QOS ASSURANCE WITH COLOCATED WIRELESS ACCESS POINTS

by

INDRANEEL CHAKRABORTY

Bachelor of Technology (Computer Science and Engineering)
Indian Institute of Technology, Guwahati, India (2001)

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2003

Author .....
Department of Electrical Engineering and Computer Science
May 9, 2003

Certified by .....
John T. Wroclawski
Research Scientist
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

QoS Assurance with Colocated
Wireless Access Points
by
Indraneel Chakraborty

Submitted to the Department of Electrical Engineering and Computer Science
on May 9, 2003, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

The proposed research aims to provide the technical framework which will enable multiple Wireless LANs to provide the assured network service while being in communication range of each other. The Access Points involved compete and cooperate while keeping the overall goal of maximum system utilization above individual optimization. We use channel access time as the means of resource allocation, while keeping the system modular so that any other unit can also be used without changing the protocol. We provide a secondary protocol to ensure robustness of our main algorithm for Access Allocation.

We also propose a model to measure Transport Level Goodput with the help of MAC level parameters. In turn, we also derive MAC level performance in terms of the physical environment characteristics such as number of other stations, APs and their clients. Although simple estimates based on current Internet conditions can be made already, our work helps us estimate the performance which can be assured to clients with a much higher accuracy. This result can be independently applied to model the performance of any wireless network.

Thesis Supervisor: John T. Wroclawski
Title: Research Scientist

Acknowledgments

I would like to take this opportunity to express my heartfelt gratitude to my project supervisor **John T. Wroclawski** for his constant encouragement, guidance and support throughout the course of this thesis. I would also like to acknowledge and appreciate his superb technical advice and constructive criticism. I have learned a lot from him and look forward to learn more.

I also want to thank my parents and my sister, as they continue to inspire, encourage and help me through everything in my life.

I am also grateful to my friends and officemates George, Joanne and especially Steven, for providing stimulation and help whenever I needed.

I would also like to thank the faculty, students and staff of LCS who make LCS such a great place.

Contents

List of Figures

Chapter 1

Introduction

"Engineering is the professional art of applying science to the optimum conversion of natural resources to the benefit of man."
-Ralph J. Smith

1.1 Background

In the initial days, as with everything else, wireless networks were generally underutilized systems. Most applications that ran satisfactorily were adapted to the characteristics of the system, such as limited bandwidth, high packet losses and unpredictable network conditions. Users were satisfied in general as most of the time the network was not congested. If they were unfortunate once, they would retry and succeed in running their network application "pretty soon".

Times have changed.

As the consumption of wireless services grows, in many countries, present day cell-phone networks are already over-allocated making unsatisfactory performance more frequent than ever. Modern applications further exacerbate the problem by demanding better network services than they expected in the past (for ex. streaming video against e-mails). Since the capacity available per Access Point[1] is limited, the most likely option to increase wireless capacity in a location is to deploy more Access Points in the area. This brings one to the obvious problem of proper, automatic and dynamic coordination among these Access Points so that they cause least interference to one another while maximizing the overall system capacity. We envision that in places of commercial interest such coordination might face the added problem that all APs do not belong to the same Internet Service Provider (ISP).

This is not the only challenge modern wireless systems face because of increasing wireless service consumption. With higher system utilization, it can no longer be assumed that users will be able to run their applications in a satisfactory manner (satisfactory being subjective as it depends upon the user concerned) whenever they want, just because they are connected to the network by a "high speed" link which can "potentially" support good Quality of Service. There is a need to provide intuitive assurances to the users of a network service about the quality they can expect. ISPs already claim service assurances in crude forms while selling long term network services. It should not be a surprise to anyone when consumers of short term wireless services also compare their options by seeking quality assurance from the different APs which they can communicate with.

[1]The client hardware which would communicates with the ISPs for network services is called a *Station* or STA and a wireless access point capable of providing assured QoS is referred to as an AP in this document.

Besides comparison of the same service by various APs, a third issue is the ever divergent set of network services required for the entire gamut of wireless devices on the market to serve the varying needs of users. Some of these devices have already started appearing and more will appear to support richer applications or to provide cheaper conveniences. Once such options appear in the market, there is a need to predict what services can actually be assured under the current network conditions and the number of clients and competitors. Besides understanding such a mapping, there is also a need to ensure fair competition among APs and provide a smooth mechanism of changing services to the clients should the need arise.

All this creates the need for a new paradigm of network services, an idea championed by the Personal Router Project [18] of which our work is a part. Already, we see that entrepreneurs are starting to deploy service assured wireless systems in public places. It is a matter of time before many different ISPs choose to deploy multiple Access Points assuring a variety of network services at popular geographical locations.

For this to happen, one needs to provide a technical framework to encourage a business environment where many ISPs compete and cooperate to provide service to users. This, as we mentioned earlier, has been the aim of the Personal Router Project, under the scope of which this research has been done.

1.2 Problem Specification

There are a lot of issues outlined in the above Section which need to be resolved to realize the Personal Router (PR) vision: a new paradigm of wireless network services where clients choose from a whole gamut of options provided by many APs in an area with very short term service contracts. This thesis addresses some of the technical research challenges involved in the umbrella project. Let a *continuous system* mean the exhaustive set of APs which can intercommunicate by wireless links where a message might take multiple hops but reach all the other nodes in the system. This system is composed of APs which may or may not belong to the same ISP. In the latter case, APs are not under a centralized control system which has a complete view of the system.

Then, the main problem this thesis solves is: *"Given a continuous system which uses a common communication protocol (ex. IEEE 802.11b), we aim to provide a protocol which ensures that the Access Points auto-configure themselves dynamically to share the medium in such a way so as to maximize their individual goals of QoS assurance to clients under the restriction that the overall throughput is maximized and the latter objective gets priority"*.

In this thesis, we allow for nodes which can stop and restart, but we do not deal with APs which maliciously fail to follow the protocol. This is because that requires tackling problems involving the business models which will be employed on the PR vision: an issue we seek deal with in the future. (Details of future work are in Chapter 10)

To validate our claims, we also do a simulation study of the protocol, which we call *Access Allocation Protocol* in the rest of the document. The simulation uses IEEE 802.11b [47] communication model, as we focus on that system throughout this project, due to its popularity. However, we do aim to use our protocol on all wireless standards, with minor modifications.

The other issue mentioned in Section 1.1 is the requirement of intuitive assurances to clients about network services in the market. Present day wireless networks do not guarantee QoS to applications in a manner which can be properly comprehended by users. For example, even though the link speed is the same, perceived network performance varies significantly from time to time and user to user. This is because of the known factors of congestion, interference losses, packet drops and other phenomena affecting network traffic. An effort on this front is very useful as that would lead to realistic assessment of Quality of Service on a network.

The third problem outlined in the first Section is of devising bounds for QoS assurances which can be promised under the given network conditions. To realize this objective, we control the variables we can

affect (for example, the number of clients being serviced) and factor in the parameters outside our domain of control (for example, the packet loss because of nearby microwave emissions).

To handle the last two problems, in this thesis, we provide an analytical model to derive MAC level goodput in terms of number of clients and other lower level network parameters.

Further research needs to be done to derive the goodput of TCP/IP connections in terms of MAC level parameters. However, this is beyond the scope of our work.

In the upcoming subsections, we discuss the various facets of the aforementioned problem.

1.2.1 Dynamic Resource Allocation : Sharing the Spectrum

In a competitive environment which the Personal Router Project strives to realize, an important problem faces the networking community. There is only one medium and all competitors have to share it in such a way that they satisfy their personal goal of serving their clients with enough bandwidth, while giving due consideration to the needs of other protocol members. Furthermore, the requirements of each AP change over time as the clientele and the set of preferences of clients keep changing. Hence the allocation scheme needs to deal with the changing environment in real-time. Channel usage needs to be coordinated among nodes which share the same channel and are within communication range of each other.

1.2.2 Maximizing Overall System Throughput

Even though none of the participating APs care for the overall system performance, we presume that the owner of the space (where the wireless environment is located) desires to maximize the utility of the whole infrastructure. Hence, we aim to decouple the problem into two parts:

1. **Aggregate Efficiency:** This work attributes primary importance to maximizing the utility of the whole infrastructure including all the APs and stations.

2. **Individual Optimization:** Each node tries to avail the maximum utility it can provide to its client stations without compromising the first goal.

Such a decoupling of interests helps us to provide a modular technical framework to the entrepreneurs who wish to deploy the PR environment and champion yet unforeseen economic models in the future, including, if they wish, a change in either of the two goals, without affecting the other. We have also kept the unit of system optimization orthogonal to the protocol itself. Our approach to the solution has been detailed in Chapter 4.

1.2.3 Channel Reservation Planning

Each Access Point has some clients (represented by their PRs) with possibly a variety of connectivity requirements. Access Points have to ensure that each of these connections continue to enjoy the QoS agreed upon. This requires each Access Point to have a channel request strategy which strives to ensure that the traffic definitely gets delivered within a stipulated time interval. On the other hand, over-aggressive channel request is penalized[2] as it is undesirable for overall system efficiency. Hence, a proper balance is required between these two decisions.

The problems which need to be addressed here are:

[2]The method of penalty is inextricably linked to the metric which is being optimized, hence we do not specify it in this thesis and leave that for the business model implemented on top of our framework.

1. **Traffic Modeling:** To create an aggregated traffic modeling scheme so that each Access Point can perceive its requirements and request for a share of the channel according to a semi-accurate time-table.

2. **QoS Assurance:** To create a traffic scheduling scheme at each Access Point so that each client of the Access Point gets its due QoS.

However, a lot of research has addressed these problems and hence we do not emphasize channel reservation planning issues in our research efforts. (A discussion of relevant work is in Section 2.5.) This project, in fact, assumes a Distributed Coordination Function (DCF) mechanism for channel access which is a priority-less and reservation-less scheme detailed in the IEEE 802.11 standard [47].

1.3 Research Objectives for this Thesis

Out of the various problems needed to be addressed for deploying the whole framework, in this thesis, we choose to address the research issues which require a solution before a testbed of the framework can be implemented. Once that is done, the next stages of the project can proceed. Hence, the following are the objectives of our work:

1. Devising a protocol so that colocated Access Points dynamically auto-configure themselves to share the spectrum.

2. Maximizing continuous system throughput.

3. Doing it in such a way as to decouple system throughput optimization and individual allocation.

4. Analyzing the performance to decide upon the QoS assurances which can be made.

The next Section gives an overview of the main components of the system required to realize a PR environment which would enable clients to choose between various services and enable us to design, simulate, test and realize the above mentioned objectives.

1.4 The Solution Framework

Our scheme needs to work at various levels to provide the expected QoS assurance. It involves at least the protocols (figure 1-1) discussed in the following subsections. Please note that only Sections 1.4.1-1.4.2 refer to the work we do in this thesis to realize the scheme. The remaining Sections refer to functions which can be created by drawing upon work already done by other researchers.

1.4.1 Access Allocation Protocol

The Access Allocation Protocol seeks to satisfy the following objectives:

1. **Correctness:** A correct execution of the Access Allocation Protocol would assure that more channel access time is never allocated than is available.

2. **Local Maximal Performance:** Each Access point selects the best option for itself without affecting the remaining objectives.

3. **Global Maximal Performance:** The total system performance is maximized under the given execution sequence.

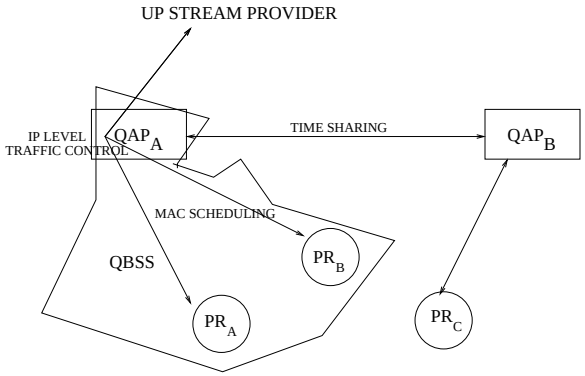

Figure 1-1: System to Assure QoS in Wireless Networks

4. **Stability:** The system should not oscillate between a few solutions, but converge monotonically to one. Also small optimizations (for a few nodes) may be sacrificed for the sake of decreasing network transience.

