than the extra-buffering time at the AP.

Case 2: VSTAs have different Duty Cycle. We now consider a station connected to 3 APs, with the scheduler giving an output the set: f_1=0.5, f_2=0.125, f_3=0.375. The corresponding number of slots given by the resource allocation algorithm are g_1=4, g_2=1, g_3=3. Let us also suppose that the *SlotTime* of the system is equal to 12.5 ms.

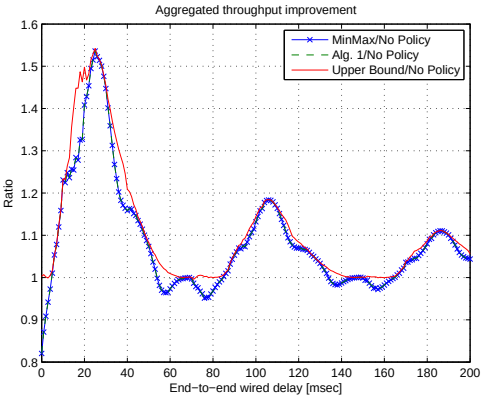
Without using the resource allocation algorithm, $VSTA_1$ would be disconnected 50 ms, $VSTA_2$ for 87.5 ms, and $VSTA_3$ for 62.5 ms. The *min-max disconnection time* algorithm increases the granularity of the AP assignment so that $VSTA_1$ will be disconnected for 12.5 ms, $VSTA_2$ (still) for 87.5 ms, and $VSTA_2$ for at most 37.5 ms. Note that the TCP throughput at $VSTA_2$ can be improved only by reducing the *SlotTime*, since it uses just one slot per period and a resource allocation algorithm cannot contribute to improve the performance of $VSTA_2$.

In Fig 5.9(b), we observe that the ratio between the aggregated throughput obtained by the *min-max disconnection time* algorithm over the one obtained without any algorithm is higher (up to 1.5 times) in most cases. Particularly, the *min-max disconnection time* allocation algorithm gets higher throughput corresponding to the scenarios where the delay added by the disconnection periods causes higher RTT. In general, the resource allocation algorithm can significantly improve the performance, since the VSTAs have different throughput demands. Our resource allocation algorithm tries to meet these demands by selecting the slot combination that minimizes the disconnection time.

Besides, the "Alg. 1/NoPolicy" line in Fig 5.9(b) is identical to the one achieved running the algorithm in (5.2), that needs 280 rounds to analyze all the feasible solutions, respect to the 6 runs needed by the *min-max disconnection time* algorithm. Finally, "Upper Bound/NoPolicy" line shows

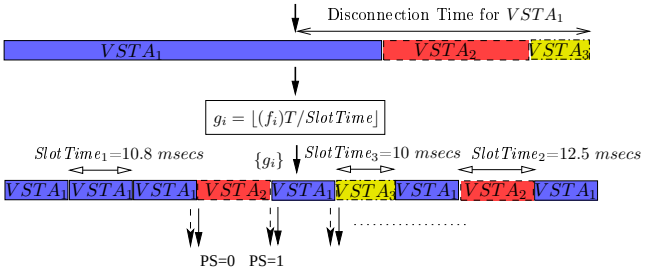


(a) Slot distribution.

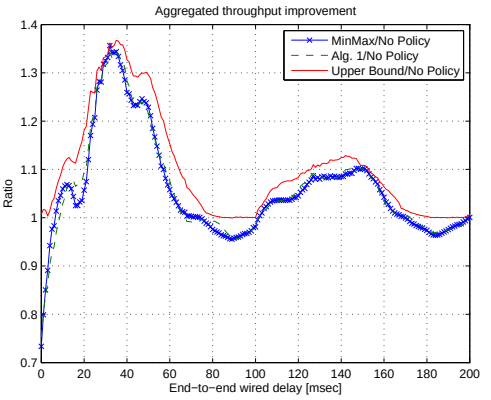


(b) Throughput improvement.

Figure 5.9: Min-max disconnection time allocation algorithm. Case 2.



(a) Slot distribution.



(b) Throughput improvement.

Figure 5.10: Min-max disconnection time allocation algorithm. Case 3.

that the throughput improvements achieved with the optimum solution (eq. 5.1) is negligible.

Case 3: Different slot size per VSTA. We finally consider a scenario where the slot time length are different per each *VSTA*, caused by a set of duty cycle equal to f_1=0.65, f_2=0.25, f_3=0.10. The corresponding slot distribution is given in Fig. 5.10(a), supposing a *SlotTime* of 10 ms. According to the slot allocation procedure defined in 5.4.1, the *VSTAs* use $SlotTime_1 = 10.8$ ms, $SlotTime_2 = 12.5$ ms and $SlotTime_3 = 10$ ms.

The results in Fig. 5.10(b) show that the *min-max disconnection time* algorithm achieves the best trade-off between performance and computational cost. Particularly, we can see from "Alg.

1/NoPolicy" line, that the throughput achieved by the *min-max disconnection time* algorithm is very similar to the one defined in the equation 5.2. Interestingly, in some scenario the *min-max disconnection time* algorithm achieves a slighly higher throughput, despite the cost of only 7 runs compared to the 252 runs needed in (5.2).

Chapter 6

Load-aware Channel Allocation in WLANs

In this chapter we present LAC, a Load-Aware Channel assignment scheme for WLANs that discovers the channel assignment that approximately *maximizes the AP throughput* in the network. Section 6.1 describes the airtime metric that we employ, based on which LAC discovers the channel with the maximum long-term throughput. In section 6.2, we present the design of LAC. We evaluate our protocol through simulations, in section 6.3.

6.1 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 [7] for the uplink channel). The airtime metric was first discussed in the 802.11s [43] 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:

$$C_{a,c}^{i} = \left[O_{ca} + O_p + \frac{B_t}{R_i^{a,c}} \right] \frac{1}{1 - e_{pt}^{c}}. \tag{6.1}$$

In (6.1), 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[1]. Some representative values for these constants, for 802.11g, are: $O_{ca} + O_p = 1.25$ms and $B_t = 8224$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:

$$\begin{gathered} \overline{C_{a,c}} = \frac{1}{N_a} \sum_{i=1}^{N_a} \left[O_{ca} + O_p + \frac{B_t}{R_i^{a,c}} \right] \times \frac{1}{(1 - e_{pt}^{c})} = \\ \left[\frac{1}{N_a} \sum_{i=1}^{N_a} (O_{ca} + O_p) + \frac{1}{N_a} \sum_{i=1}^{N} \frac{B_t}{R_i^{a,c}} \right] \times \frac{1}{(1 - e_{pt}^{c})} \end{gathered} \tag{6.2}$$

The case for uplink traffic. First, we discuss the relationship of the *airtime metric* with the average uplink network throughput. In [20] and [52] 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

[1]The transmission of test frames is necessary, in order to derive values for the computation of the airtime cost.

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 [20], 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 (6.3),

$$\begin{aligned}&\tfrac{1}{\theta(a,i,c)} = \\ &\tfrac{1}{\mu(1-\mu)^{N\alpha-1}L_i^a x_i^a} \times \Bigg[1 + N_a\mu(1-\mu)^{N_a-1}\left(T_0 - T_c + {}^1\!/_{N_a}\sum_{q=1}^{N_a} L_q^a/R_q^{a,c}x_q^a\right) \\ &+(1-(1-\mu)^{N_a-1})T_c\Bigg]\end{aligned} \tag{6.3}$$

where μ is the attempt rate (probability) in the equilibrium, $N_a = \sum_{q=1}^{N} x_q^a$ is the number of STAs associated with AP a (x_i^a is a binary value that takes value 1 in case that 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 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:

$$G(\gamma) = \frac{\sum_{k=0}^{K}\gamma^k}{\sum_{k=0}^{K}\gamma^k b_k}, \tag{6.4}$$

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

$$\Gamma(\mu) = 1 - (1-\mu)^{N_a-1}, \tag{6.5}$$

due to the decoupling approximation [20]. The behavior of the system at equilibrium is governed

by the solution of the fixed-point equation:

$$\gamma = \Gamma(G(\gamma)) \tag{6.6}$$

The solution of this equation yields the collision probability, from which the attempt rate in the equilibrium μ can be determined from (6.4).

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

$$\begin{aligned} &\frac{L_i^a}{\theta(a,i,c)} = \frac{1}{N_a\mu(1-\mu)^{N_a-1}} + (T_0 - T_c) \\ &+\frac{1-(1-\mu)^{N_a}}{N_a\mu(1-\mu)^{N_a-1}}T_c + \frac{1}{N_a}\sum_{q=1}^{N_a}\frac{L_q^a}{R_q^{a,c}}x_q^a \end{aligned} \tag{6.7}$$

The first three terms in (6.7) 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, CTS, DATA and ACK frames may also get corrupted not only due to collisions but also due to channel errors. The work in [38] 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 equation (6.7) is experienced. The product of $1/(1 - e_{pt}^c)$ and the average per-packet delay of a client approaches the average delay until successful transmission of this client:

$$\begin{aligned} &\frac{L_i^a}{\theta(a,i,c)} \times \frac{1}{1-e_{pt}^c} = \\ &\left[\frac{1}{N_a\mu(1-\mu)^{N_a-1}} + (T_0 - T_c) + \frac{1-(1-\mu)^{N_a}}{N_a\mu(1-\mu)^{N_a-1}}T_c\right. \\ &\left.+\frac{1}{N_a}\sum_{q=1}^{N_a}\frac{L_q^a}{R_q^{a,c}}x_q^a\right] \times \frac{1}{1-e_{pt}^c} \end{aligned} \tag{6.8}$$

Comparing equations (6.2) and (6.8) we can clearly see that the average uplink *airtime cost* of AP a, is an approximation of the average per-packet delay expressed in (6.8). In (6.2), $O_{ca} + O_p$ is the delay due to channel contention and protocol overhead, and $\frac{1}{N_a} \sum_{i=1}^{N_a} \frac{B_t}{R_i^{a,c}}$ 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 [49], the long term average downlink saturated throughput for each client in cell a and for channel c, is:

$$\theta(a, c) = \frac{M_a}{\sum_{q=1}^{N_a} \frac{1}{R_q^{a,c}}}, \tag{6.9}$$

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:

$$\frac{L_i^a}{\theta(a, c)} = \frac{1}{M_a} \sum_{q=1}^{N_a} \frac{L_q^a}{R_q^{a,c}} \tag{6.10}$$

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 a STA to receive a packet successfully is:

$$\frac{L_i^a}{\theta(a, c)} \times \frac{1}{1 - e_{pt}^c} = \left[\frac{1}{M_a} \sum_{q=1}^{N_a} \frac{L_q^a}{R_q^{a,c}} \right] \times \frac{1}{1 - e_{pt}^c} \tag{6.11}$$

Finally, comparing equations (6.2) and (6.11) 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):

$$C_a^c = \overline{C_{a,c}^{up}} + \overline{C_{a,c}^{down}}, \tag{6.12}$$

and applies a channel allocation methodology where the channel with the minimum C_a^c is chosen. This also approximates the maximum thoughput 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 [71], [49]. 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.

6.2 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, i.e., 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 regards to the uplink channel conditions. This way the AP has knowledge with regards 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.

[Step1]. 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 [8]. In brief:

- 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),
- A_i stores the transmission rate $R^{A_i,c}$ for each downlink communication on channel c,
- A_i computes the airtime cost for each downlink communication on channel c,
- A_i computes the average downlink airtime cost on channel c.

[Step2]. 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 [8]:

- 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),
- STAs store their uplink transmission rates on channel c,
- STAs calculate their individual uplink airtime costs on channel c and inform the A_i,
- 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 [45]. 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.

[Step3]. 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 pre-defined threshold, T, then A_i remains in the same channel; otherwise it initiates a channel discovery process.

[Step4]. 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).

6.2.1 LAC Properties

We now discuss some properties of our LAC mechanism:

1. ***Band dwell duration***: Calculating $C_{A_i}^{c}$ 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. [58], [49], 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 re-scanned 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 the next section 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 de-associate 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 [49], [58], 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 [7].

3. ***Embedding the airtime cost value into control and data frames***: With LAC, clients piggyback 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 [45]. Note however that, as reported in [22], 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 [22]. Thus, we consider that the airtime cost value is repeatedly piggy-backed into every data packet transmission, for reliability.

6.3 Evaluating our Load-aware Channel Allocation Scheme

In this section, we evaluate our load-aware channel allocation scheme through extensive OPNET [72] simulations. We compare our scheme against the frequency selection approach, proposed in [49] 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.

6.3.1 Simulation set-up details

We have implemented LAC in OPNET [72], 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 1000m$\times$1000m 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, bi-directional 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:

- ***Single-channel assignment (SC)***: This is a very common approach, where the APs use a pre-selected 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.

- ***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.

- ***Gibbs-based Frequency Selection (GFS)***: We consider a scheme that is very similar to the one proposed in [49]; 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; i.e., packets are assumed to flow only from the APs towards their clients (please see [49] 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.

6.3.2 Simulation results and observations

We present our simulation experiments and the interpretations thereof, in what follows.

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). Fig. 6.1 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 *week* in a purely

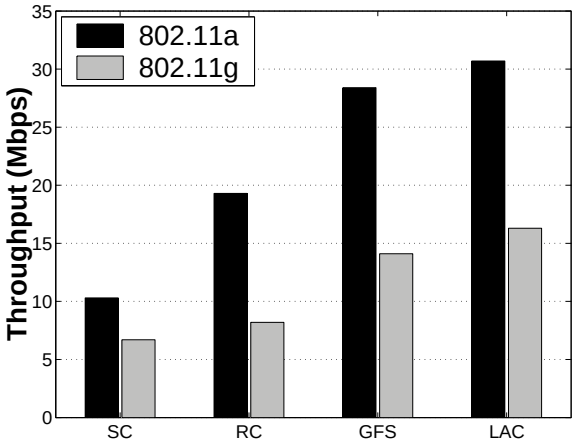


Figure 6.1: Total network throughput with saturated downlink UDP traffic.

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).

Applying bi-directional 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[2]. Hence, both uplink and downlink UDP traffic

[2]In our simulations, APs are connected through a wireline Ethernet network, and they use the Ethernet interface to exchange packets with other cells.

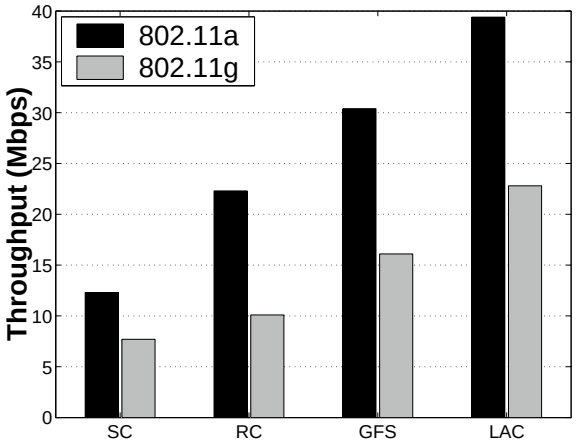


Figure 6.2: Total network throughput with both-directions UDP traffic.

takes place in every cell (uplink traffic in the client→AP (source) link and downlink traffic in the AP→client (destination) link). Fig. 6.2 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). Fig. 6.3 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,

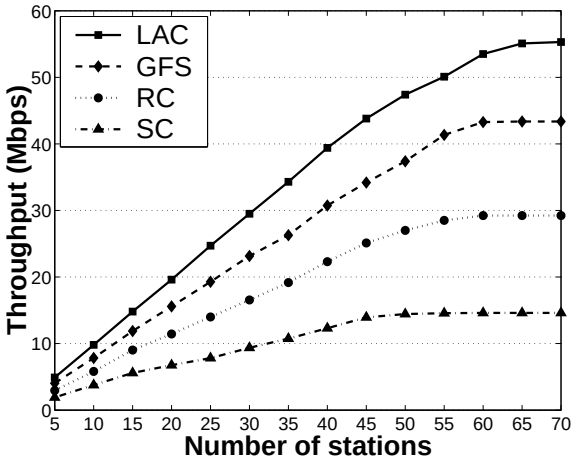


Figure 6.3: Total network throughput Vs. Number of clients.

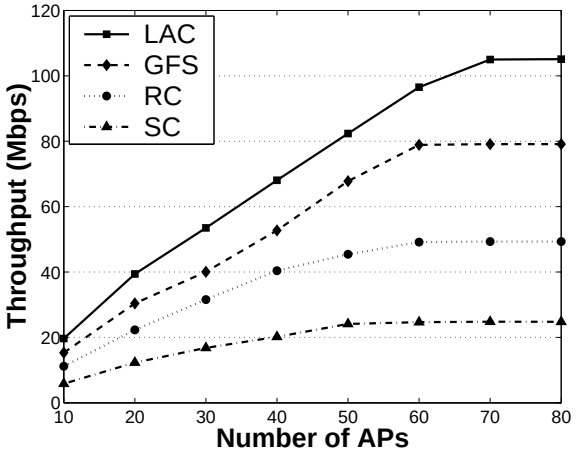


Figure 6.4: Total network throughput Vs. Number of APs.

from 10 to 80 (meanwhile, we deploy twice the number of clients: 10 APs - 20 clients, 20 APs - 40 clients, etc.). Fig. 6.4 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 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.

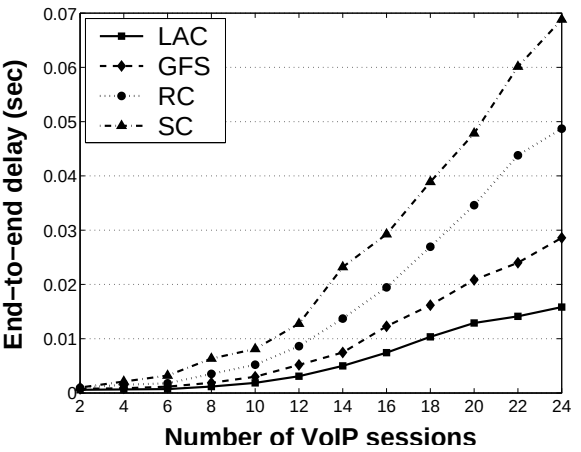


Figure 6.5: Average end-to-end delay with VoIP traffic.

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 set-up is the same as for the previous bi-directional experiments; i.e., VoIP traffic is exchanged among clients in our network – hence we again have both uplink and downlink traffic at every cell. Fig. 6.5 and 6.6 present the network performance with VoIP.

In particular, Fig. 6.5 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 enable to support highly loaded VoIP traffic (where both uplink and downlink traffic is introduced in the system). Finally, Fig. 6.6 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.

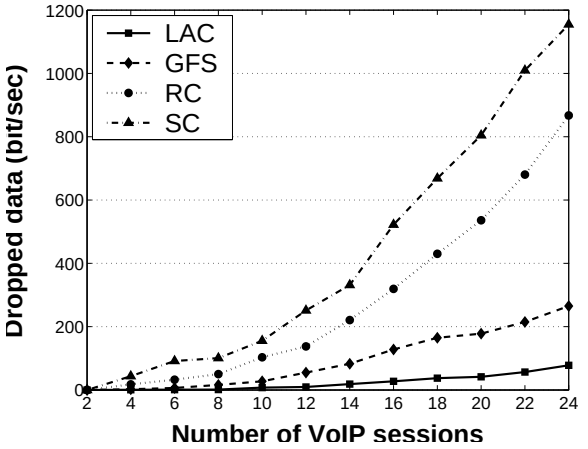


Figure 6.6: Average dropped data with VoIP traffic.

Table 6.1: Video traffic parameters

Parameter	Value
Coding format	QCIF(176x144)
Transmission interval	20 ms
Frame rate	25 frames/sec
Frame structure	IBBPBBPBBPBB
Mean frame size	0.77 KByte
Compression ratio YUV:MP4	49.46

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 a $60m$ movie: *Jurassic Park*, which is available in [60]. Then we have created a traffic profile in OPNET where a transmission interval of $20ms$ is introduced. Table 6.1 contains more information about the parameters of the video traffic that we use in our experiments. We keep the same simulation set-up 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.

Fig. 6.7 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

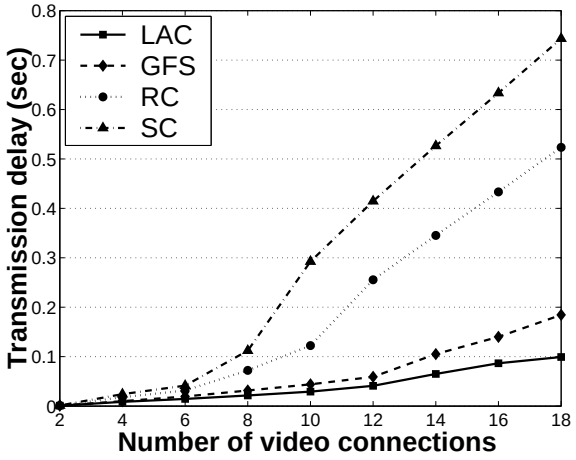


Figure 6.7: Average end-to-end delay with video traffic.

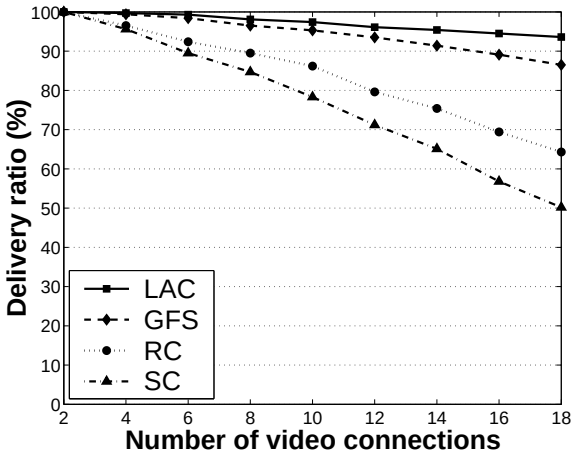


Figure 6.8: Average delivery ratio with video traffic.