5. **Robustness:** The system should show graceful degradation of performance when facing deteriorating network conditions such as packets loss and link failure.

Below we give an overview of the sub-protocols needed for implementing the Access Allocation Protocol.

Spectrum Allocation with Ideal Communication Model

To avoid radio interference, it is important to have Access Points within communication range of each other use different frequencies whenever possible. However, the number of channels available for communication is limited by the standard in question, for example 4 channels can be used in IEEE 802.11b at 90% of their maximum capacity or 3 at the maximum capacity of 11 Mbps [38]. So in unfortunate circumstances, (which might happen often in dense areas) some APs within communication range might have no option but to share a channel. Naturally, such nodes cannot send more data than the channel allows. These are some constraints on our algorithm. A detailed description of AAP follows in Chapter 4.

Neighborhood Awareness Protocol to Ensure Robustness

The Access Allocation Protocol requires that each AP needs to keep relevant information about all other APs which are within the communication range of its stations. This includes APs on channels other than that of the AP in question. Also, in many cases, such APs may not be in direct communication range. In the worst case, there might be no wireline connection through the Internet between them and the intermediate stations establishing a wireless link might also be temporary. In chapter 5, we develop a protocol to deal with such scenarios and yet provide information accurate enough to the Access Allocation Protocol for making the correct channel distribution.

1.4.2 Channel Allocation Unit

After allocating channels to APs, the second step is to allocate resources within the channel. This can be done in many ways, the main options being:

1. *Bandwidth Controlled Access:* Each BSS is allocated a fixed amount of bandwidth.
2. *Channel Access Time Control:* Each BSS is allocated a fixed amount of time to access the channel.
3. *Priority-based Access:* Each BSS is prioritized for channel access.
4. *Uncontrolled Access:* There is no control on channel access, simple DCF mechanism is followed.

We have decided to use channel access time control as the unit of resource allocation. We understand that use of bandwidth as a unit would be more intuitive to users. Our choice of channel access time is the unit of resource allocation, is however a matter of preference. It is not binding on the protocol discussed in Chapters 4 and 5. We did not decide to use bandwidth control because in equal amounts of channel access time, different APs might be able to achieve different levels of bandwidth with their clients due to their relative distance and other parameters of the wireless environment. Channel access time allocated to each AP is an easily monitored metric (which will help us in future work to enforce protocol compliance) and does not depend upon such extraneous characteristics as applications run by clients and upstream network conditions or the prevalent channel conditions (which is the case for bandwidth). Hence allocated channel access time seemed more of an equitable trade unit. However, our protocol is completely separated from the choice of the allocation unit, and bandwidth can as easily be used. As for the last two choices of Priority-based and uncontrolled access, these mechanisms simply cannot provide QoS assurance because of their very nature of ensuring no fairness among nodes competing for channel access.

1.4.3 Additional Issues

There are many other important research questions which need to be answered for the whole technical framework to be implemented. The important ones out of them are mentioned below. However, since there is a lot of work already available on these issues, we do not deal with them in this thesis.

IP Layer Traffic Control

An IP layer protocol is required to control the traffic, based on application service parameters. This is required because even if the MAC layer channel access time is bounded by the Access Allocation Protocol, the exact throughput achieved at IP layer might exceed that which is allowed to the connection. In case the policies of the ISP require that such extra throughput be capped, it can be easily done by using available Traffic Queue Control programs.

MAC Layer Traffic Control

As the IEEE 802.11 standard has been revised [48], APs can now choose to implement a hybrid protocol allowing them to have a reservation based channel access scheme which can provide QoS guarantees. But since fine-tuned synchronization is difficult among APs which are not in direct communication range, DCF is the only choice for inter-BSS coordination, forcing the whole access mechanism, even within a BSS, to be DCF. However, if tight coupling among APs can be achieved because of direct communication, then APs can employ the hybrid scheme. A lot of work is available in this field. (See Section 2.5) Hence, we do not delve into this issue any further.

1.5 Organization of the Thesis

We begin by a survey of the work in related fields in Chapter 2. We suggest the metrics which can be used to allocate the spectrum in Chapter 3. Chapter 4 provides the resource allocation algorithm for sharing

the spectrum. A support mechanism to provide robustness to the Access Allocation Protocol under transient and indirect communication is described in Chapter 5. A proof of correctness, stability and maximal performance is given in Chapter 6, along with a formal description of the main portions of the algorithm. Chapter 7 provides simulation results of the Access Allocation Protocol. We give an analysis of the maximum attainable system throughput in Chapter 8. We describe the implementation work for realizing the prototype of the PR Project in Chapter 9. We conclude our work and discuss the future research direction in Chapter 10.

Chapter 2

Related Work

"I've never known any trouble that an hour's reading didn't assuage."
-Charles De Secondat

A plethora of literature is available on all the topics related to this thesis. Relevant work can be broadly classified into the following categories:

2.1 QoS with Multiple Base Stations

Some research has been done about the scenario where multiple Access Points compete to provide various network services to the stations present in that area [51, 53, 54]. These papers describe a penalty system to discourage greedy use of shared channel resources such as transmission duration, bandwidth and power. It is clear that with a proper penalty system, greed can be avoided. (For ex. for sharing transmission duration, linear penalties on the basis of time the channel was held by a device suffices to avoid greed completely.) The authors assume that greed leads to the undesirable situation commonly known as the "tragedy of commons", and hence should be penalized.

However, the penalty model discourages every channel access, some of which are obviously justified. Given that all APs are not servicing equal amounts of traffic, it is fair for some APs to utilize more resources than others. Hence, even though such a penalty system can avoid greed, the problem of resource allocation for justified requirements remains unsolved. In Chapter 4, we do precisely that. APs get network access depending upon their need by mutual agreement. This can be combined with the penalty system which may be used for distributing uncontrolled access time among APs.

2.2 Graph Coloring

Channel distribution among a set of proximal Access Points can be modeled as a minimum coloring problem. A lot of research has been done on minimum coloring algorithms. The minimum coloring problem is to assign a color to each node so that every incompatible pair is assigned a different color. This is NP-hard for general graphs [26].

The minimum coloring algorithms of interest in our case are those which have the following characteristics[1]:

[1]While we require many more characteristics from our Access Allocation Protocol like stability against topology changes, robustness against network conditions and maximal system performance, these objectives are not under the purview of graph coloring problems.

- *Partial View:* Algorithms which do not require global view of the system, as this is difficult to obtain among many ISPs and for large regions.
- *Local Maintenance:* Algorithms which can react locally to a change in graph, rather than requiring complete reallocation for correctness.
- *Fast Convergence:* Algorithms which converge fast to a correct solution which may not be unique.

The technique which can serve our objectives is successive augmentation. In this approach a partial coloring is found on a small number of vertices and this is extended vertex by vertex until the entire graph is colored. Examples of variants of this approach include [4, 9, 28, 44, 50, 62, 63, 64]. For our problem, we take inspiration from a distributed version of the successive augmentation approach along with some new algorithms for robustness, stability and maximal performance (Details in Chapter 4).

General heuristic techniques could have also been used. This includes simulated annealing [13, 37] and tabu search [32]. Problems associated with the application of these techniques for Frequency Assignment will be discussed in Section 2.3.

2.3 Frequency Assignment for Cellular Mobile Systems

The studies of a frequency assignment problem (also called channel assignment problem) in cellular mobile systems have a long history [8, 25, 30, 52, 57]. Various AI techniques, including constraint satisfaction, simulated annealing, neural networks, tabu search and GA have been applied to this problem [12, 23, 24, 31, 35, 42, 45, 58].

While the problem is similar, there are marked differences. Cell tower positioning and frequency allocation to towers is not done in realtime. This also means the architects of the system have good estimates of the traffic on each tower beforehand. There is no intelligence at a tower per se to negotiate channel access. Negotiations between providers, if any, are done manually. Hence, the system is neither supposed to auto-configure nor react to varying network conditions in real time.

Therefore, solutions in this field, with proper reason, assume global knowledge, and expect to be executed offline with perfect information about the system. Even though distributed versions of the algorithms are available [67] and may be applied to the cellular problem, these algorithms still assume a predefined priority order among cells. This is not possible among multiple APs belonging to many ISPs with dynamic network conditions. Characteristic WaveLAN difficulties, for ex. hidden terminal problem and channel access coordination issues, do not even figure in the literature of this field.

Our solution on the other hand, works with wireless LAN issues in mind. It does not assume any offline priority system between APs and lets each AP negotiate its own channel access time with its neighbors. This allows us to auto-configure whole or part of the system without a global view, enables us to react to changing network conditions in real time and provides a framework for multiple or a single ISP to work in a region without manual monitoring of system performance. We also guarantee quick convergence to a stable solution.

Though we aimed to solve the spectrum allocation problem for only Wireless Access Points in an area, our approach covers all the requirements of cellular industry as well. Hence, it can be applied there also for online frequency assignment among cells.

2.4 Analysis of IEEE 802.11 MAC Performance

Research about performance of CSMA protocols over radio channels has been studied in [43, 60] and improvements like MACA [39] and then MACAW [5] have been available for a while. Several other papers

(seminal ones being [7, 11, 19, 33, 61]) have studied the maximum throughput achieved in IEEE 802.11 protocol under various network configurations.

However, such performance estimates of the theoretical maximum throughput cannot be directly applied to practice. We require an understanding of the real MAC layer throughput achieved when multiple stations are competing using DCF[2] mechanism of IEEE 802.11 standard. In case of maximum system utilization, all stations will always have data to transmit, a state referred to as *saturation throughput*. In this regard, the work of Bianchi [6] which computes saturation throughput performance is more pertinent. The assumptions in their work are a finite number of stations and ideal channel conditions. Reference [65] is based on the same model and takes into account the frame retry limit.

While the hidden terminal and packet capture problems for IEEE 802.11 [17] and CSMA [34] have been considered, the analyses in these papers are not for saturation throughput. These works assume an exponentially distributed rate of packet arrival. Also, the stations in the system have equal probability of communication with any other station, while in our case we assume all stations communicate with only their APs. As the models described in these works quickly become complex, the authors are forced to approximate and can only provide performance lower bounds for the system.

We utilize the analysis done in [6, 65] for deriving the saturation throughput for each BSS in an environment with co-located APs. The simplicity of this model with its accurate estimation of saturation throughput in single WLAN case, proves attractive to us for extending it to include multiple APs and the hidden stations thereof. As we assume RTS/CTS access mechanism, capture effects can be disregarded according to reference [29] which does this study under the influence of Rayleigh Fading and near/far effect.

2.5 QoS Support in Wireless Networks by Modifying MAC

The PR environment requires an 802.11 MAC compliant protocol which provides the flexibility to implement various service profiles ISPs come up with from time to time. A lot of work has been done on QoS Guarantees in Wireless Networks following 802.11 standard, some of the recent ones being [16, 27, 41, 55, 56, 66, 68]. [1, 2, 3, 14, 15, 36, 40, 59, 69] give example of work for wireless networks in general. While all the above mentioned research is good for providing QoS assurances for specific needs, none have the flexibility which is required for the PR environment. QoS support protocols available today are designed to improve or assure performance for a predefined class or set of services. However, the PR environment requires a technical framework where yet unforeseen services can be implemented without any change in the protocols. Two main requirements from the protocol are:

- **Flexible Sharing Scheme:** Without a change in the underlying protocol, but with only a change of parameters, contending stations should be able to get prioritized, equal share or preallocated shares of the medium.
- **Bounded Delay and Jitter:** Similarly, by tweaking the parameters only, the protocol should enable bounds on delay and jitter in absolute or statistical terms.

We foresee a need for research to address this issue.

A general understanding of CSMA MAC layer shows that three mechanisms can be used to give priority to traffics [1, 2]:

1. Different Back-off Periods after collision.

[2]We choose DCF as the coordination mechanism under study as it allows untrusting nodes to function without a coordinator. We believe that to be the best option for the PR environment.

2. Different DIFS periods.

3. Different Frame Lengths.

The third approach of different frame lengths is the only one which assures proper prioritizing among UDP as well as TCP flows. This is useful for our work as we have to implement our system with a simple CSMA MAC. However, this aspect has not been researched in this thesis and will be handled in the future (See Chapter 10 for details).