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 Fig 6.7 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. Fig. 6.8 shows the average delivery ratio in the video data packet transmission and Fig. 6.9 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

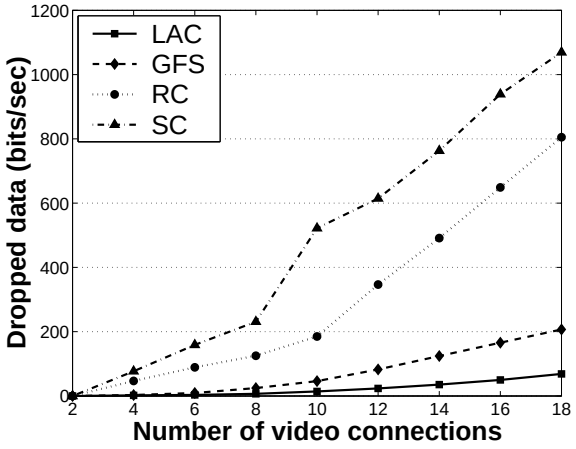


Figure 6.9: Average dropped data with video traffic.

than 8 or 10 video connections in the network and that LAC provides the best network performance.

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

In Fig. 6.10 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. Fig. 6.11 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

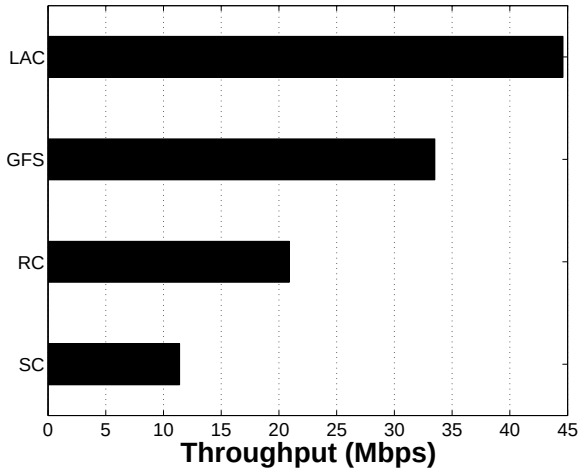


Figure 6.10: Total network throughput supporting different types of traffic.

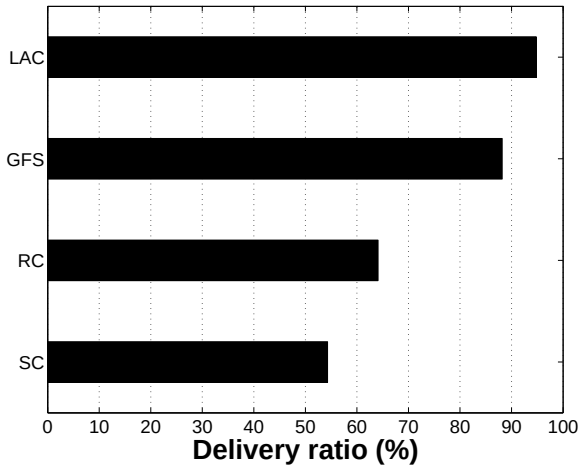


Figure 6.11: Average delivery ratio supporting different types of traffic.

average delivery ratio is average.

LAC convergence

In the previous section 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 bi-directional UDP traffic in the network.

In the first scenario we simulate a network with 20 APs and 40 STAs, all uniformly-randomly

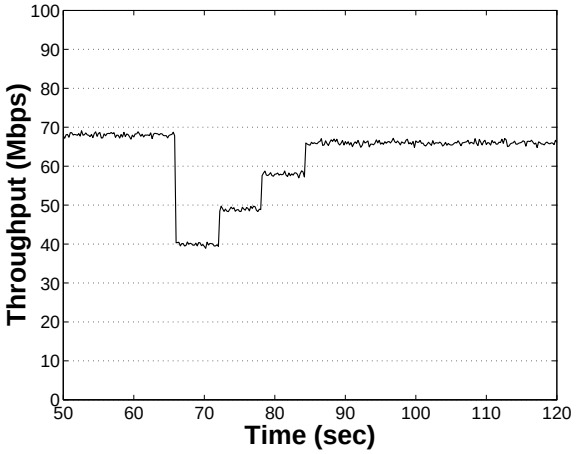


Figure 6.12: LAC convergence.

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 (for example we manually change the channel that the AP uses) in 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. Fig. 6.12 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 Fig. 6.12 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 re-converges). The time that passes till convergence in our simulation scenario is close to $20ms$.

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). Fig. 6.13 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

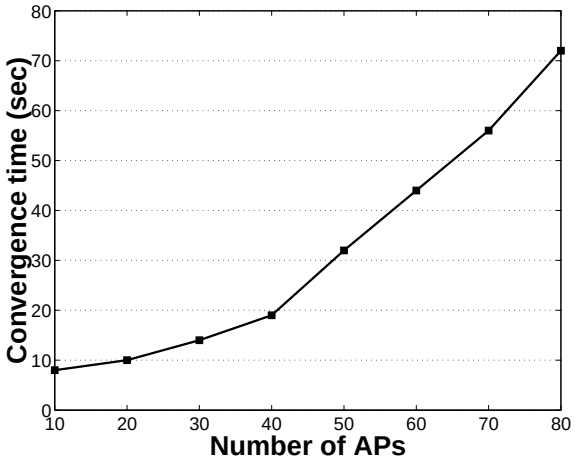


Figure 6.13: Convergence time Vs. Number of APs.

the APs that must execute LAC and share the available channels in 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), etc. 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

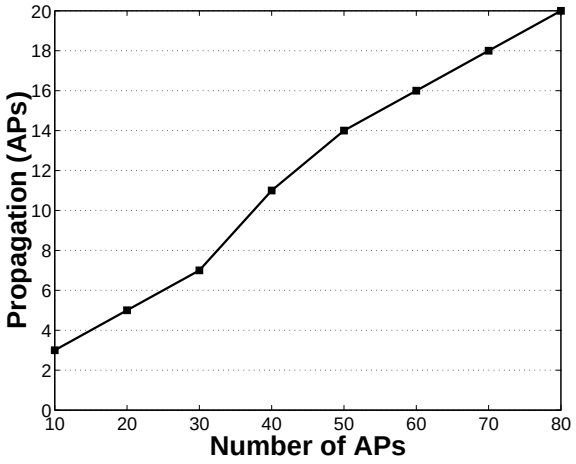


Figure 6.14: Propagation effect when the AP that "fires" the execution of LAC is placed at the center of the network.

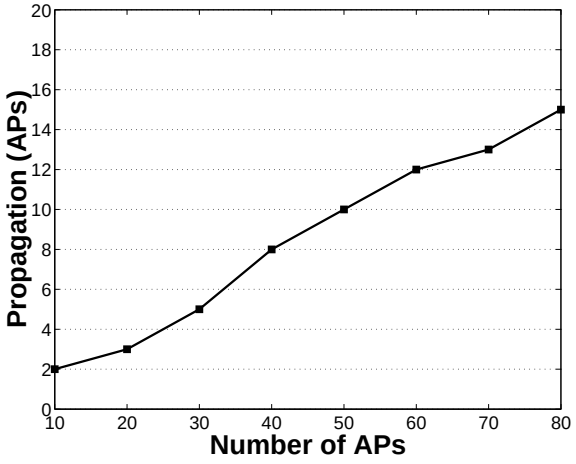


Figure 6.15: Propagation effect when the AP that "fires" the execution of LAC is placed at the edge of the network.

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 [7].

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 Fig. 6.14, 6.15, 6.16 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

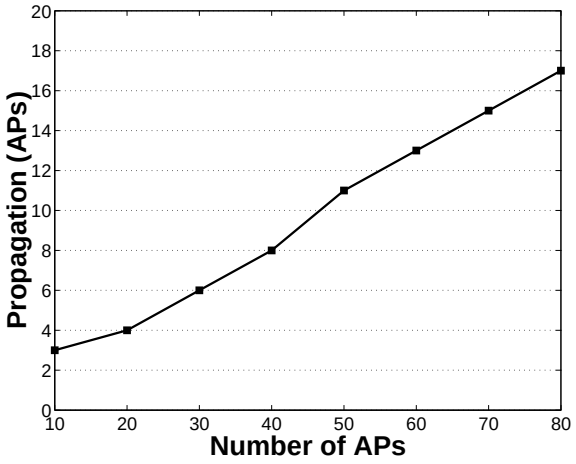


Figure 6.16: Propagation effect when the AP that "fires" the execution of LAC is placed at random of the network.

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.

Chapter 7

Routing-Aware Channel Selection in Multi-Radio Mesh Networks

In this chapter we present ARACHNE, a load and routing aware channel selection protocol for wireless mesh networks. ARACHNE performs end-to-end channel selection along a route, by adopting a variation of a load characterization metric [8]. In section 7.1 we present our preliminary experiments, which motivate the design of ARACHNE. In section 7.2, we present the design of ARACHNE for both the access and the backhaul levels. We evaluate our protocol through simulations, in section 7.3.

7.1 Motivating our Channel Allocation Policy

In this section, we describe a set of preliminary experiments on our wireless testbed, which motivate the design of ARACHNE. We first provide a description of our testbed platform, and subsequently we discuss our experiments and their interpretations. We also discuss relevant previous work.

Experimental set-up: Our wireless testbed deployment (Fig. 7.1) consists of 9 nodes that are based on the ORBIT hardware configuration, and run a Debian Linux distribution with kernel v2.6 over NFS. Each node is equipped with 1 GHz CPU, 512 Mbytes of memory, and a WN-CM9 wireless

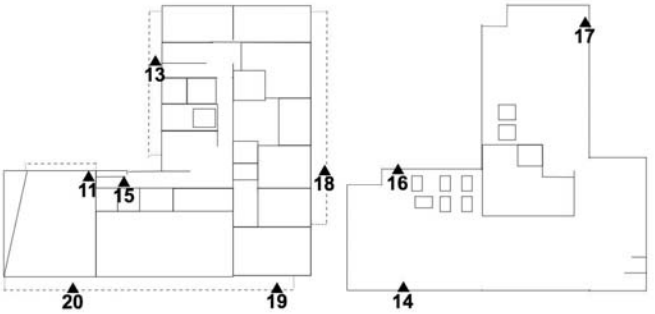


Figure 7.1: The testbed deployment in the 4th (left) and the 5th (right) floor of our building.

card, which carries the AR5212 Atheros chipset. We set the cards to 802.11g mode and we use channels 1, 6 and 11. For the purposes of these preliminary experiments we consider fixed routes between source and destination. We inject UDP traffic with various constant bit rates (CBR) and with packet size equal to 1500 bytes. For each end-to-end traffic session we set different application data rates; we utilize the *iperf* measurement tool.

Channel-load measurements: We provide a representative experiment in what follows. Consider the following two simultaneously active routes (see Fig. 7.1): **(a) 16**→15→20→**19**, and **(b) 13**→15→20→**14**. These routes have one link in common, 15→20, while all links are of similar quality in terms of achievable throughput in isolation. We apply two different channel selection policies; for both policies we make sure that connectivity is maintained between the end-hosts, *i.e., two consecutive relays use the same channel on one of their interfaces.* We first consider the channel policy, $A1$, which assigns channels according to the interference experienced (allocates the channel with the minimum aggregate interference, observed through RSSI measurements). Subsequently we assign channels to nodes, additionally taking into account the link loads, in terms of both link quality and aggregate traffic service; we call this policy $A2$. In other words, we manually prioritize the channel selection on the link 15→20; the rest of the links choose their channels as previously, *given the selection on link* 15→20. For both approaches, we measure the total end-to-end throughput for the two routes. Fig. 7.2 depicts these measurements for different source data rates.