For the purposes of this work, we will not place any conditions on the frame lengths for each transmitting station. We assume stations use Distributed Coordination Function (DCF) [46] for channel access coordination. DCF proves useful as we do not want to assume the ability of time-synchronization between APs.[3]

This Chapter provided a survey of the relevant research material. The next Chapter introduces the data items required to understand the Access Allocation Protocol.

[3] An inter-BSS reservation based MAC scheme (following the latest draft supplement to IEEE 802.11 standard 1999 edition [48]), would require fine-grained time synchronization between APs - a rather difficult technical feat given present technology.

Chapter 3

System Inputs and Performance Indicators

"Not everything that can be counted counts,
and not everything that counts can be counted."
-Albert Einstein

In this Chapter, we discuss the inputs to the Access Allocation Protocol described in Chapter 4 and the performance indicators of our solution.

It is our aim to provide Quality of Service Assurance to users using intuitive terms. Users understand performance in relative terms like "better" and "cheaper".[1] The perceived performance of the same network service is completely dependent on the application concerned. Application developers can map user satisfaction for their particular application to goodput, delay and jitter bounds. However, even these are not applicable directly at the physical or MAC layer.

Hence, this Chapter provides the "missing link" in the understanding of QoS Assurance to wireless networks by providing parameters which are applicable at lower layers. These variables are mapped to system performance indicators intuitive to application developers, who then map it to user intuitive performance metrics.

We introduce here metrics which are meaningful for our protocol and then show in Chapter 8 the method to calculate system performance for MAC layer and even TCP connections for all file sizes.

This work studies the first hop link performance for connections available to each station. The final perceived goodput is also a function of network conditions beyond the first hop, which in most cases, are beyond the control of the ISP of the AP.

3.1 System Inputs and Outputs

As mentioned above, this Chapter will introduce terms relevant to Access Allocation Protocol (AAP). AAP needs two inputs and generates two outputs:

- **Inputs:**
    1. Required Performance: Access time or Link capacity
    2. Associated Stability: Disruption Cost
- **Outputs:**

[1] A separate research work is on within the personal router project to understand user satisfaction and choose services accordingly[22].

1. Operating Channel: One of the possible ones
2. Available Performance: In terms of input specified

One input to the system needs to be a quantifiable amount of the network resource to be allocated, specified using any metric. Link capacity is an obvious choice for the metric. Another possible option is channel access time. We will discuss our choice of input metric in Section 3.2. The second input is referred to as *Disruption Cost* in our work. This refers to the internal measures of stability of each resource allocation (section 3.3). In case of a change in network conditions, the nodes which are the least resistant to change are reallocated first. This minimizes the disruption which happens as a consequence of spectrum reallocation.

Channel of operation of an AP and its clients is the first output of the Access Allocation Protocol. Even though this is not a user-perceived concept of performance, for any network service at all, a communication channel needs to be allocated. Available performance at AAP level does not mean user-perceived performance. Instead, it is a fraction of the network resource requested by a client and is measured in the same unit as that of the request. As mentioned earlier, the derivation of QoS in terms intuitive to clients from available channel access time has been detailed in Chapter 8.

Below we introduce the system inputs and outputs which are used in the Access Allocation Protocol described in Chapter 4.

3.2 Channel Allocation Unit: Access Time as Input and Output

Required channel share is an input while available channel share is the output of the Access Allocation Protocol. As discussed in Chapter 1, channel allocation can be done in a variety of ways. We choose to do it by allocating the total access time each BSS[2] gets. The negotiations for the cumulative access time required by a BSS is done by the AP responsible for the BSS. Choosing channel access time for allocation gives us the following advantages:

1. Constant Unit: Regardless of any temporal effect, absolute access time is an equitable measure of channel share. Bandwidth, however, would vary with network conditions. In other words, APs can share access time in absolute units; in the end, how much bandwidth they get out of it, is an internal issue among the AP and its stations. The transparency thus obtained is appealing to us.

2. Simpler Delay Estimations: If we know how much time each access lasts, it is much simpler to estimate and assure delay bounds. Bandwidth on the other hand would require us to know what link speed is available between a station and its AP.

3. Easier to Monitor: Rather than monitoring how much payload is being transferred per communication between two nodes (which one needs to do for bandwidth control), simply monitoring access time is easier. This would help us to devise a protocol to monitor nodes to ensure their compliance with the protocol.

The disadvantage of using channel access time is its non-intuitive nature to clients, a higher access time as such does not mean higher user perceived performance. Hence each AP calculates[3] the bandwidth it can get for a given access time and physical network conditions, such as distance between the AP and a station, prevalent channel error rate etc. However, as is obvious to the reader, this disadvantage is inseparable

[2] A system where one AP is connected to wired network and a set of wireless stations is referred to as a Basic Service Set (BSS).

[3] Chapter 8 shows a method to calculate values intuitive to customers from non-intuitive system parameters by demonstrating the intrinsic mapping between them.

from the advantages: the fact that channel access time is an absolute scale giving different performances to different stations is the precise reason why it is not an intuitive performance metric.

For this thesis, without loss of generalization, we arbitrarily assume that the available access time is 1000 units. Chapter 4 describes how channel access rights are divided among the APs by using channel access time or any other unit as the metric.

3.3 Disruption Cost as Input: Explanation and Estimation

Chapter 4 discusses the notion of *disruption cost* as the penalty the network has to pay when an AP switches from its present channel to the channel next in the order of preference. The total disruption cost is a sum of the individual penalties each AP pays when a channel switch by one node forces others to switch channel. This Section provides a method to calculate the individual penalty for a switch. We assume all stations decide to switch their channel when the AP servicing them switches.[4] This is composed of two types of performance losses:

1. Transition Penalty: The immediate and complete loss of service while and just after switching.
2. Long Term Penalty: The loss of service due to reduction in the channel access share of the BSS after the transition is over, as compared to the value before the channel transition started.

3.3.1 Transition Penalty

The transition penalty is hardware dependent, and is attributed to the following causes:

1. Protocol overhead for coordinating a switch.
2. Time to switch channel and the penalty of complete loss of service in that time.
3. Time until connections stabilize after switching channel.

The protocol overhead is minimal. A message $SWITCH\_CHANNEL$ with the time of switching, channel to which AP is switching and new channel access time of each station is broadcast by the AP. This message may be broadcast upto a maximum of $MAX\_BROADCAST$ times at random time intervals. At the time which was being broadcast, all stations and the AP switch without any further coordination.

The exact time interval t_s required to switch channels is hardware dependent. The loss of throughput during the switch is dependent on the average throughput and t_s.

3.3.2 Long Term Penalty

The long term penalty essentially is the decrease in channel access share for the BSS from the previous configuration. This can be easily calculated by the method explained in Section 8.2. Hence, APs can estimate their new link capacity even before switching occurs. Using the $SWITCH\_CHANNEL$ message, it can then allocate channel access rights to its client stations, which will be enforced after the upcoming switch. This leads to minimum disruption when switching occurs.

[4] This might not always happen, as stations are free to choose a different AP at any moment of time, but due to a lack of a model regarding this possibility, we choose to not consider it.

3.4 Outputs

There are two outputs generated by an execution of the Access Allocation Protocol:

1. Operating channel for each AP
2. Available channel share

AAP allocates the spectrum among various contending APs by assigning one of the hardware specified communication channels to each AP. In cases when some APs need to share a channel, we require a channel share allocation unit, which as discussed in Section 3.2.

3.5 Other Options for System Performance Measurement

Many other system inputs and outputs can be used in place of channel access time and switching penalty. For example, allocation choices might be based on the monetary compensations obtained from stations for the services provided. However, we do not explore such options in this work, as we concentrate on providing the technical framework to realize all the possible business strategies for wireless service, and not any particular strategy.

Chapter 4

The Access Allocation Protocol

" The greatest achievement of the human spirit is
to live up to one's opportunities and make the most of one's resources."
-Vauvenargues

In this Chapter, we present the main protocol of this thesis which aims to allocate resources in a manner which maximizes system throughput under the set of constraints imposed on the system. Formally, our protocol seeks to satisfy the following objectives:

1. **Correctness:** A correct execution of the Access Allocation Protocol would assure that more channel access time is never allocated than is available.
2. **Local Maximal Performance:** Each Access point selects the best option for itself without affecting the remaining objectives.
3. **Global Maximal Performance:** The total system performance is maximized under the given execution sequence.
4. **Stability:** The system should not oscillate between a few solutions, but converge monotonically to one. Also small optimizations (for a few nodes) may be sacrificed for the sake of decreasing network transience.
5. **Robustness:** The system should show graceful degradation of performance when facing deteriorating network conditions such as packets loss and link failure. (This objective is addressed by the subsidiary protocol described in Chapter 5.)

We first present the terminology used to describe the protocol, and the assumptions we make. Then we give an overview of the protocol and in the end we delve into the details of each module of the algorithm.

4.1 Notations and Assumptions

We model a network as a graph $G = (N, L)$, where N is a finite set of nodes and L is a set of undirected links. Each node[1] $i \in N$ is assumed to have a unique node identifier (ID), and each link $(i, j) \in L$ is assumed to allow two-way communication (i.e. nodes connected by a link can communicate with each other in either direction). We also assume reliable communication between nodes for the protocol described in

[1] We interchangeably use the word *node* to refer to APs.

this Chapter. [2] Due to possible AP failures, new connections between APs through stations or new APs starting operation, the set of links L is changing with time (i.e. new links can be established and existing links can be severed). From the perspective of neighboring nodes, a node failure is equivalent to severing all links incident to that node.

Each node $i \in N$ may subsequently be assigned one of two states; (1) Dead (2) Alive. If a node is alive, then it has two state variables set, $channel$ and $accesstime$. These describe the channel range on which the AP is communicating and the time on the channel an AP and its clients need to communicate with each other.[3] For each node i, we define the "neighbors" of i, $N_i \in N$, to be the set of APs i.e. nodes j such that $(i, j) \in L$.

For the subsequent discussion, we assume the existence of a protocol (described in Chapter 5) which tries to ensure that each node i has the latest information about its neighbors in the set N_i. However, we understand that this might not be always possible and there may be arbitrary delays in the time between a change in the network and subsequent protocol notification of the change. Our protocol has been designed keeping that in mind, and strives to provide the best results it can with the help of the available information. We also assume that between any two nodes, transmitted packets are received in the order of transmission. Finally, we have assumed that when a node i transmits a packet, it is broadcast to all of its neighbors which belong to the set N_i. In case of packet loss, subsequent protocol steps rectify the situation, hence we do not need to assume eventual delivery of all packets. This is an important advantage of our protocol as it seeks to handle congested wireless environments.

4.2 Foundation and Basic Structure

A logically separate version of the protocol is run at each node. The protocol can be separated into three basic functions:

1. Decision to allow or deny the neighbors a channel and/or access time
2. Calculating Disruption Cost of switching one's channel
3. Allocating one's channel based upon information available

The protocol accomplishes these three functions through the use of three types of messages: REQUEST, REJECT and VOTE. REQUEST is received by a constraint-evaluating node from a value-sending node, asking whether the value chosen is acceptable. It also provides the state information of value-sending node to the constraint-evaluator so that the latter can update its neighborhood information. REJECT is a message that a value-sending node receives, indicating that the constraint-evaluating node has found a constraint violation and that the former wants the value-sender to change its state to rectify the situation. VOTE is a packet sent by all nodes to exchange system performance information and arrive at a decision to reallocate channels system-wide.

The decision to allow or deny the neighbors a channel with a a certain access time is taken whenever a neighbor wants to change state or when the node itself is readjusting its channel or access time. The neighbor is always allowed its choice of channel in case the channel is a different one or when the total capacity of

[2] We understand that two-way communication and reliable message delivery are not possible in 802.11 and other radio technologies. Chapter 5 discusses in detail how to establish a reverse communication link if there is only a unidirectional wireless link. It suffi ces to say here that the reverse link in such cases consists of multiple hops. Chapter 5 also relaxes the assumption of reliable message delivery by introducing a separate data collection protocol.