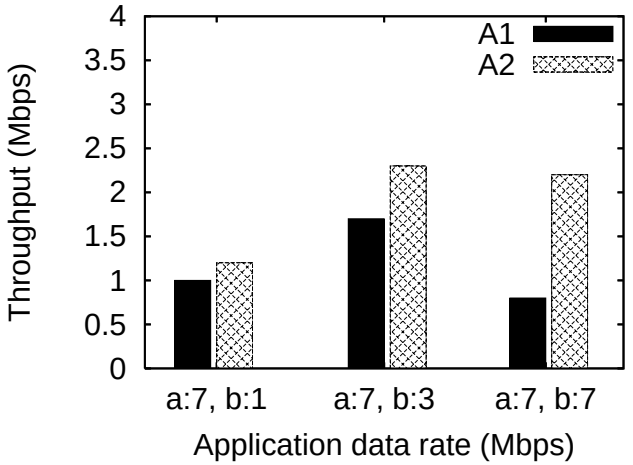


Figure 7.2: Policy A2 outperforms policy A1, especially at high loads.

First, we observe that the total end-to-end throughput is always higher with $A2$. Second, we see that the throughput improvement with $A2$ is much higher than with $A1$, when the data rates of the routes are of similar magnitude. This is because $A2$ assigns a "better" channel to the 15→20 link, which has to serve more traffic than the other links. With $A1$, however, this acts as a bottleneck to the performance of the individual routes: this bottleneck situation cannot be captured by simply measuring the interference at each channel. On the other hand, when one of the routes, (say (a)) has significantly lower traffic demands, the aggregate traffic that traverses link 15→20 comes mainly from the route (b) over a period of time; in such a case, $A1$ works quite well, too. This observation constitutes a key element for the design of ARACHNE, in terms of selecting channels and allocating the available interfaces to specific routes, as we explain in section 7.2. In what follows, we present the metric that our protocol uses during channel assignment.

Previous studies: Kauffmann et al. in [49] propose a distributed frequency selection algorithm that minimizes the global interference in the network. However, their algorithm considers WLANs only. Similarly, Rozner et al. in [81] propose a channel assignment scheme for WLANs, taking into account traffic demands. The MaxChop mechanism [69] provides high levels of fairness among users using channel hopping. However, it requires tight synchronization between AP and clients, while it is difficult to implement efficiently with off-the-shelf hardware. In [5] the authors study the

joint channel allocation and routing problem, assuming that traffic demands and network topology are known. They propose a centralized algorithm that maximizes the aggregate throughput. Raniwala et al. [79] propose a tree-based mesh architecture, Hyacinth, where the local channel load information exchange facilitates the channel selection. Hyacinth tries to address the joint problem of channel assignment and routing in wireless mesh networks. The latter two approaches, [5, 79], however, assume the availability of a global network view. Our protocol differentiates from these approaches by providing efficient end-to-end channel selection in a distributed manner (access level and mesh backhaul). In ARACHNE there is no need of synchronized channel access. Moreover our protocol does not employ any tree-based architecture, which in some cases cannot represent the actual network topology and its dynamics. Finally, our work is fully compliant with 802.11s wireless mesh networks and it can be implemented on top of the existing IEEE 802.11 standards.

7.2 Our Channel Selection Protocol

In this section, we present ARACHNE, our load and routing aware channel selection protocol. The design of ARACHNE makes the following assumptions: a) Nodes of the mesh backhaul are equipped with at least 3 radios (2 for the mesh backhaul communication and 1 for the access level), one of which is set to a pre-arranged channel, c_p, which is the same for all nodes in the network, b) A client associates to the AP with the strongest signal (RSSI) among all neighbor APs (802.11-based association procedure), c) The set of channels used at the access level (AP-client interfaces), is different from the channel-set used at the mesh backhaul. For simplicity in our work, the AP-client interfaces operate in the 2.4 Ghz band; the 5 GHz band is used exclusively for the interfaces in the mesh backhaul, and d) We assume that traffic is exchanged among end hosts belonging to the same mesh network, i.e., traffic is not crossing different networks.

The metric that is used in ARACHNE is called *airtime metric* and it is an approximation of the per packet latency (as shown in [7] and in our extended analysis [14]). The airtime metric was first discussed in the 802.11s [43] 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 end-to-end paths with the minimum total *airtime cost*. We adopt this metric in ARACHNE for the purposes of our proposed channel selection functionality.

Formally, the airtime metric of station i, that is associated with AP a which communicates using channel c, is given as:

$$C_{a,c}^{i} = \left[O_{ca} + O_p + \frac{B_t}{R_i^{a,c}} \right] \frac{1}{1 - e_{pt}^{c}}. \tag{7.1}$$

In (7.1), 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[1]. Some representative values for these constants, for 802.11g, are: $O_{ca} + O_p = 1.25$ms and $B_t = 8224$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 $R_i^{a,c}$.

Several studies have shown that the number of erroneously received packets increases and the transmission rate decreases when the cells in the network interfere [71], [49]. The *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. Besides, our analysis in [14] shows that the average *airtime cost* in the uplink and the downlink is an approximation of the average per-packet delay in both directions. Therefore, the average *airtime cost* is a representative metric that reflects the uplink and downlink channel performance and also approximates the maximum throughput in the cell.

ARACHNE captures the performance of an AP in terms of estimated throughput at a particular channel, by measuring the average *airtime cost* for both uplink and downlink, $C_a^c = \overline{C_{a,c}^{up}} + \overline{C_{a,c}^{down}}$ (*airtime cost* for AP a, in channel c), and applies a channel selection methodology where the channel with the minimum C_a^c is chosen. This channel selection policy determines the frequency with

[1]The transmission of test frames is necessary, in order to derive values for the computation of the airtime cost.

the minimum average per-packet delay in both uplink and downlink, thereby approximating the maximum throughput in the cell [14]. The goal of ARACHNE is to assign channels to mesh nodes, such that the average airtime metric is minimized, both at the access and the backhaul levels. Hence, ARACHNE involves two procedures, P1 and P2, one for each level. ARACHNE is executed exclusively by the nodes that belong to the mesh backhaul, i.e., the relay nodes as well as the APs that connect the end-hosts with the mesh network. The channel discovery is initiated by the AP of the source host. In what follows, we describe the operations of the protocol, P1 and P2, for each of the two levels of operation.

7.2.1 Channel Selection at the Access Level (P1)

The access level involves the communication of the end hosts (clients) with mesh nodes at the edge of the backhaul (APs). Let us assume that host A wishes to send traffic to another host B. This traffic will flow through A's AP, M_A, to the mesh backhaul and it will eventually reach M_B; the latter will finally forward the traffic to B. With ARACHNE, the frequency selection at the access level involves a channel discovery process at the two aforementioned APs, in order to find the channel with the minimum airtime cost value (we provide the steps of this process below). The calculation of this value is performed for every scanned channel, and involves all the downward and upward links of M_A, M_B. Let us again assume that host A wants to send traffic to host B. This part of the protocol, P1, is executed by the APs at the access level, and consists of the following steps.

[P1a]. Deriving interference information for the current channel. At the nominal start of ARACHNE, M_A is informed about the operational frequencies of its neighbor *co-channel* APs. This can be performed through passive scanning of periodically transmitted beacons [22]. By the end of this step, M_A is aware of the total received power from all co-channel APs.

[P1b]. Computing the downlink airtime cost. M_A calculates the aggregated downlink airtime cost with its clients, through a link performance-measurement procedure, described in detail in [8] where airtime metric is used for user association.

[P1c]. Computing the uplink airtime cost. The clients of M_A calculate their individual uplink costs. They piggy-back this information through their data frame transmissions towards M_A. By the end of this step, M_A has received information regarding the uplink channel qualities from all its clients.

[P1d]. Deciding if the current channel is appropriate. M_A receives the information from the client and computes the average airtime cost (for both uplink and downlink) for the access level. If this is higher than a pre-defined threshold T, M_A remains in the same channel; otherwise it initiates a channel discovery process. The value of threshold T is decided and controlled by the system designer.

[P1e]. Computing the cumulative airtime cost at the next available channel. M_A and its clients switch to the next channel and repeat steps $P1a - P1e$.[2] If all available channels have been visited, M_A finally selects the channel with the minimum average airtime cost (for both uplink and downlink).

More information about the calculation procedure, the threshold T value (which determines the frequency of invocation of the above procedure) and the convergence duration of ARACHNE can be obtained in [14]. Note here that process P1 (of selecting a channel at the access level) is actually independent of P2; although they are executed in parallel, they do not affect each other to a large extent, in terms of convergence time, since they utilize different sets of channels.

7.2.2 Channel Selection at the Mesh Backhaul (P2)

The role of the mesh backhaul is to serve the forwarding of packets towards their final destinations. Undoubtedly, the network connectivity is affected by a potential channel selection policy that may be applied in the mesh backhaul. ARACHNE's channel selection framework ensures connectivity at all times in the network, while the selection of a frequency at a particular link takes into account the load of the routing sessions that are traversing the link.

[2]The measurements on our testbed indicate that this channel switching of the AP and its clients is performed quite quickly.

Load-aware routing with RM-AODV: For the purposes of this study, we assume that the RM-AODV routing protocol [43] is applied on the mesh backhaul. RM-AODV is the default routing protocol proposed in the context of 802.11s mesh networks, where the airtime cost is used as a routing decision metric in the mesh backhaul. In particular, the airtime metric is appended to the RREQ and RREP messages, during the route discovery process; finally, the end-to-end path with the minimum total airtime cost (the route that has the minimum load) is selected. Our choice of RM-AODV is motivated by the fact that this routing protocol is based on the airtime metric. *Note, however, that ARACHNE can operate in conjunction with **any** load-aware routing protocol!*

Control information exchange using LABA: ARACHNE employs the Local Association Base Advertisement (LABA) mechanism introduced in 802.11s [43], in order to disseminate information with regards to inform the entire mesh network about the clients (end-hosts) that are associated with each mesh AP (MAP). MAPs periodically broadcast LABA frames, which consist of the MAC addresses of the hosts that are associated with. We have enhanced the LABA frames to include load related information, as we explain later in the description of the main parts of our protocol.