[3] In the Access Allocation Protocol, an AP negotiates for the total time it and its client stations need on the channel. Distribution of the obtained time is considered an internal issue of the set.

the channel is not exceeded. If it is not the case, then the decision is made in a way to reduce the disruption in the network.

The disruption cost d essentially measures the relative ease with which a node can switch its channel. In case a node can switch channel without causing any difficulty for any other nodes, the disruption cost is just the loss of service it perceives itself. However, when a node switches channel and the nodes present on that channel are perturbed by this situation, then d includes the effect on the whole network of such a change. The measurement of d is central to solving conflicts in this protocol. For this, each node keeps a table of the disruption costs of all its neighbors and a list of the nodes which cause that neighbor to have that d. As a simplification, the disruption cost which a given node reports is the sum of the respective costs of its neighbors, all of which are on a channel, say c. Channel c is the next best choice of a node, if it is forced to switch from its present channel. Whenever a node includes the disruption cost of other nodes into its own, it checks for and removes duplicate weights which are caused by common ancestor nodes. For this, apart from the numerical value of disruption cost, the state of each node also includes a list of all the other nodes which go to create the given value of d for the node. This helps to avoid counting the same node twice.

Each node chooses the channel which has the maximum channel capacity. If many channels have this capacity, then the node selects a channel at random. If no channel can support the access time the node seeks, it decreases its required access time to the maximum available capacity. It then informs all its neighbors about its choice of channel. In case any neighbor objects to the choice of the channel (by sending a REJECT message), the node takes the next best choice.

4.3 Working of Access Allocation Protocol

Figure 4-1 is a flowchart of the algorithm as a whole. When an AP has no allocated channel, it chooses the channel which provides it maximum channel access rights. Figure 6-1 shows the pseudo-code for the channel selection module. Section 4.4.1 explains it in detail.

Once a channel is chosen, state information is sent as REQUEST message to all neighbors. This is periodically repeated on all channels. If a REJECT message is received from a neighbor, then the AP switches to the next best which gives it maximum channel access under given constraints. If a REQUEST message is received from a neighbor, the receiving AP ascertains whether the configuration thus arrived violates any constraints. If so, the APs with the lowest disruption cost are sent a REJECT message. The only exception is when the AP detecting constraint violation is itself among the lowest disruption cost nodes which need to switch. In that case it switches silently without sending any other REJECT messages, on the assumption that the violation, if it still exists, will be addressed by the remaining nodes. A guarantee that it will be the case will be given in chapter 6. If no violations are detected because of the state information available due to the REQUEST message, then no action is taken. Figure 6-2 shows the pseudo-code for the communication logic for interaction with neighbors. Section 4.4.2 explains it in detail.

The protocol works under various modes:

- *Complete Allocation Mode:* In this all the nodes in the network are triggered to start the protocol at the same time. This can happen when the whole network is initialized or when the network sees it fit to improve system performance (through an election mechanism described in Section 4.4.5).

- *Suppressed Allocation Mode:* This is the default mode in the absence of a specific complete reallocation trigger. In this mode, nodes trigger partial channel reallocation in a portion of the network. However, many nodes can trigger suppressed allocations with overlapping time intervals.

In the next Section, we discuss the protocol in greater detail.

4.4 Detailed Description of Access Allocation Protocol

At any given time, an ordered quintuple $C_i = (i, f_i, l_i, d_i, array_i)$ is associated with each node. Conceptually, the quintuple associated with each node represents the present network state of the node and the cost of changing that state. The state is represented by the first three values while the cost of changing the state is represented by the last two values. A new cost of switching channel again is defined each time a node switches its channel due to a change in network state. The first value, i, is the unique ID (UID), of each node. This ensures proper communication and book-keeping. The second value, f_i, is the actual channel the node is using for communication presently. The third value, l_i is the time which the AP and its clients have won after negotiations to access the network. In other words, this is the total share of the network resource the AP and its clients rightfully get to complete all of their communication. This value is used to choose a channel by the node. The fourth value, d_i is the cost incurred by the network if node i changes channel. This value is instrumental in stabilizing the network and ensuring partial ordering among nodes with conflicting choices. The last value represents a list of UIDs representing the nodes which will be affected if node i switches channel. This is explained in detail in Section 4.4.3.

Initially the channel f_i of each node is chosen at random with the access time l_i it wants. Subsequently, the network state of each node i can be modified in accordance with the rules of the protocol. In addition to its own state, each node maintains a list nbr_list of its immediate neighbors with their latest quintuple. This information is piggy-backed each time there is protocol correspondence between nodes. We realize that this information might be old and wrong, but we allow that in our protocol as we seek to handle nodes which are connected intermittently. whole.

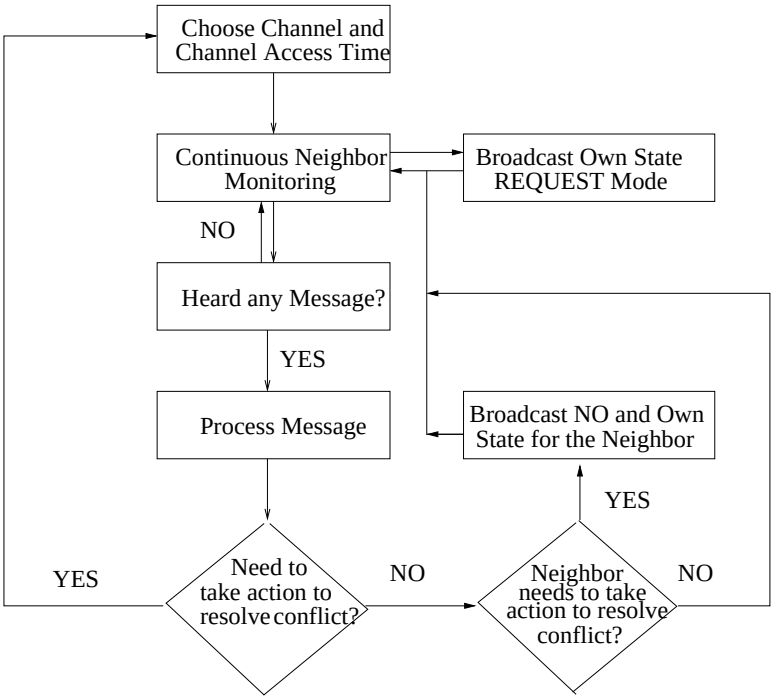

Figure 4-1: A Flowchart of the Access Allocation Protocol

4.4.1 Choosing Channel

Each node i checks the list of its neighbors and computes the total access time available on each channel. By a local computation on the nbr_list information, the channel f_i selected for communication is, the one which has the minimum value of Σl_j, where $j \in nbr(i)$ and $f_j = f_i$; i.e. the channel has the most available bandwidth among other options. This is in consonance with our goal of maximizing system throughput. If there is more than one channel satisfying the criterion, a channel is chosen out of the qualifying set at random. If none can sustain the amount of access time the node needs, the node reduces its demands to the maximum available resources on the channel of choice. The node now broadcasts a REQUEST packet with its quintuple and a time-stamp to its neighbors. In case the best choice has been denied, node i chooses the next best channel, provided reassessment does not give a new priority list among choice of channels. In that case the new order is pursued.

4.4.2 Processing Messages

In this subsection, we describe how incoming messages are processed by a node. Figure 4-2 is helpful in visualizing the steps involved.

Handling REQUEST packets

REQUEST messages coming from neighbors are utilized to serve two purposes: (1) Update Neighborhood Information (2) Decide if constraints are violated. Let node i request response from node j about the choice of channel node i has made. For performing the first function, the timestamps are compared and latest information is maintained about the transmitting node in the neighbor list nbr_list along with the time-stamp of the packet which brought the information. To perform the second function, there are a few cases to choose from depending upon the relative states of the communicating nodes.

- **Acceptable Channel Assignment:** If the channel of the requesting node j is different from the responding node i, i.e. $f_i \neq f_j$[4], or if the access times can be provided by the same channel i.e. $\Sigma l < T$, nodes agree to each other's configuration. In such a case, a reply serves the sole purpose of providing the latest information about node j to the original requester.

- **Impossible to co-exist:** This case happens when a node gets information (because of a latest REQUEST packet) that nodes around it have caused the channel access time to be higher than maximum channel capacity T. In such a case, neighbors are sorted according to their disruption cost d. Working up from the node with lowest value of d, as many nodes as required on the list are sent a REJECT message, until the total access time of the channel due to all remaining nodes is below maximum channel capacity T. The information about the nodes to whom REJECT is sent, is deleted from the list of neighbors maintained by the node in question, as their channel assignment is going to change when they get the REJECT message. The response time guarantees for REJECT messages to arrive after an undesirable configuration happens will be discussed in Chapter 6 along with other time guarantees.

  If, however, node i itself falls in the list of those who need to change channel, then the node simply changes channel without sending any REJECT messages and broadcasts a REQUEST packet with its latest information to its neighbors. This is done because the conflict had been perceived by node i; node j a potential candidate for receiving a REJECT, might not face a conflict with another neighbor

[4]This can happen only through indirect communication or when one AP switches its channel to broadcast its current state on other channels. These two situations arise due to the protocol described in Chapter 5

of node i, say k, as they might be out of range of each other (thus solving the hidden terminal case for our protocol). If they do, it is left upon themselves to resolve it rather than a third party (node i in this case). The guiding principle is thus being 'selfish', by solving only the conflicts which affect the node which detects them. *This is also a attractive policy decision for our whole system as we finally wish to establish a system where nodes working with only self-interest can co-ordinate.*[5]

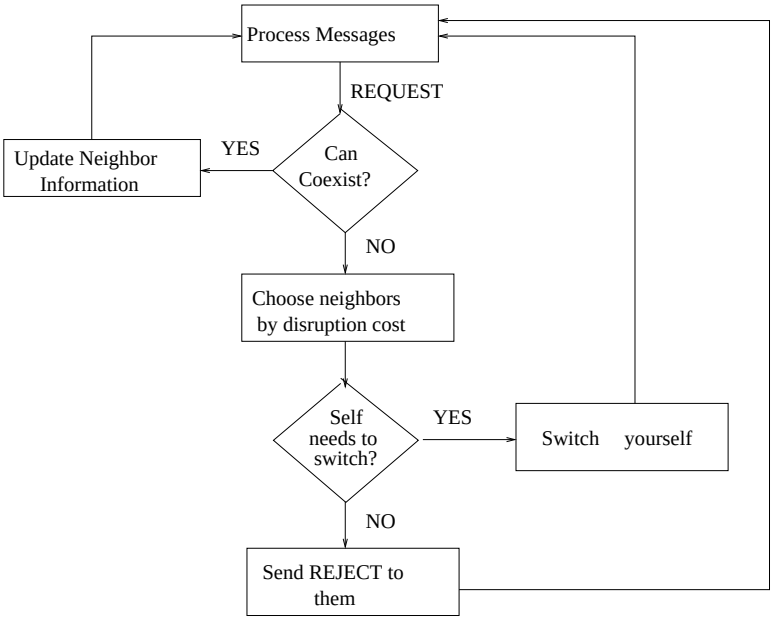

Figure 4-2: A Flowchart of the Message Processing Module

Handling REJECT packets

REJECT messages coming from neighbors also serve the purpose of providing the latest information about the sender. The main goal is to inform the recipient node that it needs to switch channel. In such a case, the node chooses the next best channel which has the maximum available bandwidth.

4.4.3 Calculating Disruption Cost

For proper stabilization of the protocol, and to properly identify the interconnections of channel assignment in the network, the protocol uses a new concept, which we call *disruption cost*. Essentially, *the disruption cost of a node provides a metric to measure the expense which the whole network bears for a change in channel assignment from its present choice c to its next best choice c'*. The calculation is done as follows:

[5]Even though we have chosen to be selfi sh whenever possible with an eye on the fi nal business models which will essentially be self-serving, we understand that nodes in our protocol are not always working in self-interest. The case under consideration is that nodes subscribing to our present protocol defer to overall system optimization against personal goals if required. However with the introduction of an incentive model , that requirement will be relaxed.

- Nodes which can switch their channel without causing any other node to switch channel, cause a disruption proportional to only their own clients. The exact penalty of switching a channel and relinquishing a portion of one's share of the network capacity is calculated in Section 3.3.

- A node which, because of its switching to a channel, cause a disruption in service to the nodes already present on that channel reports a total disruption cost equal to the sum of the disruption cost of the nodes it would disrupt plus its own disruption cost.

To avoid counting the same node twice, disruption cost is maintained as a list of all the nodes which are causing the value being reported. Thus when values are added, the disruption cost of the same node is not added twice.

It is notable that disruption cost is in a way creating the dependency graphs of channel assignment, under the present allocation, from the system. This avoids unnecessary fluctuations in channel assignments. Also if t represents the time bound on the delivery of a message on a link and d represents the network diameter, then disruption cost mechanism provides $O(td)$ worst case time bound on the completion of the protocol if nodes never change their network access time demand. In the case when nodes are allowed to have a monotonic decrease in channel access time, the stabilization time is worse. Let us assume that there are n channels to choose from. If all respective channel access time demands are sorted in a decreasing order, let the access time of the node which is at $(n+1)$th position be represented by l. If all nodes are within communication range, and l cannot co-exist with a node having higher access time, then an execution sequence can be created where the node with channel access time l is denied its choice of channel until it decreases its demand. In such a worst case scenario, $O(tdl)$ represents the time bound for the stabilization of the system. A detailed worst case performance analysis is given in chapter 6.

4.4.4 Continuous State Monitoring

Nodes continuously update neighborhood information. This helps in two ways:

- In case a new node appears or a node's channel access time changes, a suppressed re-allocation protocol is triggered by the node which has appeared. In such a case, the same protocol is followed as described above, however, a hop-count field is provided in the REQUEST packet. Nodes which broadcast their response to the initial REQUEST packet, decrease the hop count by 1. This creates a *radius of turbulence* as nodes outside the radius do not respond to the protocol as they see a hop-count of 0. The nodes at the periphery of the radius of turbulence decrease their channel access time to ensure the correctness of the protocol. We have chosen the radius to be 3 hops as simulation results show that most channel reallocations which occur due to the addition of one new node to an already stable network end within 2 hops from the epicenter of change.

- Nodes always try to ensure that they get the resources they originally requested. Hence, nodes which receive a lower share than their initial demand try to find a channel which allows higher resources than what they are presently receiving. This might happen if a neighboring node reduces network demand or a neighboring node ceases to exist. In both cases, the node which detects an opportunity for increasing its available resources, reinitializes its channel access time to the maximum available capacity and starts a suppressed reallocation protocol.

4.4.5 Complete Reallocation at Regular Intervals

Multiple suppressed reallocations may eventually lead to sub-optimal results. In such a case, a complete reallocation becomes advisable. To decide upon the need for complete reallocation and to trigger one on

consensus is achieved by using a VOTE message. This is essentially a list of all nodes participating in the protocol and the channel access time they need to handle their clients in the present configuration. For this sub-protocol, nodes maintain three pieces of information. System performance threshold for reallocation, time interval between VOTE packet broadcasts and a limited history of system performance after every reallocation. At regular intervals of time, nodes broadcast these packets with their current throughput and their desired throughput. Each recipient combines the packets it has so far received and relays them. This ensures that within time proportional to the network diameter, all nodes know the total system performance. A history maintained of the system performance reveals how acceptable the current configuration is, and if it is below system performance threshold, any or all nodes start a complete reallocation algorithm. As our protocol is asynchronous, nodes need not start together. If repeated attempts of reallocation fail to improve the system performance, nodes learn that the system has asymptotically converged and all nodes change the system performance threshold accordingly.

An alternative, a two phase commit protocol, may avoid some reallocation rounds when system performance has asymptotically converged. However, that would require the invalid assumption that the system state remains constant between the initiation and the commit phase.

4.4.6 Versions of the Protocol

The Access Allocation Protocol decreases its load when it cannot fit its requirements on any channel. However, this might possibly lead to nodes decreasing their throughput because of transient capacity shortages[6]. Hence, we performed simulations with a different version of the protocol where nodes search for a channel starting from their maximum requirement each time they switch channel. Hence, in such a case, a node might $increase$ its throughput when it switches channel, a case not allowed by the basic version. We show the comparison in Section 7.6 and describe why this is not a good option.

In this Chapter, we described our main protocol which we claim attains all the goals we set for ourselves in Chapter 1. In the next Chapter we provide a support protocol so that Access Allocation can work properly under unreliable network conditions.

[6]This can happen because of a variety of reasons for example, microwave emissions, transient channel allocations etc.

Chapter 5

The Neighborhood Awareness Protocol

"Gather in your resources, rally all your faculties, marshal all your energies, focus all your capacities upon mastery of at least one field of endeavor."
-John Haggai

In this Chapter, we provide a secondary protocol which ensures all the nodes have the best information they can have under the given network conditions, so that they can run the Access Allocation Protocol to the best of their ability. The next Section describes our assumptions for this protocol. After that, we give the basic protocol overview and then detail each sub-protocol.

5.1 Assumptions

Our work is done keeping in mind the IEEE 802.11b system. We assume that we are using only the non-overlapping channels, thus avoiding cross channel interference. We also assume that bi-directional communication is possible, whenever one device can clearly receive the signal of the other device. In case one device cannot transmit as far as the other device, while it can receive the latter's transmissions, we assume that the interference does not result in significant bandwidth loss, and hence can be ignored.

5.2 Communication Model

We expect APs and stations within communication range to hear and understand the advertisements (regarding state information or for offering services) broadcast by each other. This includes cases when a station is not associated with the AP concerned, or vice versa. This is a realistic expectation as present devices respond to the $beacon$ broadcasts from other transceivers on the same channel. Presently, packets from a different service set are dropped at MAC layer once a station associates to a particular AP. In our protocol, we require that such packets be snooped at if they contain an advertisement. No other conditions are imposed on the communication model.

5.3 Foundation and Basic Structure

A logically separate version of this protocol, like AAP, is run at each node. The protocol can be separated into three basic functions: discovering neighboring stations and APs, classifying client stations into interference pattern groups, and keeping latest information about neighboring APs.

To carry out the functions listed above, the protocol maintains two data structures: *Neighbor List* and *Interference Table*. Neighbor List is used to keep the state information of the neighboring APs. This information is used to help the Access Allocation Protocol (discussed in Chapter 4) resolve channel conflicts. The other data structure Interference Table aims to provide an AP some topological understanding of its client stations and of other APs which interfere with its clients. Each AP maintains an individual Interference Table with information about which of its clients interferes with which neighboring AP. This allows each AP to run the Access Allocation Protocol by using different channel access time negotiations (dependent on the clients affected) with each of its neighboring APs.

5.4 Working of the Neighborhood Awareness Protocol

The Neighborhood Awareness Protocol is a data processing mechanism. Participating APs periodically broadcast their state information. If the receiving node is a station, then it simply archives the information and relays it to its own AP. When an AP hears about another AP directly, it incorporates that information into its neighbor list. When a station, say S, informs its AP, say A, about another AP, say B, then apart from the updating of the neighbor list structure, AP A also gains the knowledge that S is within the interference range of AP B. This knowledge is added to the interference table. We give here the pseudo-code for the working of the protocol at an AP 5-1 and a station 5-2.

In the next Section we give details of each module of the protocol.

5.5 Detailed Description

As discussed earlier, Neighborhood Awareness Protocol can be divided into three basic functions. We explain them in the following Sections.

5.5.1 Discovering Neighboring APs and stations

Standard routines of device discovery would allow us to discover neighboring devices in bidirectional communication range on the same channel. However, our protocol requires that each AP knows of all the devices in the system which satisfy one or more of the following criteria:

- APs in bi-directional communication range.
- Stations which are being serviced.
- APs which may interfere with client stations.

APs in Bi-directional Communication Range

APs which are on the same channel and within bi-directional communication range communicate using a protocol similar to ICMP Router Discovery [21]. The AP discovery messages are called “AP Advertisements” and “AP Solicitations”. Each AP periodically broadcasts an AP Advertisement on each of its hardware dependent channels. For this, APs periodically steal cycles from their existing channel where they service clients to other channels. (See Section 5.5.3 for protocol overhead analysis.) APs discover the attributes of the neighbor list structure (which is filled by information about their neighboring APs) in communication range by simply listening for advertisements. If (and only if) no advertisements are forthcoming, an AP may transmit an AP solicitation a small number of times, but then must desist from sending any more

$Neighborhood\_Awareness\_Protocol\_AP_i$

Signature:

Definition :
$state = \{\mathbb{I}, \mathbb{I}, \mathbb{I}, \mathbb{I}, array\}$ {state represents the quintuple which has represents the state of each node as required for implementing Access Allocation Protocol}
For every $k \in \mathbb{N}$: $array[k] \in \mathbb{I}$

Input :
$AP\_Advertisement(state\_info)_{j,i}, state\_info \in state$
$Advertisement\_Digest(message)_{j,i}, message = \{x_i \in state | i \in [1, ALL\_NEIGHBORS]\}$

Output :
$AP\_Bcast\_Advertisement(state\_info)_i, state\_info \in state$
$Channel\_Switch_i$

State:
For every $j \in \mathbb{I}$: $nbr\_list[j] \in state$

Transitions:

Input $AP\_Advertisement(state\_info)_{id,i}$
Effect: {An advertisement from another AP has arrived}
$update(nbr\_list[id], state\_info)$

Input $Advertisement\_Digest(message)_{id,i}$
Effect:
for all $k \in \mathbb{I}, state\_info_k \in message$ **do**
$update(nbr\_list[k], state\_info_k)$
$update(interference\_table[k], id)$ {Station id is in interference range of AP k}
end for

Output $AP\_Bcast\_Advertisement(state\_info)_i$
Precondition:
$idle\_flag = TRUE$ {Not servicing any station presently}
Effect:
$None$

Output $Channel\_Switch_i$
Precondition:
$idle\_flag = TRUE$ {Not servicing any station presently}
Effect:
$num\_transmit := num\_transmit + 1$

Figure 5-1: Pseudo-code for Neighborhood Awareness Protocol for an AP

$Neighborhood\_Awareness\_Protocol\_Station_i$

Signature:

Definition :
$state = \{\mathbb{I}, \mathbb{I}, \mathbb{I}, \mathbb{I}, array\}$ {state represents the quintuple which has represents the state of each node as required for implementing Access Allocation Protocol}
For every $k \in \mathbb{N}$: $array[k] \in \mathbb{I}$

Input :
$AP\_Advertisement(state\_info)_i, state\_info \in state$

Output :
$AP\_Solicitation_i$
$Advertisement\_Digest(message)_i, message = \{x_i \in state | i \in [1, ALL\_NEIGHBORS]\}$

State:
For every $j \in \mathbb{I}$: $message\_digest[j] \in state$

Transitions:

Input $AP\_Advertisement(state\_info)_{id,i}$
Effect: {An advertisement from another AP has arrived}
$update(message\_digest[id], state\_info)$

Output $Advertisement\_Digest(message)_{id,i}$
Precondition:
$idle\_flag = TRUE$ {Not servicing any connection presently}
Effect:
$None$

Output $AP\_Solicitation_i$
Precondition:
$idle\_flag = TRUE$ {Not servicing any connection presently}
$satisfied\_with\_present\_network\_condition = FALSE$
Effect:
$None$

Figure 5-2: Pseudo-code for Neighborhood Awareness Protocol for Stations

solicitations. Neighboring APs that were not discovered because of packet loss or temporary link partitioning, are eventually discovered by reception of their periodic (unsolicited) advertisements. (Links that suffer high packet loss rates or frequent partitioning are accommodated by increasing the rate of advertisements, rather than increasing the number of solicitations that hosts are permitted to send. The precise number of retransmissions depends upon the hardware concerned.)

Stations that are being serviced

Information about stations being serviced is easy to collect. This information does not go into the nbr_list table. Rather, it is used to classify client stations into interference pattern groups. Each station provides information about the APs within its communication range which share the same channel to the AP servicing it. stations also provide the AP Advertisements they hear from nodes on other channels to their own AP. For this, the same port is used with a special "Advertisement Digest" packet, which provides the messages the station had not yet reported to its AP. All stations flush their buffers periodically and whenever network conditions change drastically. An AP can also flush all messages by broadcasting "Channel Switch" message, thus declaring an intention to change channel.