Channel assignment preliminaries: Our protocol prioritizes the channel assignment on the most loaded mesh APs, as well as on the mesh APs that are expected to be highly loaded in the near future. In particular, ARACHNE observes the per-link load, for the links that serve one or more routes in the mesh; the load is captured in the airtime cost metric [14]. The main problem in designing distributed channel selection policies is the channel dependencies that arise between the nodes that are part of the mesh network. For example, we assume that in the simple network in Fig. 1.2 nodes are equipped with 2 wireless interfaces. Node D uses channel *a* in order to communicate with node E and node E uses channel *b* to communicate with nodes F and G. In case that in the link E-F the current channel is heavily loaded, node E uses a new channel *c* that operates better at the link E-F. As E has only two interfaces, channel *c* must be used at the link E-G too. Unfortunately, a strong dependency between links E-F and E-G is established. Channel *c* may not operate effectively at the link E-G and therefore the performance of the network is decreased. The aforementioned *ripple effect* could be further propagated in dense/huge mesh networks. In order to avoid *ripple*

effects in the channel selection process [79], we categorize the wireless interfaces at each mesh AP that serves as a relay as:

- ***IN** network interfaces* that are used for data reception,
- ***OUT** network interfaces* that are used for data forwarding.

In addition, each node can assign channels only to its ***OUT*** interfaces. The ***IN*** interfaces follow the channels that are assigned by the nodes that communicate with the current node. In our protocol we assume that each node is equipped with at least one ***IN*** and one ***OUT*** interface.

In what follows, we describe the steps that are executed by ARACHNE, for assigning channels at the mesh backhaul (P2). To begin with, we consider a network state, in which: (a) All nodes have already dedicated one of their ***OUT*** interfaces (I_p), to control channel c_p; (b) The RM-AODV protocol has converged to a set of routes between end-hosts; (c) All of the other wireless interfaces of a mesh AP (besides I_p) have been randomly assigned a channel in the 5 GHz band. The channel assignment in P2 is comprised of the following steps.

[P2a]. Constructing a priority list. With ARACHNE, an AP starts the channel-scanning process at a time-instant dictated by its priority ranking. This priority is highly-related to the load of each mesh AP; the higher the load (the data that must forward), the higher the priority. In addition, we believe that in this prioritization the estimated load of a mesh AP in near future must be taken into account. In this way we guarantee that our protocol will converge to a long-lasting and a stable channel allocation. Hence, it is imperative that this list is constructed, before APs start scanning each channel. Each mesh AP a calculates its priority rank according to its current load and its expected load, as:

$$P_r^a = L_{crnt}^a w_1 + L_{est}^a w_2,$$

where L_{crnt}^a is the current load, L_{est}^a is the estimated load and w_1, w_2 are the weights that are used in the calculation (we will give more details about the selection of w_1, w_2 in the evaluation of our protocol).

As far as the calculation of the estimated load L^a_{est} is concerned, we adopt the estimation method (based on historical data), proposed in [27]. In this approach the authors design a trace-based traffic model in order to predict the aggregated traffic demands of an AP in near future. Time-series analysis is the basis of this traffic estimation model. The accuracy of this model is high and in combination with our load-aware channel selection protocol we achieve long-lasting and stable channel allocation in the mesh network.

Besides, the priority ranks are incorporated into the LABA frames and are broadcasted in the network. Consequently, a sorted list (in terms of channel selection priority rank) is disseminated and maintained by all APs. Hence, by the end of this step each AP knows the priority of all APs in the network.

[P2b]. Performing channel-scanning. Let us assume that each AP can scan a set of K channels. The first mesh AP in the priority list measures the cumulative airtime cost for each of the K channels, and constructs a local, *temporary*, sorted list with the cumulative airtime cost, per channel.

[P2c]. Assigning route sessions to available *OUT* interfaces. Each mesh AP typically prefers to assign low-airtime channels to links that serve high-load end-to-end traffic sessions. ARACHNE manages to efficiently utilize the available spectrum, by providing a balanced channel and interface assignment in the network. A mesh AP a calculates the maximum load L^a_{sh} that can be assigned at each of its OUT interfaces, as (clearly, when the number of available interfaces is higher than the number of served sessions, ARACHNE allocates an interface to a particular session): $L^a_{sh} = L^a_{crnt}/N^a_{OUT}$, where N^a_{OUT} is the number of the available OUT interfaces of mesh AP a, and L^a_{crnt} is the current load that must be served by the OUT interfaces. The main constrains of the interface-assignment strategy are: (a) The load assigned to an OUT interface is less than L^a_{sh}, and b) The load is proportionally allocated to the available OUT interfaces in terms of the load of the flows that pass through the current mesh AP. In other words, the load is balanced to the available OUT interfaces.

[P2d]. Assigning channels to *OUT* interfaces. In order to assign a channel to an OUT interface, a coordination with the IN interface of the mesh AP that receives the traffic is required. In particular,

a mesh AP *A* that selected the channel with the minimum airtime cost in the communication with a mesh AP *B*, sends an RTC frame to *B* announcing in this way that a new channel must be used in their communication. Then, *B* responds with a CTC frame and sets one of its *IN* interfaces to the selected channel. In case that *B* hasn't responded in a time window (timeoff), *A* retransmits its RTC frame. We must note here that there are special situations where *B* has recently assigned channels to its *IN* interfaces (during the same protocol iteration) and there are no available interfaces to assign the channel proposed by *A*. In other words, higher priority APs (compared to the priority of *A*) have previously assigned those channels to *Bs' IN* interfaces and there are no available *IN* interfaces to use. In this case *B* responds with an XTC frame announcing this situation to *A* and the available channels that are assigned to its *IN* interfaces. Lastly, *A* is restricted to use one of those channels in the communication with B (the channel with the minimum airtime cost in the link A-B). After the completion of the aforementioned process, the AP sends a "good-to-go" unicast[3] message to the next AP in the list, to initiate its channel scanning and assignment process.

[P2e]. Selecting channels iteratively. When all APs have completed a round of channel scanning, RM-AODV updates the routing tables at the mesh backhaul, taking into account the new state of the network. Note that RM-AODV discovers routes based on the computation of the airtime cost, which will have likely changed after an iteration of the channel selection process. Steps P2a-P2e are repeated until the channel selection has converged.

7.3 Evaluating ARACHNE

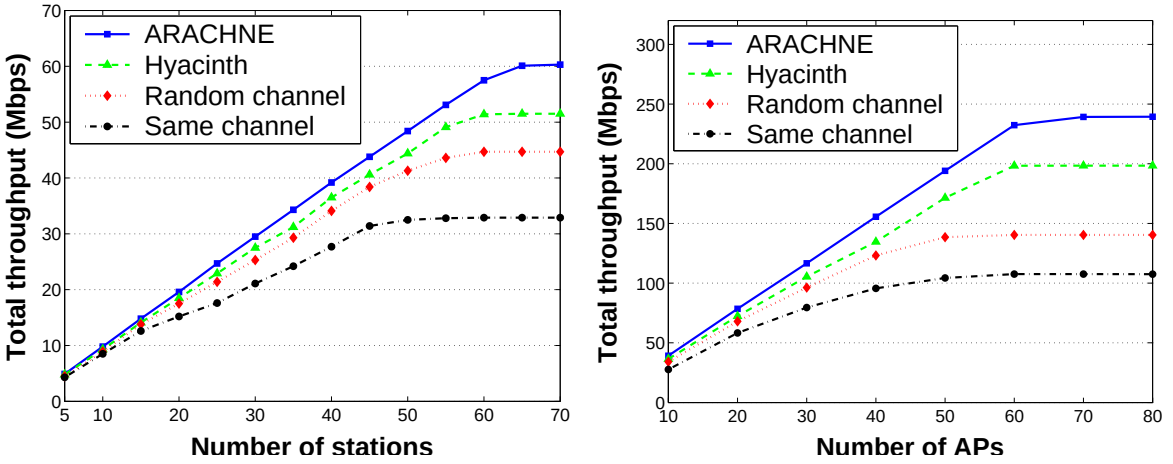
In this section, we evaluate ARACHNE through extensive simulations, which import both synthetic [72] and real traces [28], [41]. We compare our protocol against other potential channel selection schemes, and we present ARACHNE's predominance in terms of the total network throughput, average packet dropping and average transmission delay.

Simulation set-up: We have implemented ARACHNE in OPNET [72]. We have also implemented

[3]This unicast message speeds up the process of completing a full iteration of channel scanning for all participating mesh APs.

the RM-AODV protocol and we consider this routing protocol in our simulations. The clients are uniformly distributed (at random) in the 1000m$\times$1000m simulation area. All nodes use a default transmit power of 20 dBm and the source-destination pairs are randomly chosen in the network. We experiment with: (a) fully-saturated, end-to-end UDP traffic, (b) VoIP traffic and (c) real traffic traces from Dartmouth College [28] and IBM [41]. The weight values that are used in our simulation experiments are: $w_1 = 0.6$ and $w_2 = 0.4$ (the current load affects the channel selection procedure more than the estimated load in near future). As far as the duration of the measurement period is concerned, in our experiments the APs remain $100ms$ in each scanned channel in order to gather the necessary information. The convergence of our mechanisms is reached rather quickly, in approximately 2-3 iterations in our network topologies (close to $20sec$ in average). As we have mentioned the traffic keeps flowing during the execution of ARACHNE and the network operation is not affected. We chose accordingly the values of the performance threshold T in order to avoid unnecessary and frequent protocol executions. Throughout our simulation experiments we compare the network performance with ARACHNE, against the single-channel assignment, a random-channel allocation strategy, as well as the Hyacinth protocol [79].

Simulations with saturated UDP traffic and VoIP: To begin with, we opt to observe the variation in the total network throughput, as we increase the number of source-destination pairs. For this, we progressively increase the number of associated clients from 5 to 70. Here the network consists of 10 mesh APs, each of which is equipped with 2 wireless interfaces for the backhaul communication. We observe in Fig. 7.3(a) that the performance with ARACHNE is similar to the other 3 policies, when the number of clients (and therefore the communication load) is low. However, with increased load ARACHNE manages to provide much higher end-to-end throughputs (up to **85**% difference!), due to its efficient end-to-end channel selection strategy. Hyacinth is designed especially for wireless internet traffic which is directed to/from the wireless gateways. In our simulation scenario where saturated UDP traffic is supported, the tree-based architecture is incapable to support dynamic traffic variations in the network (especially in high communication load). In low load conditions Hyacinth performs close to ARACHNE. Contrarily, when the number of clients in



(a) Total throughput Vs. # clients.

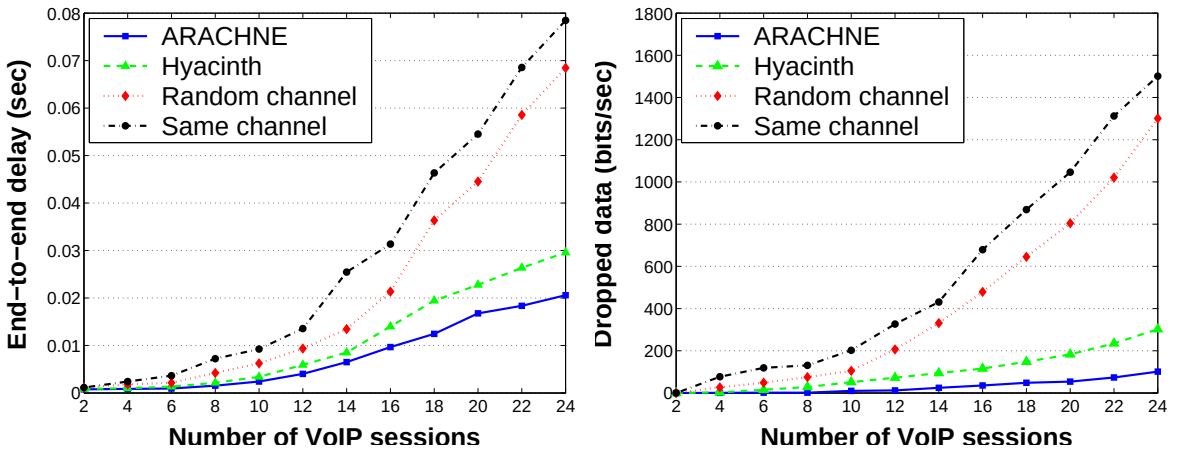
(b) Total throughput Vs. # APs.