APs that may interfere with clients stations

This information is automatically provided by client stations to the AP. Stations collect this by the AP Advertisements done by the interfering APs on the same channel. Apart from those APs which are already interfering, APs located nearby on different channels are also found out as those switch channels and send AP Advertisements on all channels. As a station chooses to join or remain with its AP, it still would be listening to the periodic AP Advertisements from the neighboring APs on the same channel. In the Personal Router project, stations (which double up as personal routers) are also allowed to send AP Solicitations on all channels. This helps in quick information collection about APs which may interfere, and also helps the station get all the services available in the region for Service Selection [22].

5.5.2 Classifying Client stations into Interference Pattern Groups

Each AP needs to classify its client stations on the basis of the APs they are interfering with. This is important because the term channel access time l described in Chapter 4 needs further refinement in final implementation. Negotiations for cooperation need to take place between AP A and AP B (as in figure 5-3) on only the amount of channel access time for each AP which is affected by the other AP. Thus, AP A and AP B can simultaneously receive from stations in the non-overlapping regions, while the interactions with the stations in the region of overlap are sequential and so is transmission by both APs. This means the coordination for channel access time should be based upon only that share of communication time which cannot be achieved simultaneously. This would allow us to get more system bandwidth.

Even though we share time according to interference patterns, it does not mean that our protocol requires inter AP MAC level coordination. Channel access is done by Distributed Coordination Function. Collisions might occur due to DCF; the share on interference patterns give us statistical share of channel and not any particular order or time for access.

Network Load l is now a quadruple $(nor_x, not_x, or_x, ot_x)$ referring to separate channel access times of non-overlapping reception from stations, non-overlapping transmission to stations, overlapping transmission and overlapping reception with respect to another AP X. Such a tuple is calculated for each neighboring AP and stored in the table *Interference Table* as part of traffic information to be reported to the respective

AP. However,

$$nor_x + not_x + or_x + ot_x = l, \forall x \in N \tag{5.1}$$

which is the same value as reported to all the neighbors. To measure its own channel access time in terms of

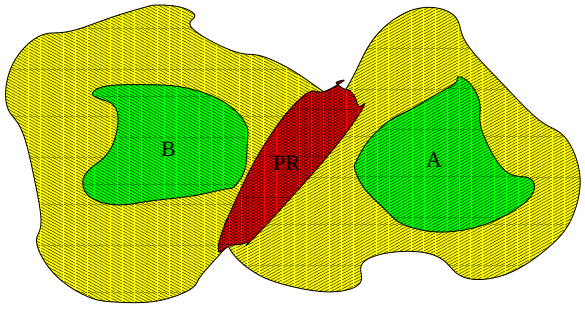

Figure 5-3: Overlapping APs

overlapping and non-overlapping zones with respect to another AP B, AP A needs to monitor the communication time with each of its stations separately. Also each station provides the information of the APs it can hear, and thus is interfered by, in the Advertisement Digests it sends to its AP. This information is then sufficient to determine the entries of the Interference Table.

Collecting this information about the stations an AP is serving, allows the AP to have an educated estimate on the amount of interference its stations would be facing once it switches its channel in case of a *Suppressed Channel Reallocation* Procedure. Please note that identifying interference groups helps to optimize the performance and if it is not done, it does not affect the correctness of the protocol.

5.5.3 Keeping Latest Information about Neighboring APs

APs which are in direct communication range update their nbr_list table as described in Chapter 4. However, now that we allow for relayed communication between APs because they may interfere with each other's client stations, some complications arise. In case the client stations which caused the two APs to coordinate have ceased functioning (because of switching off or moving away), the corresponding APs assume that their counterparts themselves have ceased operation. In that case, both APs delete the each-other's from their neighbor list data structure.

Stealing Time to Exchange Information

APs need to send control packets for maintaining neighborhood information in such a way that service disruption is minimal. This requires each AP to use a packet arrival prediction algorithm to estimate an interval of time when packets might be missed without significant loss to service. As we do not need a proper service profiling algorithm, but a simple traffic prediction scheme, a frequency histogram or a neural network trained on recent packet arrivals would be enough for the purpose. As a simplification, we use a frequency histogram for packet arrival prediction in this project.

Protocol Overhead Analysis

The bandwidth overhead of the Access Allocation Protocol and the subsidiary protocol described here is limited. Assuming that each AP has at most 10 APs surrounding it, the maximum message size would be 40 Bytes (assuming all terms in the quintuple to be representable by 4 Bytes each). If we allow for 250 nodes in the system with the worst case of channel allocation dependencies of each node on the rest, then we would arrive at a message size of 1KB assuming a 4 bytes AP Identifier used in disruption cost dependency. This is the theoretical maximum, but with limited channel dependencies (with dependency chains as long as 50 links), messages would be well within 256 Bytes. Total number of messages per node will be O(N) in the worst case, thus inflicting an average cost of less than 64KB per node per reallocation assuming 250 nodes. As the frequency of reallocation need not be very often (at most once every 15min), the overhead is further contained.

In this Chapter, we described a secondary protocol to ensure proper information is available to the Access Allocation Protocol described in chapter 4. This protocol ensures that AAP is robust even against unreliable network conditions.

Chapter 6

Formal Evaluation

"Proof is the idol before whom the pure mathematician tortures himself."
-Sir Arthur Eddington

In this Chapter, we provide formal proofs of the claims we made regarding our protocol. As mentioned earlier, our protocol seeks to satisfy the following conditions:

1. **Correctness:** A correct execution of the Access Allocation Protocol would assure that more channel access time is never allocated than is available.
2. **Local Maximal Performance:** Each Access point selects the best option for itself without affecting the remaining objectives.
3. **Global Maximal Performance:** The total system performance is maximized under the given execution sequence.
4. **Stability:** The system should not oscillate between a few solutions, but converge monotonically to one.
5. **Robustness:** The system should show graceful degradation of performance when facing deteriorating network conditions such as packets loss and link failure. (This objective cannot be evaluated formally.)

6.1 Definitions and Assumptions

Assumptions: In this Chapter, we assume reliable[1] and in-order message delivery from each node to the rest of the network. However, there is no given ordering of messages between any two pair of nodes, except when there exists a causal relationship (as in the case of a reply message). We also assume that within time Δ each node broadcasts its state at least once.

We give here some definitions which will be used later in the Chapter.

Definition - State: The state of a network (in this Chapter) is represented by a tuple (c, t) where c represents the channel of the AP and t represents the average time the station access the channel to communicate to the AP.

[1] We understand that this is not a realistic assumption. There is a scope of research work here to formally evaluate the protocol for unreliable message delivery. However, even though messages are not reliable, the same effect is achieved as message information is generally available through the Neighborhood Awareness Protocol discussed in Chapter 5.

Definition - Illegal State: A node is in illegal state when it is on a channel c which has been allocated to a set of nodes, the cumulative value of t of which exceeds the maximum permissible value.
Definition - Conflict: A conflict is defined by the states of the two nodes which are in illegal configuration. Thus two conflicts at different instants of time happening between the same two nodes are different if they have different state configuration. (We will later show that to be the case.)
Definition - Disruption Cost: The penalty the whole network has to pay when an AP switches from its present channel to the channel next in the order of preference is known as the disruption cost of that node.
Definition - Parent Node: When a node switches, all those nodes which have to also adjust to this new situation are the parent nodes of the node which switched.

6.2 Correctness

A guarantee that after the execution of the Access Allocation Protocol, no illegal states remain in the system, can only be given when nodes do not increase their $channel\_access\_time$ in one execution sequence of the protocol as in the pseudo-code module $Choose\_Channel$ (Figure 6-1). We assume that to be the case under consideration. We need to lay our ground to the proof of correctness by stating some auxiliary facts.

Lemma 1 *Within a time period Δ, one of the two nodes sharing a conflict will detect a conflict.*

Proof: Say the nodes are A and B. Without loss of generality, let us assume that node A needs to switch to resolve the conflict. Even if the knowledge of the state of node B is not available to the node A at time t, A knows the state of node B within time $t + \Delta$. Then node A has to switch channel or reduce load by the following pseudo-code.

Lemma 2 *Whenever an Access Point switches channel it does so to avoid a conflict.*

In the actual execution, an AP can also switch channel to avail itself of a higher channel access time. However, it is not possible to prove correctness for that version of the protocol.[2] Hence, we have assumed in this Chapter that APs never increase their channel access time.

Theorem 1 *When a conflict is detected, it is resolved by one round of successful message exchange among the nodes which are party to the conflict.*

Proof: A conflict can happen when an AP is switched on or when it switches from another channel where it was in an illegal state. Let there be an AP A which has chosen a channel on which it was not before. As soon as a node chooses a channel, it has to broadcast its presence on that channel, as pointed out in the pseudo-code module $Choose\_Channel$ (Figure 6-1). Assuming reliable message delivery, this message would eventually be processed by all neighbors and if there is any resulting illegal state, the node which detects this would take action to resolve the conflict, as pointed out in $Process\_Message$ (Figure 6-2).

Theorem 2 *Whenever an illegal state between two nodes A and B with given respective configuration (c, t_1) and (c, t_2) is resolved, a conflict with the same state configuration will not happen again.*

Proof: Resolution of a conflict entails decreasing of channel access time t or switching the channel c. The most suitable option is selected by the node detecting the conflict, and the correct decision for itself and its neighbors is then taken by it, as described in the pseudo-code module $Process\_Message$ (Figure 6-2).

[2] Indeed, in that case, incorrect configurations can happen which are then resolved when awareness of each other is gained by the nodes in question.

$ChooseChannel_i$

Precondition:
  $alive = TRUE$
Effect:
  $changed\_channel := changed\_channel + 1$
  if $changed\_channel > MAX\_CHANGED\_CHANNEL$ **then**
    $alive := FALSE$
  end if
  $required\_capacity := initial\_demand$ {For local performance optimization}
  $channel := channel\_with\_min\_utilization()$
  if $required\_capacity > channel\_capacity$ **then**
    $required\_capacity := channel\_capacity$
  end if
  $disruption\_cost := \sum_{j \in Nbr(i), Channel(j)=Channel(i)} disruption\_cost_j$
  $bcast(REQUEST, state_i)_{i,ALL}$

Figure 6-1: Pseudo-code for Choosing Next Channel

$Process\_Message_i$

Precondition:
  $alive = TRUE$
  $incoming\_messages > 0$ OR $complete\_reallocation = TRUE$
Effect:
  $changed\_channel := changed\_channel + 1$
  if $incoming\_messages = 0$ **then**
    $check\_for\_conflicts()$
    $return$
  end if
  $message_j = get\_from\_incoming\_queue()$
  $update\_neighbor\_information(message_j)$
  $update\_disruption\_cost(message_j)$
  if $mode(message_j) = REQUEST$ OR $destination(message_j) \neq i$ **then**
    if $channel_i \neq channel_j$ OR $\sum_{j \in Nbr(i), Channel(j)=Channel(i)} time_j + time_i \leq MAX\_CHANNEL\_ACCESS\_TIME$ **then**
      $bcast(YES, state_i)_{i,j}$
    else if $(disruption\_cost_i, id_i) \geq (disruption\_cost_j, id_j)$ **then**
      $bcast(NO, state_i)_{i,j}$
    else
      $bcast(YES, state_i)_{i,j}$
    end if
  end if

Figure 6-2: Pseudo-code for Processing Messages

Without loss of generality, let us assume that A has to switch channel or decrease required channel access time t_1 because $t_1 + t_2 > T$ where T is the maximum permissible channel access time. In case A has to decrease channel access time to t_1', we made the assumption above that it would never raise it again in an execution sequence of the algorithm. Hence, a conflict with (c, t_1) and (c, t_2) state configuration will not happen again as A would always have channel access time t_1' lower than t_1 in one pass of the algorithm.

In case A switches channel, it would still have a channel access time $t_1' \leq t_1$, depending upon the capacity of the channel it is switching to. In case A returns to channel c after some more switches, it would at maximum only take value $t_1' \leq T - t_2$. Thus again the same conflict does not arise.

Theorem 3 *The Channel Allocation Protocol converges to a solution without any conflicts within finite number of steps.*

Proof: Theorem 2 shows that no conflicts are repeated and we have assumed that channel access time never increases. Hence, the possibility of conflicts decreases as channel access time requirements of node decreases. As the channel access time for all APs is finite, the protocol exhausts all possible conflicts in finite number of steps when APs with illegal states switch or decrease their required channel access time.

6.3 Stability

The introduction of the concept of *disruption penalty* was primarily to stabilize the system and to converge quickly on a single solution of the allocation problem.

Claim 1 *No node is considered more than once in the calculation of disruption cost of another node.*

Proof: This is ensured as each node maintains a list containing the UIDs of all the nodes to whom its attributes its total disruption cost. When a node gets information about the lists of its parent nodes, it sifts out nodes which are repeated, thus ensuring the correctness of the above claim.

Lemma 3 *Eventually i.e. within time, no pair of nodes attribute their distribution cost to each other.*

Proof: If it happens and the number of hops between a pair of nodes is d, then they do not figure in each other's disruption cost metric within time $d\Delta$, where Δ, as mentioned earlier, is the maximum time period between two successive broadcasts of the state of an AP.

Lemma 4 *The distribution cost metric of nodes is a monotonically increasing term as the number of hops increases from the root of the tree.*

Proof: This is clear from the fact that disruption cost is always non-negative.

Lemma 5 *Whenever a challenge between two dependencies is resolved, one dependency graph is strengthened at the cost of other and then the challengers never cause a conflict again.*

Proof: By Lemma 4, as cost metrics are monotonic, when a tie is resolved at the expense of one of the contenders, the tie cannot happen again.

Lemma 6 *The system reaches its final state when all disruption cost dependencies are unchallenged.*

Proof: By Lemmae 3 and 5, in the worst case, $Diameter \times \Delta$ would be required for the resolution of all contentions, where $Diameter$ is the diameter of the graph of all APs in an area. No more contentions would then result and the system would have reached its final state.

6.4 Maximal Performance

Claim 2 *The problem of maximization of system throughput on c channels and n APs each with discrete demands t_i when the nodes are connected as in graph G, is NP complete.*

Proof: Attaining maximum performance out of the network is harder than graph coloring itself. But given any performance level, it can be verified in linear time whether a given solution achieves better than that or not. This can be iterated for all achievable performance values. As such a verification can be done in polynomial time, the problem at hand is NP complete and not intractable. Yet, as the problem cannot be solved by any polynomial approach, our solution cannot guarantee maximal performance but only follow the greedy heuristic to converge to the best solution.

Lemma 7 *Each node chooses the maximum channel access time it can get under the given execution sequence.*

Proof: Whenever a node switches channel to avoid a conflict, it chooses a new channel which provides it maximum channel access time it can get given that it does not increase its requirements during the execution of a total reallocation sequence. This is clear from $choose\_channel$. However, after complete reallocation ends, nodes check the status of their neighbors at random intervals of time and claim higher access time if it is available on any channel as shown in $choose\_channel$. This allows us to make the following claim.

Claim 3 *No node can switch its channel and get higher channel access time without disrupting other nodes.*

We can only claim maximal performance as we have made a trade-off between stability and performance. As the exact execution sequence is decided by the random order of protocol messages between member nodes, on the basis of which message reached first, dependency trees get formed in the system. In cases when the topology is dense and conflicts occur, a node which has high throughput maybe chosen to be eliminated or reduced in strength in the interest of stability of the system.

This Chapter furnished formal evaluation of the correctness, stability and maximal throughput directive of our algorithm.

Chapter 7

Simulation Results

"Get the facts, or the facts will get you. And when you get them, get them right, or they will get you wrong."
-Dr. Thomas Fuller

The overall goal of the experiments described in this Chapter, is to measure the ability of AAP (Access Allocation Protocol) to achieve the stated objectives. As mentioned earlier, our protocol seeks to satisfy the following goals:

1. **Correctness:** A correct execution of the Access Allocation Protocol would assure that more channel access time is never allocated than is available.
2. **Local Maximal Performance:** Each Access point selects the best option for itself without affecting the remaining objectives.
3. **Global Maximal Performance:** The total system performance is maximized under the given execution sequence.
4. **Stability:** The system should not oscillate between a few solutions, but converge monotonically to one.
5. **Robustness:** The system should show graceful degradation of performance when facing deteriorating network conditions such as packets loss and link failure.

7.1 Evaluation Model

Access Points in the simulation are randomly placed in a 600×800m space. The range of each Access Point is 75m unless otherwise stated. We understand that the graph thus obtained might not provide a correct model of the real topologies, because of the presence of a variety of interference patterns and fading phenomenon affecting the real situation. Our model is simple and abstracts away characteristics of any particular communication system, and it allows for intuitive understanding of the protocol for a given topology of Access Points. A more accurate model will create links differently among nodes, which will create a different graph. Our experiments are done with random graphs created by placing specified number of nodes in the given area where the ordinates and abscissa are chosen with equal probability form over the whole interval. We believe that such a model should be able to capture all types of graphs. There is a scope for further research here by applying our algorithm on accurate models of various communication systems, especially 802.11.

All nodes communicate by broadcasting their messages, and follow the IEEE 802.11 MAC Model. We assume that protocol messages fit within one Maximum Transmission Unit[1] (this is a valid assumption for systems up to 375 Access Points in Size). More than one frame will be required in case the number of nodes in the system is over 375 and all of them participate in creating a dependency graph for frequency allocation of a node. Our simulation assumes 4 channels, each with a capacity of 1000 units. Each Access Point is randomly assigned an initial requirement of channel capacity, in a range of [0,1000]. We assume IEEE 802.11 Communication Model with random exponential backoff and limited retransmits, 7 in number.

7.2 Simulation Methodology

We have developed an event driven distributed message passing simulation system for testing our algorithm. Each node in the distributed system corresponds to an AP. Links between the access points have two different propagation delays - δ when there is direct communication, and $2.\delta$ when the communication is through a station in the real environment. The choice between the two is done by assuming both situations can happen with equal probability.

As mentioned earlier, we have implemented the IEEE 802.11 MAC into our system so that we can emulate collisions, retransmissions and packet drops which happen in a real communication system.

We run our simulations for both the versions of the protocol (as discussed in Section 4.4.6) for a maximum of 500 nodes. In other words, we have run the complete reallocation mode of the algorithm for populations of node ranging from 1 to 500. The simulation stops when all nodes vote that they have reached a stable state where no one needs to switch anymore. All results are reported for this stable state. The following Sections describe our findings.

7.3 Performance

We test our protocol in a variety of conditions. One of the main criteria of the success of the protocol is the overall system performance. Figure 7-1 shows the overall throughput achieved from the system with the number of nodes being on abscissa. As can be seen, the system reaches a saturation level around 95000 units. After that, the increase in system utilization by adding a new node is offset by the increasing length of dependency chains and hence suboptimal allocation of resources in the network.

An example case is shown in figure 7-2.

Let there be only one channel in the network. As long as APs B and C are not linked, they can both provide high channel access time to their stations. Now a new AP is injected in the system between B and C, so that A can communicate with both B and C. If A has higher system utilization than each of B and C respectively or if A combined with nodes on the other side of A represent a higher disruption cost than B and C respectively then B and C decrease their access requirements. This however might lead to lower overall system throughput. Clearly, this happens only when nodes do not have the choice of switching to another channel and their own frequency has too many APs. As more nodes are added to the system, more such suboptimal allocations happen, bringing down total system performance.

Indeed, figure 7-3 shows the number of Access Points which can be removed ("Dead" nodes) from their location without decreasing performance. There are no dead nodes for the first 100 nodes, however, as the number of nodes increases, not only the number of dead nodes, but also the ratio of dead nodes to live nodes increases.

[1]The largest physical packet size, measured in bytes, that a network can transmit. Any messages larger than the MTU are divided into smaller packets before being sent. Most network administrators specify the MTU of their network to be 1500 bytes.

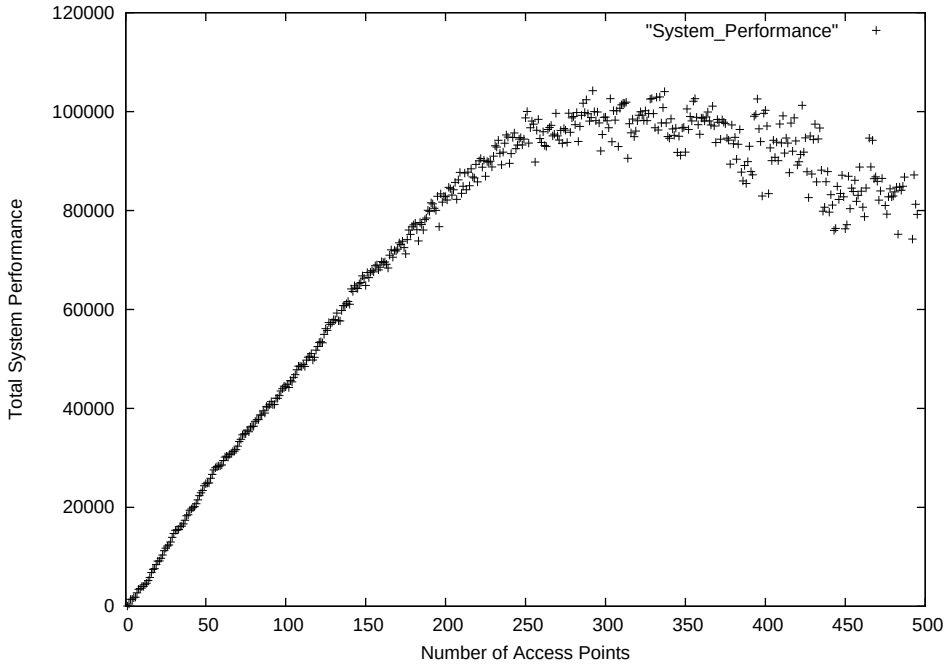

Figure 7-1: Overall System Performance against Number of Nodes

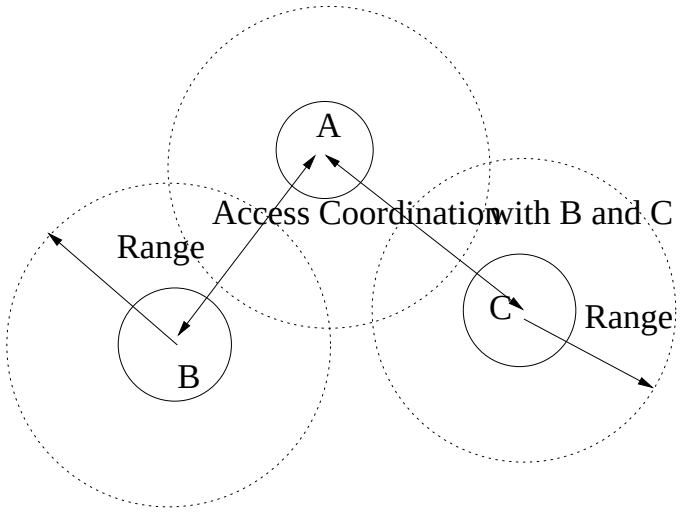

Figure 7-2: An Example Case of Sub-Optimal Resource Allocation

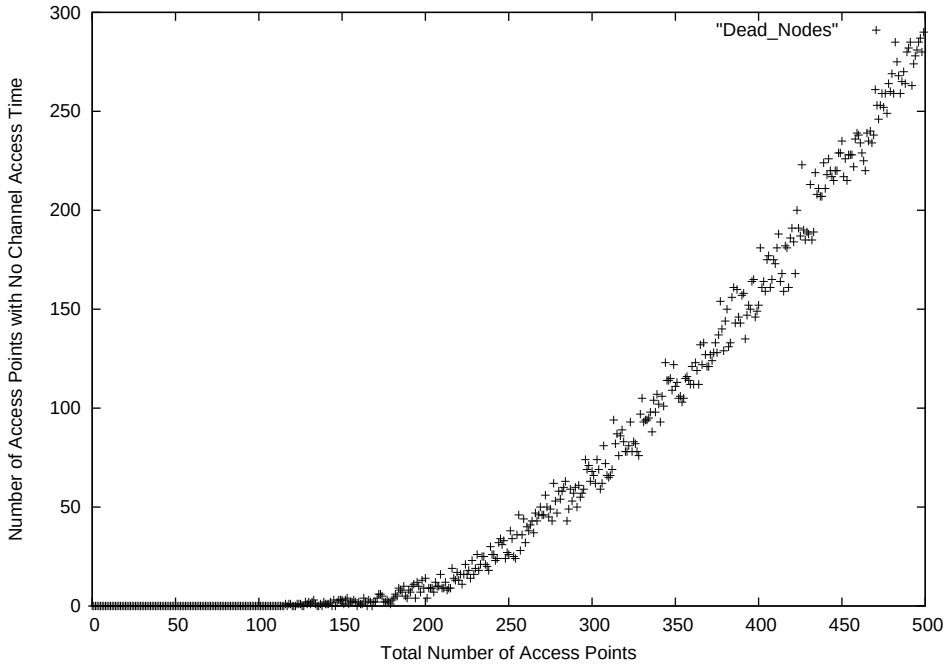

Figure 7-3: Number of Nodes without Channel Allocation against Total Nodes

As number of nodes increases, the probability of nodes getting the bandwidth they seek decreases. The ratio of Available vs Required bandwidth is given in figure 7-4. If required bandwidth can be achieved, naturally that would be the case of maximum performance. Indeed, our protocol provides a performance ratio of 0.9 and more until significant number of nodes start “dying” (until 200 nodes). There is a linear decrease in the ratio after some bandwidth is achieved (approx. 50% of saturation bandwidth. This is also the level when nodes start ”dying” (after a population of 125 nodes in the example system). Figure 7-5 shows the protocol communication overhead in packets. The linear curve shows that the protocol has an average message complexity of $O(n)$.