Figure 7.3: ARACHNE povides very high total end-to-end throughput with saturated UDP traffic.

the networks increases the performance drops and the throughput saturation point is reached quite quickly.

Next, we examine the scalability of our protocol. We measure the total end-to-end network throughput as we increase the number of APs (i.e., the mesh density) from 10 to 80 (we increase the number of clients along with the mesh APs: 10 APs - 40 clients, 20 APs - 80 clients, etc.). Note that the interference is dynamically changing while the number of the APs in the network increases; the airtime cost metric manages to effectively capture the varying co-channel interference. Fig. 7.3(b) depicts the network performance in terms of total network throughput. We observe that ARACHNE is able to adapt to topology and load variations, and therefore it manages to provide the most beneficial selection of the available channels at the mesh backhaul. The tree-based architecture that is introduced by Hyacinth does not scale well and therefore the achieved performance is worst.

In order to observe the performance of our protocol with delay-sensitive data exchange, we utilize varying, parallel, end-to-end VoIP traffic sessions. Fig. 7.4(a) depicts the average end-to-end delay of VoIP packet transmissions. We observe that ARACHNE achieves low end-to-end delays, since it considers the load of the individual links across a route in the process of assigning frequencies to wireless interfaces. Recall here that ARACHNE is expected to provide high benefits when operating in conjunction with load-aware routing protocols, such as RM-AODV (this is case with



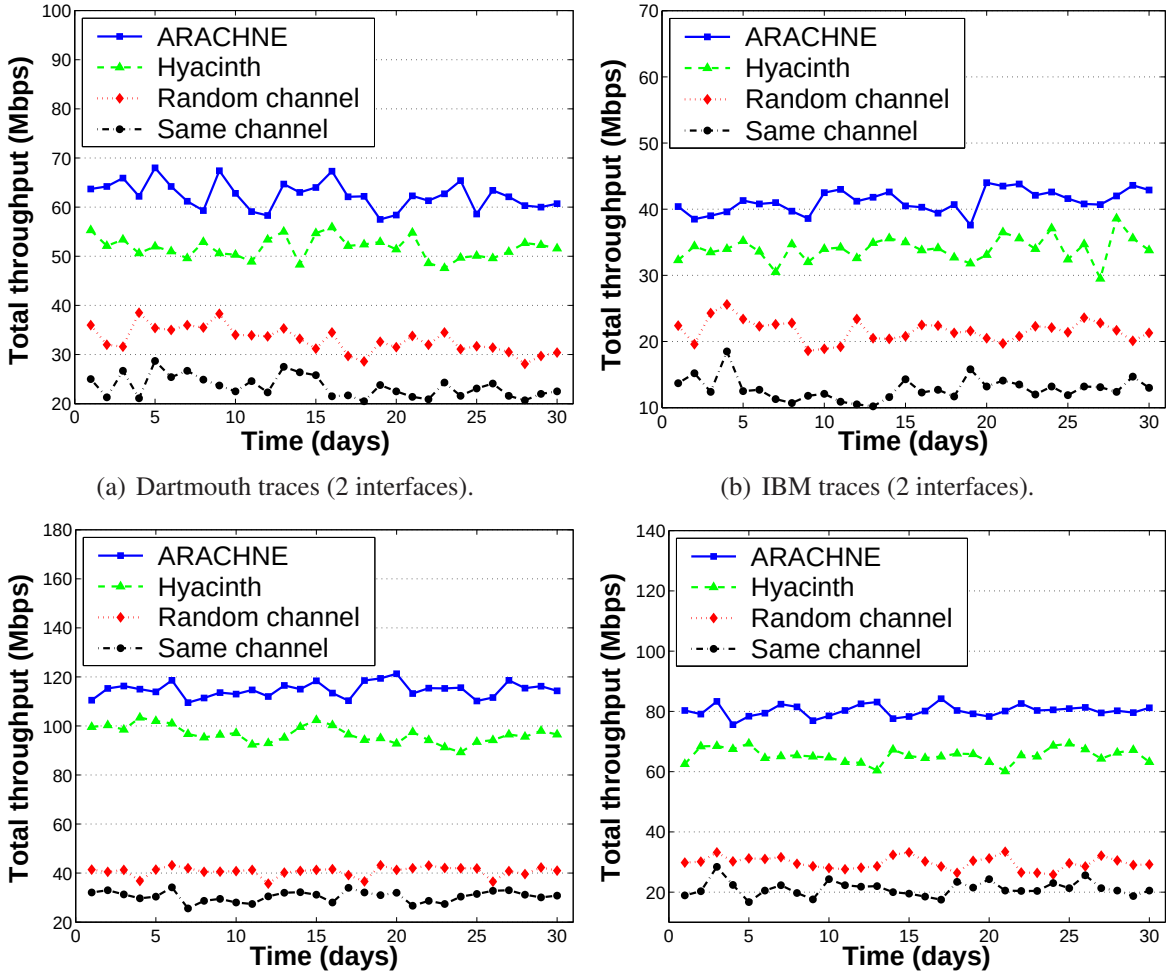
(a) Average end-to-end delay.

(b) Average dropped data.

Figure 7.4: VoIP simulation results.

our simulations), since they both take into consideration the load that is experienced by individual links. Furthermore, Fig. 7.4(b) shows the average number of dropped data packets due to channel errors and contention. The performance of ARACHNE is impressive, since packet dropping is kept in very low levels, as compared to Hyacinth and to the other strategies. Hyacinth does not consider the channel allocation at the access level and therefore, it can support less VoIP sessions than ARACHNE can (which provides end-to-end channel selection and therefore the provided QoS to the VoIP clients is higher).

Simulation results with real traces: Next, we present our simulation experiments with real traces from Dartmouth College and IBM. This will reveal the behavior of ARACHNE in more realistic scenarios. By analyzing the SNMP logs from each AP, we derive the dynamic behavior of the aggregated traffic demand. Note that these traces have been gathered from real wireless networks settings. Therefore, they represent actual traffic demands and behaviors of the APs and clients in a real network. The traces from Dartmouth College [28] (3/2001) involve the buildings labeled as *AcadBldg10, SocBldg4*, and include approximately 40 APs (the traces contain the AP location; we have placed them accordingly in our simulation space). In addition, the traces from IBM [41] (8/2002) have been gathered from the buildings labeled as *MBldg, and SBldg*. The number of the APs that are placed in these buildings varies (the traces don't contain their locations and therefore



(a) Dartmouth traces (2 interfaces).

(b) IBM traces (2 interfaces).

(c) Dartmouth traces (4 interfaces).

(d) IBM traces (4 interfaces).

Figure 7.5: Simulation results with real traces: ARACHNE is predominant!

we have simulated various random topologies with them). Note that the number of active clients (end-hosts) varies in both trace sets and they are uniformly distributed in our experiments (the traces do not contain information about client locations). In order to import the traces in our simulation we periodically compute the average rate for each client, once every 3 minutes, and we correspond the computed average values to the traffic demands of each client.

To begin with, we consider 2 wireless interfaces at each AP for the backhaul communication. Fig. 7.5(a) and 7.5(b) show that ARACHNE achieves the highest total network throughput in all cases. Our simulation experiments reveal that channel selection at the access level plays an important role in the overall end-to-end performance. Being based on the airtime cost metric, ARACHNE

takes into account the channel conditions as well as the communication load in both access level and the mesh backhaul (not considered by the other approaches, like Hyacinth) it manages to boost the network throughput. Furthermore, Fig. 7.5(c) and 7.5(d) depict the performance of the compared strategies when the mesh APs are equipped with 4 wireless interfaces. The results are similar as above.

Chapter 8

Conclusions

In this last chapter we summarize the contribution of the current dissertation. Besides, we pave the way for future research actions in the research field that could exploit the capabilities of the 802.11 wireless technology.

Firstly, we summarize and briefly describe the main components of the proposed *Resource Management Framework* in the thesis:

- ***A cross-layer association control framework for wireless mesh networks.*** In this framework we have introduced two association schemes that can be applied to WLANs and wireless mesh networks. The first scheme (channel-quality based) is fully compliant with 802.11h standard and allows STAs to associate with the best AP by taking into consideration the quality of the channel on both the uplink and the downlink. The second scheme (airtime-metric based) takes under consideration the local cell information about the channel conditions in both the uplink and downlink as well as the load of each cell. Extending the second scheme we go one step further, combining this information with the information about the routing of the packets from the candidate AP to the destination. Lastly, inspired by the previous schemes we have designed a hybrid association scheme that utilizes dynamic mode changing according to the communication environment. Bellow we summarize our work and we appose the main contributions in this work:

- A new association control framework that can be applied to WLANs and wireless mesh networks (compliant to 802.11s).
- A simple-to-implement channel-quality based association/handover mechanism, which utilizes the new mechanisms included in 802.11h.
- A novel association/handover mechanism that use the notion of the airtime cost in making association decisions. This new scheme, considering dynamically the channel and candidate cells condition, adapts the association procedure optimizing the throughput of the network.
- A cross-layer association/handover scheme for wireless mesh networks that combines information about the links between the STAs and the candidate APs as well as the information about the routing of the packets from those APs to the destination.
- A hybrid association scheme that combines the previous associations schemes in order to optimize the association procedure according to the operational conditions of the network.
- Realistic simulation scenarios and testbed experiments where we study not only the throughput of the wireless cells in a mesh network but also the end-to-end performance of the network.

- ***A handoff framework that introduces cooperation between STAs/APs.*** The proposed cooperative framework is compliant to 802.11k and it utilizes information exchange and measurement mechanisms that are specified in the standard in order to eliminate the scanning/probe delays in the handoff process. The mechanisms work independently of the underling association/handoff procedures and therefore they can be applied in combination to any association/handoff protocol. Besides, the proposed mechanisms in this framework can be applied to 802.11-based WLANs and wireless mesh networks. Our main contributions in the current research field are:
  - A 802.11k compliant cooperative handoff framework for wireless networks.

  - Two cooperative schemes that take full advantage of the mechanisms that 802.11k provides.
  - Extensive simulation experiments where we support QoS sensitive applications. We evaluate the performance of our framework by applying different underlying handoff decision protocols and we measure the performance improvement that is achieved.

- ***A study for aggregating the ADSL bandwidth via 802.11 wireless connectivity via TDMA (multiple associations).*** In this work we have studied the problem of aggregating the ADSL bandwidth via 802.11 wireless connections, both via extensive experiments and via a simulator that accurately correlates the TCP RTT with the time spent by the station on each AP. We have presented a model that accurately predicts the effect that such TDMA schemes have on the observed RTT of TCP, and we have introduced a resource allocation algorithm that improves the aggregated throughput with a complexity that grows linearly with the number of APs. Our solution does not require modifications to the rest of the network, and operates locally at the station. Furthermore we showed that its performance is very close to the theoretical upper bound for a number of scenarios.

- ***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 compared LAC against 3 other channel selection approaches and we showed 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 is 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.

- ***ARACHNE, a routing-aware channel selection mechanism for multi-radio mesh networks.*** ARACHNE adopts a metric that provides an estimation of the average packet transmission delay in a communication channel. ARACHNE captures the interference effects in the networks and applies an end-to-end channel selection policy that manages to provide approximately the maximum end-to-end network throughput. Our work is compared against 3 other channel selection approaches and we show that it outperforms all of them, for different traffic scenarios and network densities.

An important part of the dissertation was the extensive study of several new 802-11-based standards. Some of the mechanisms that are defined in these standards were applied in our framework in order to achieve compatibility and efficient execution. We briefly describe the standards that we have studied in this thesis:

- **IEEE 802.11h**: IEEE 802.11h was originally developed to extend the 802.11 operation in the 5 GHz band in Europe. In order to co-exist with the primary users in the 5 GHz band in Europe - the radar and satellite systems, the 5 GHz WLAN devices are required to support DFS (Dynamic Frequency Selection) and TPC (Transmit Power Control). For example, the WLAN devices are required to switch its operational frequency channel to another channel once a radar signal is detected in the current channel. On the other hand, when a satellite signal is detected, the WLAN devices are allowed to use the transmit power up to the regulatory maximum minus 3 dB while normally they can transmit at up to the regulatory maximum level. The 802.11h defines the DFS and TPC mechanisms on top of the 802.11 MAC and the 802.11a PHY for these purposes. Note that, even though the 802.11h has been developed to satisfy the European regulatory requirements, it can be apparently used in other countries for multiple purposes, such as automatic frequency planning, reduction of energy consumption, range control, reduction of interference, and QoS (Quality of Service) enhancement.

- **IEEE 802.11s**: IEEE 802.11s is the IEEE 802.11 standard for ESS Mesh Networking. It specifies an extension to the IEEE 802.11 MAC to solve the interoperability problem by

defining an architecture and protocol that support both broadcast/multicast and unicast delivery using ”radio-aware metrics over self-configuring multi-hop topologies.

The 802.11s Task Group is working on an infrastructure mesh amendment to allow 802.11 access points or cells from multiple manufacturers to self-configure into multi-hop wireless topologies. We expect that a mesh standard would enlarge the range of markets and applications for the 802.11 standard. Example usage scenarios for mesh networks include interconnectivity for devices in the digital home, unwired campuses, and community area networks or hotzones. The standard is expected to be designed to be extensible by manufacturers to enable diverse usage scenarios with differing functional requirements. For example, some applications may require quick ad-hoc setup and teardown of a mesh while others require large scale and maximum throughput.

- **IEEE 802.11k**: IEEE 802.11k is focused on standardizing the radio measurements that will allow uniform measurement of radio information across different manufacturer platforms. By having standardized, repeatable measurements, system designers can utilize radio environment information to make better decisions as to frequency use, transmit power levels, etc. This will lead to 802.11 networks that are easier to monitor and manage and that can make more efficient use of the available spectrum.

  802.11k is intended to improve the way traffic is distributed within a network. In a wireless LAN, each device normally connects to the access point (AP) that provides the strongest signal. Depending on the number and geographic locations of the subscribers, this arrangement can sometimes lead to excessive demand on one AP and underutilization of others, resulting in degradation of overall network performance. In a network conforming to 802.11k, if the AP having the strongest signal is loaded to its full capacity, a wireless device is connected to one of the underutilized APs. Even though the signal may be weaker, the overall throughput is greater because more efficient use is made of the network resources.

The knowledge that we obtained from our study was the cornerstone in the design of the *Re-*

source Management Framework. In other words, the new standards provide functionalities that can be used by several mechanisms that are applied at each layer. In our try to be fully compliant to the aforementioned standards (and of course to IEEE 802.11) we based the design of the proposed schemes on the core functionalities that these standards offer. In addition, we combined and extended the operation of the standards in several parts of our framework highlighting in this way that these standards must co-exist and co-operate in a common framework. This new concept could pave the way for a new set of wireless standards.

Taking into account the outcome of our research approaches, we conclude our discussion by "manifesting" the basic directions that must be followed (basically at the MAC/Routing layers) by the future actions in the current research field in WLANs and wireless mesh networks:

- **Scalability**: The scalability issue in multi-hop wireless networks has not been fully solved yet. Most existing MAC protocols only solve partial problems of the overall issue, but raise other problems. To make the MAC protocol really scalable, new distributed and collaborative schemes must be proposed to ensure that network performance (e.g., throughput and even QoS parameters such as delay and delay jitter) will not degrade as network size increases. It is obvious that a multi-channel MAC protocol can achieve higher throughput than a single-channel MAC. However, to really achieve spectrum efficiency and improve the per-channel throughput, the scalable MAC protocol needs to consider the overall performance improvement in multiple channels. Thus, developing a scalable multi-channel MAC is a more challenging task than a single-channel MAC.

- **MAC/Physical Cross-Layer Design**: When advanced physical layer techniques, such as MIMO and cognitive radios, are used, novel MAC protocols, especially multi-channel MAC, need to be proposed to utilize the agility provided by the physical layer.

- **Network Integration in the MAC Layer**: The wireless routers are responsible for the integration of various wireless technologies since they are vital components in a deployed wireless network. Thus, advanced bridging functions must be developed in the MAC layer so that

different wireless radios, such as IEEE 802.11, 802.16, 802.15, etc., can seamlessly work together. Reconfigurable/software radios and the related radio resource management schemes may be the ultimate solution to these bridging functions.

- **Routing/MAC Cross-Layer Design**: A routing protocol needs to interact with the MAC layer in order to improve its performance. Adopting multiple performance metrics from layer-2 into routing protocols is an example. However, interaction between MAC and routing layers is so close that merely exchanging parameters between them is not adequate. Merging certain functions of MAC and routing protocols is a promising approach.

Bibliography

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, “Link-level Measurements from an 802.11b Mesh Network”, ACM Sigcomm, Portland, USA, Aug. 2004.

[2] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, “Self-Management in Chaotic Wireless Deployments”. In ACM MOBICOM, 2005.

[3] I. F. Akyildiz, X. Wang, and W. Wang, “Wireless mesh networks: a survey”, Computer Networks 47 (2005) 445487.

[4] Y. Amir, C. Danilov, M. Hilsdale, R. Musaloiu-Elefteri, and N. Rivera, “Fast Handoff for Seamless Wireless Mesh Networks”, in Mobisys 2006

[5] M. Alicherry, R. Bhatia, and L. Li, “Joint Channel Assignment and Routing for Through- put Optimization in Multi-radio Wireless Mesh Networks”, In ACM MOBICOM, 2005.

[6] W. Arbaugh, A. Mishra, and M. Shin, “An empirical analysis of the IEEE 802.11 MAC layer handoff process”, ACM SIGCOMM Computer Communication Review, 33, 2003.

[7] G. Athanasiou, T. Korakis, O. Ercetin, and L. Tassiulas, “A Cross-Layer Framework for Association Control in Wireless Mesh Networks”, In IEEE Transactions on Mobile Computing, vol. 8, no. 1, pp. 65-80, January 2009.

[8] G. Athanasiou, T. Korakis, O. Ercetin and L. Tassiulas, “Dynamic Cross-Layer Association in 802.11-based Mesh Networks”, in IEEE INFOCOM 2007, Anhcorage, Alaska, USA, May 2007.

[9] G. Athanasiou, T. Korakis and L Tassiulas, “An 802.11k Compliant Framework for Cooperative Handoff in Wireless Networks”, in EURASIP Journal on Wireless Communications and Networking, Volume 2009 (2009), Article ID 350643, 14 pages.

[10] G. Athanasiou, T. Korakis and L. Tassiulas, “Cooperative Handoff in Wireless Networks”, in IEEE PIMRC 2008 Social Mesh Networking Workshop, Cannes, France, September 2008.

[11] G. Athanasiou, I. Broustis, T. Korakis and L. Tassiulas, “LAC: Load-Aware Channel Selection in 802.11 WLANs”, in IEEE PIMRC 2008, Cannes, France, September 2008.

[12] G. Athanasiou, I. Broustis, T. Korakis and L. Tassiulas, “Routing-Aware Channel Selection in Multi-Radio Mesh Networks”, in IEEE ICC 2009, Dresden, Germany, June 2009.

[13] G. Athanasiou, I. Broustis, T. Korakis and L. Tassiulas, “Load-Aware Channel Selection in 802.11 WLANs”, under review in ACM/Springer Wireless Networks Journal.

[14] G. Athanasiou, I. Broustis, T. Korakis, and L. Tassiulas, “Load-Aware Channel Selection Scheme for 802.11 WLANs”, Technical report, University of Thessaly, 2008, http://infserver.inf.uth.gr/∼gathanas/Channel2008.pdf.

[15] P. Bahl and V. N. Padmanabhan, “RADAR: An In-Building RF-based User Location and Tracking System”, in IEEE INFOCOM, 2000.

[16] A. Balachandran, P. Bahl, and G. Voelker, “Hot-Spot Congestion Relief in Public-area Wireless Networks”, SIGCOMM Computer Communication Review, vol. 32, pp. 5959, January 2002.

[17] Y. Bejerano, and R. Bhatia, “Mifi: A framework for fairness and QoS assurance in current IEEE 802.11 networks with multiple access points”, In IEEE Infocom, 2004.

[18] Y. Bejerano, S. Han, and L. Li, “Fairness and load balancing in wireless lans using association control”, In MobiCom, 2004.

[19] G. Berger-Sabbatel, F. Rousseau, M. Heusse, and A. Duda, “Performance Anomaly of 802.11b”, Proc. Infocom 2003.

[20] G. Bianchi, “Performance analysis of the 802.11 DCF”, IEEE JSAC, vol.18, pp.535-547, Mar 2000.

[21] V. Brik, A. Mishra, and S. Banerjee, “Eliminating Handoff Latencies in 802.11 WLANs Using Multiple Radios: Applications, Experience, and Evaluation”, in Proceedings of ACM/USENIX Internet Measurement Conference (IMC), October 2005.

[22] I. Broustis, K. Papagiannaki, S. V. Krishnamurthy, M. Faloutsos, and V. Mhatre, “MDG: Measurement-Driven Guidelines for 802.11 WLAN Design”, In ACM MOBICOM, 2007.

[23] K. Brown and S. Singh, “M-TCP: TCP for Mobile Cellular Networks”, ACM Computer Communication Review, vol. 27, no. 5, 1997.

[24] R. Chandra, and P. Bahl,“MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card”, INFOCOM, 2004.

[25] D. Chiu, and R. Jain, “Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks”, Journal of Computer Networks and ISDN, Vol. 17, No. 1, June 1989, pp. 1-14.

[26] “Cisco”, http://www.cisco.com/en/US/netsol/ns621/networking_solutions _package.html.

[27] L. Dai, Y. Xue, B. Chang, Y. Cao, and Y. Cui, “Integrating Traffic Estimation and Routing Optimization for Multi-Radio Multi-Channel Wireless Mesh Networks”, In IEEE INFOCOM, 2008.

[28] Dartmouth Campus-Wide Wireless Traces. http://crawdad.cs. dartmouth. edu/meta.php?name=dartmouth/campus.

[29] C. Doerr, M. Neufeld, J. Fifield, T. Weingart, D.C. Sicker, D. Grunwald, “MultiMAC - an adaptive MAC framework for dynamic radio networking”, DySPAN 2005.

[30] O. Ercetin, “User Association Games in 802.11 Wireless Local Area Networks”, preprint.

[31] A. G. Forte and H. Schulzrinne, “Cooperation Between Stations in Wireless Networks”, in ICNP 2007, Beijing, China, October, 2007.

[32] J. Geier, “Assigning 802.11b Access Point Channels”, In WiFi planet, 2002.

[33] L. Georgiadis, M. J. Neely and L. Tassiulas, “Resource Allocation and Cross-Layer Control in Wireless Networks”, Foundations and Trends in Networking, Vol. 1, No 1 (2006)

[34] A. Giannoulis, M. Fiore and E. W. Knightly, “Supporting Vehicular Mobility in Urban Multi-hop Wireless Networks”, in ACM Mobisys, 2008.

[35] D. Giustiniano, E. Gomma, A. Lopez Toledo, G. Athanasiou and P. Rodriguez, “Breaking TDMA in Mini-Slots: Optimizing TCP Performance in Multi-AP Broadband Access”, under review in INFOCOM 2010.

[36] D. Giustiniano, E. Goma, A. Lopez Toledo, P. Rodriguez, “WiSwitcher: An Efficient Client for Managing Multiple APs”, to appear in Presto 2009. Online available at http://research.tid.es/domenic/images/presto06c-giustiniano.pdf

[37] D. Giustiniano, E. Goma, J. Morillo, L. Toledo, P. Rodriguez, “ClubADSL: When Your Neighbors are Your Friends”, Mediawin 2009.

[38] N. Gupta, and P.R. Kumar, “A Performance analysis of the 802.11 wireless LAN medium access control”, Communications in Information and Systems, vol.3, no.4, pp. 279-304, Sept 2004.

[39] D. Han, A. Agarwala, D. G. Andersen, M. Kaminsky, K. Papagiannaki, S. Seshan, “Mark-and-Sweep: Getting the ”Inside” Scoop on Neighborhood Networks”, IMC 2008.

[40] T-C Hou, L-F Tsao, and H-C liu, "Analyzing the throughput of 802.11 DCF scheme with hidden nodes", Vehicular Technology Conference, 2003.

[41] IBM Wireless Traces. http://nms.lcs.mit.edu/ mbalazin/wireless/.

[42] IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ANSI/IEEE Std 802.11, 1999 Edition.

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

[44] IEEE 802.11h: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Spectrum and Transmit Power Management Extentions in the 5 GHz Band in Europe

[45] IEEE 802.11 WG. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Radio Resource Measurement, IEEE 802.11k/D3.0. New York, USA: The Institute of Electrical and Electronics Engineers, Inc., October 2005.

[46] K. Kaemarungsi, and P. Krishnamurthy, "Modeling of Indoor Positioning Systems Based on Location Fingerprinting", in IEEE INFOCOM, 2004.

[47] S. Kandula, K. Lin, T. Badirkhanli and D. Katabi, "FatVAP: Aggregating AP Backhaul Capacity to Maximize Throughput", NSDI 2008.

[48] R. Karrer, A. Sabharwal, and E. Knightly, "Enabling Large-scale Wireless Broadband: The Case for TAPs", In Proc. of HotNets 2003, Cambridge, USA, November 2003.

[49] B. Kauffmann, F. Baccelli, A. Chaintreau, K. Papagiannaki, and C. Diot, "Measurement-Based Self Organization of Interfering 802.11 Wireless Access Networks", In IEEE Infocom 2007, Anchorage, Alaska, USA.

[50] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M.F Kaashoek, "The Click modular router", in ACM Transactions on Computer Systems, volume 18, pages 263-297, August 2000.

[51] T. Korakis, O. Ercetin, S. Krishnamurthy, L. Tassiulas, and S. Tripathi, "Link Quality based Association Mechanism in IEEE 802.11h compliant Wireless LANs", in Workshop on Resource Allocation in Wireless Networks (RAWNET), April 2005.

[52] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New Insights from a Fixed Point Analysis of Single Cell IEEE 802.11 WLANs", Proc. Infocom 2005.

[53] A. Kumar, and V. Kumar, "Optimal Association of Stations and APs in an IEEE 802.11 WLAN", in Proceedings of the National Conference on Communications (NCC), IIT Kharagpur, January 2005.

[54] Y. Lee, K. Kim, and Y. Choi, "Optimization of AP Placement and Channel Assignment in Wireless LANs", In IEEE LCN, 2002.

[55] D. Lee, G. Chandrasekaran, M. Sridharan, and P. Sinha "Association Management for Data Dissemination over Wireless Mesh Networks", in Elsevier Computer Networks, 2007.

[56] D. Lee, G. Chandrasekaran, and P. Sinha, "Optimizing broadcast load in Mesh Networks using Dual-Association", First IEEE Workshop on Wireless Mesh Networks, 2005.

[57] S. Lee, S. Banerjee, and B. Bhattacharjee, "The Case for a Multi-hop Wireless Local Area Network", IEEE Infocom, Hong Kong, 2004.

[58] D. J. Leith and P. Clifford, "A Self-Managed Distributed Channel Selection Algorithm for WLANs", In WiOPT, April 2006.

[59] B.-J. Leung and K. K. Kim, "Frequency Assignment for IEEE 802.11 Wireless Networks", In IEEE VTC, Vol. 3, pp. 1422-1426, 2003.

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

[61] S. Mangold, Z. Zhong, G. R. Hiertz and B. Walke, "IEEE 802.11e/802.11k Wireless LAN Spectrum Awareness for Distributed Resource Sharing. Special Issue on Emerging WLAN Technologies and Applications", in Wireless Communications and Mobile Computing. New York USA, John Wiley & Sons, Dec. 2004.

[62] M. Mathis, J. Semke, and J. Mahdavi, "The macroscopic behavior of the TCP congestion avoidance algorithm", SIGCOMM Comput. Commun. Rev., ACM Press, volume, 27, 1997.

[63] Meraki Networks Inc. http://meraki.net.

[64] V. Mhatre, and K. Papagiannaki, "Using Smart Triggers for Improved User Performance in 802.11 Wireless Networks", ACM Mobisys, June 2006.

[65] "Microsoft Mesh Network", http://research.microsoft.com/mesh.

[66] Microsoft Research. Self-organizing neighborhood wireless mesh networks project. http://research.microsoft.com/mesh/.

[67] A. Mishra, M. Shin and W. Arbaugh, "Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network", in IEEE INFOCOM 2004, March 2004.

[68] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, "A Client-Driven Approach for Channel Management in Wireless LANs", In IEEE INFOCOM, 2006.

[69] A. Mishra, V. Shrivastava, D. Agarwal, and S. Banerjee, "Distributed Channel Manage- ment in Uncoordinated Wireless Environments", In ACM MOBICOM, 2006.

[70] A. J. Nicholson, S. Wolchok, and B. D. Noble, "Juggler: Virtual Networks for Fun and Profit", To appear in IEEE Transactions on Mobile Computing.

[71] D. Niculescu, "Interference Map for 802.11 Networks", In ACM IMC, 2007.

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

[73] S. Pack, J. Choi, T. Kwon, and Y. Choi, "Fast Handoff Support in IEEE 802.11 Wireless Networks", in IEEE Communications Surveys & Tutorials (CST), August 2006.

[74] K. Pahlavan and P. Krishnamurthy, Principles of Wireless Networks: A Unified Approach, Prentice Hall PTR, Upper Saddle River, New Jersey, 2002.

[75] C. Park, J. Hur, C. Kim, Y. Shin, and H. Yoon, "Pre-authentication for Fast Handoff in Wireless Mesh Networks with Mobile APs", in 7th International Workshop on Information Security Applications, 2006.

[76] D. Qiao, S. Choi, A. Soomro, and K. Shin, "Energy-efficient PCF operation of IEEE 802.11a wireless lan". In IEEE Infocom, 2002.

[77] K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot, "Interference-Aware Channel Assignment in Multi-Radio Wirelss Mesh Networks", In IEEE INFOCOM, 2006.

[78] I. Ramani, and S. Savage, "SyncScan: Practical fast handoff for 802.11 infrastructure networks", In INFOCOM '05: The 23rd Conference of the IEEE Communications Society, March 2005.

[79] A. Raniwala and T. Chiueh, "Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network", In IEEE INFOCOM, 2005.

[80] A. Rao and I. Stoica, "An Overlay MAC layer for 802.11 networks", Mobisys 2005.

[81] E. Rozner, Y. Mehta, A. Akella, and L. Qiu, "Trafc-Aware Channel Assignment in Enterprise Wireless LANs", In IEEE ICNP, 2007.

[82] A. Sang, X. Wang, M. Madihian, and R. Gitlin, "Coordinated load balancing, handoff/cell-site selection and scheduling in multi-cell packet data systems", in Mobicom 2004.

[83] S. Shakkottai, E. Altman, and A. Kumar, “The Case for Non-cooperative Multihoming of Users to Access Points in IEEE 802.11 WLANs”, in INFOCOM 2006, Barcelona, Spain, April 2006.

[84] A. Sharma, E. M. Belding, “FreeMAC: Framework for Multi-Channel MAC Development on 802.11 Hardware”, Presto 2008.

[85] M. Shin, A. Mishra, and W. A. Arbaugh, “Improving the latency of 802.11 hand-offs using neighbor graphs”, In MOBISYS, New York, USA, 2004.

[86] S. Shin, A. Forte, A. Rawat, and H. Schulzrinne, “Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs”, in ACM MobiWAC, 2004.

[87] M. Siekkinen, D. Collange, G. Urvoy-Keller, and E. W. Biersack, “Performance Limitations of ADSL Users: A Case Study”, PAM 2007.

[88] W. Stevens, “TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms”, RFC2001, Jan. 1997.

[89] The MadWiFi driver, http://madwifi-project.org

[90] R. Vedantham, S. Kakumanu, S. Lakshmanan, and R. Sivakumar, “Component Based Channel Assignment in Single Radio, Multi-channel Ad Hoc Networks”, In ACM MO- BICOM, 2006.

[91] H. Wu, K. Tan, Y. Zhang and Q. Zhang, “Proactive Scan: Fast Handoff with Smart Triggers for 802.11 Wireless LAN”, in IEEE INFOCOM 2007, May 2007.

[92] G. Xylomenos, G. Polyzos, P. Mahonen, M. Saaranen, “TCP performance issues over wireless links”, IEEE Communications Magazine, 2001.