7.4 Robustness

The protocol keeps on updating information about neighbors as it gets them and acts on them without past memory. Hence, the protocol attempts to ensure correctness with the information it has, and can give correct results as long as messages are eventually transmitted from the node which changed state to the other nodes. As reliable message delivery cannot be guaranteed under our set of assumptions, so we cannot guarantee correctness as well. However, the algorithm works correctly under the assumption of guaranteed delivery of messages.

Hence, some configurations remain unresolved, where some channels are over-allocated (we call such distributions *illegal*) . However, we claim that as the packet drop rate increases, our protocol shows a graceful increase in the number of unresolved clashes. Figure 7-6 compares Packet Loss Rate with Number of Illegal States in the system after the protocol runs once. As can be concluded from the figure, our protocol is extremely robust against packet loss. This is due to the decision making procedure of nodes on the basis of the cache of information maintained by them about their neighbors.

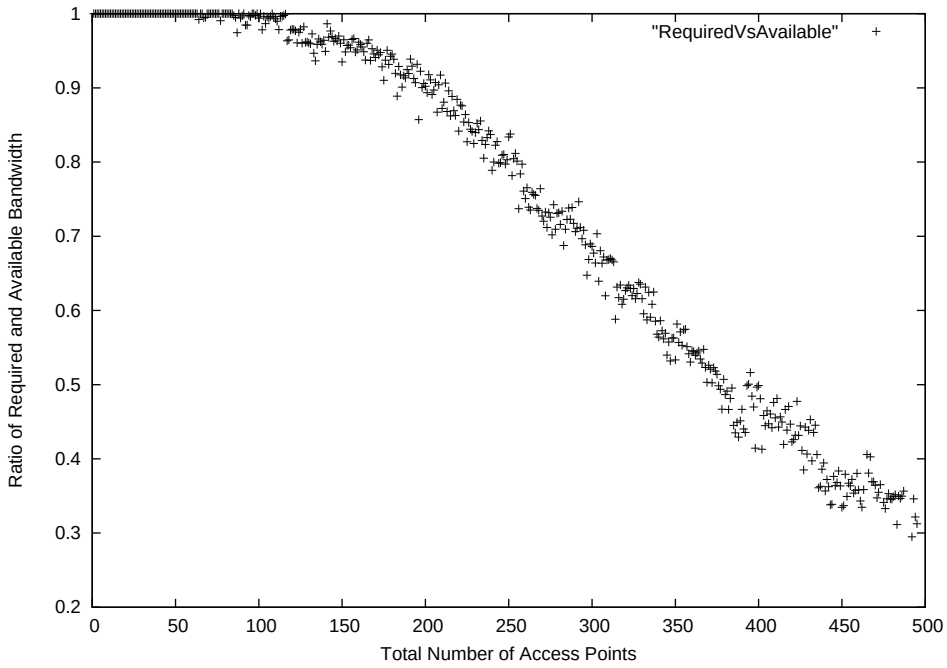

Figure 7-4: Ratio of Available vs Required Bandwidth against Total Nodes

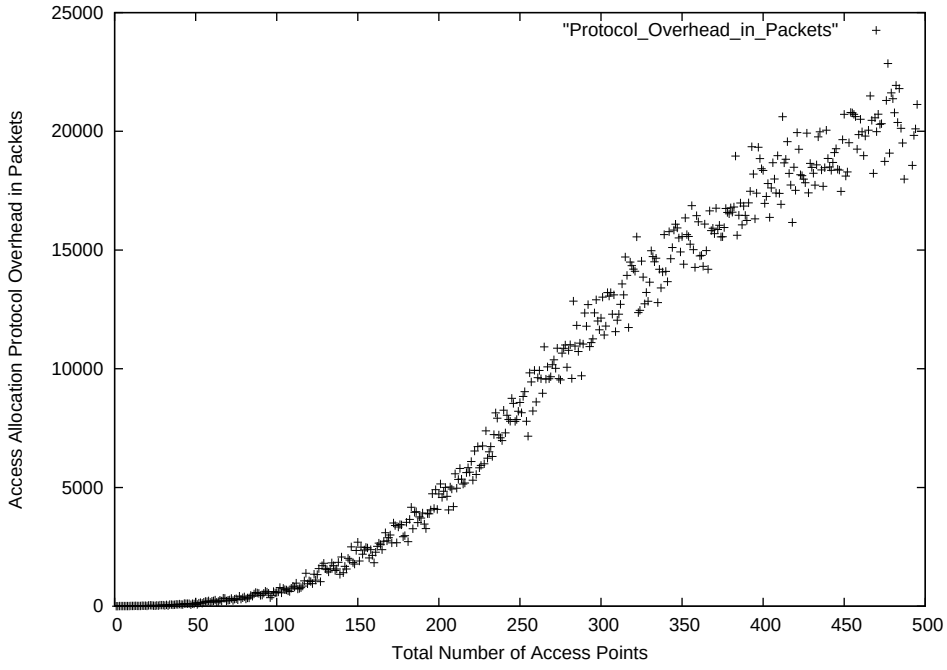

Figure 7-5: Protocol Communication Overhead against Total Nodes

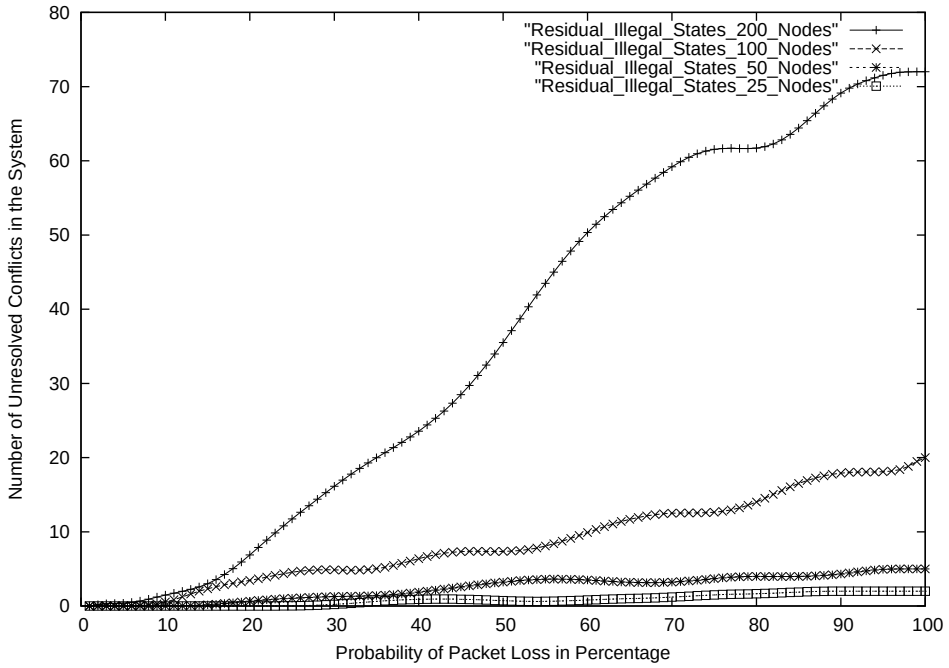

Figure 7-6: Number of Residual Illegal States against Packet Loss Ratio

The protocol can handle increasing density of Access Points. We have considered this case under 0 packet loss rate (packets however maybe dropped if they timeout). Our simulations showed that sometimes 1-2 illegal configurations remain unresolved. Figure 7-7 compares the number of initial illegal configurations in the system and final state of the network after a run of the AAP. Figure 7-8 shows only the final illegal states for better visualization.

7.5 Stability

To avoid transient changes, our protocol attempts to dampen channel reallocations once the whole network has been configured. For this we depend upon the disruption cost mechanism plus a 3 hop inhibitor mechanism. With the network system, higher degree sub-graph get formed which have a relatively high disruption cost due to channel switching dependency relationships between nodes. Such nodes would not switch channels until a sizable number of nodes change states, thus impeding proliferation of the effects of channel switches done by a few nodes. Also, if other nodes do not need to switch channels if a sub-graph reallocates shared medium, i.e. the disruption faced by the remaining network due to reallocation within this sub-graph is 0, the sub-graph can reallocate channel without ramifications. In the worst case, if the effect goes beyond 3 hops, the 3rd hop Access Point suppresses the messages, and thus the protocol is run within this subgraph of 6 nodes diameter (Suppressed Reallocation Mode).

The average number of nodes which are perturbed when one new node appears in their vicinity is completely dependent on the neighborhood where the node is placed. We observe from simulations that the average hop distance of perturbation is 1 for our scenario.

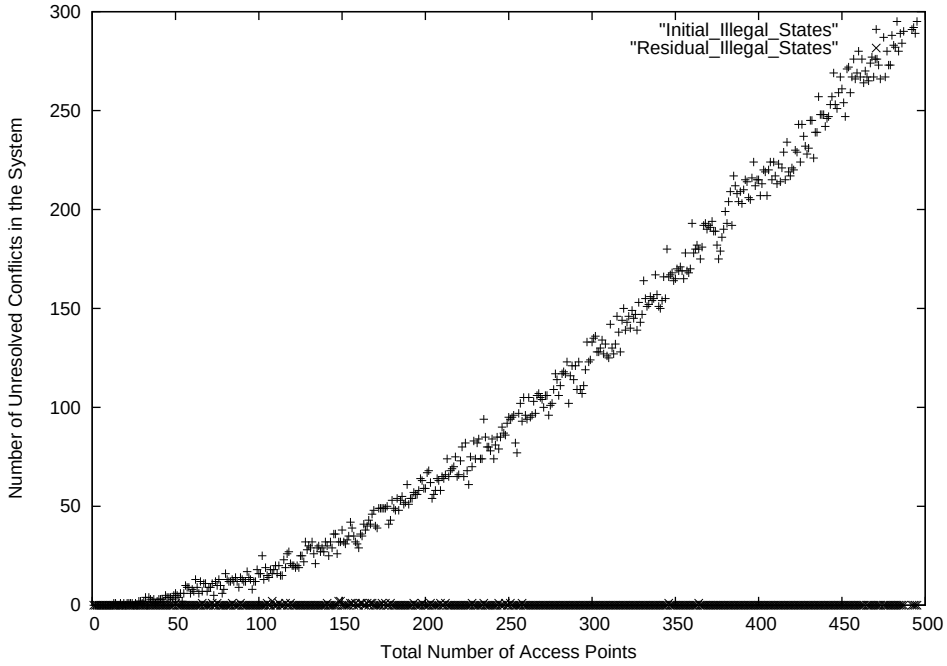

Figure 7-7: Number of Initial and Final Illegal States against Number of Nodes

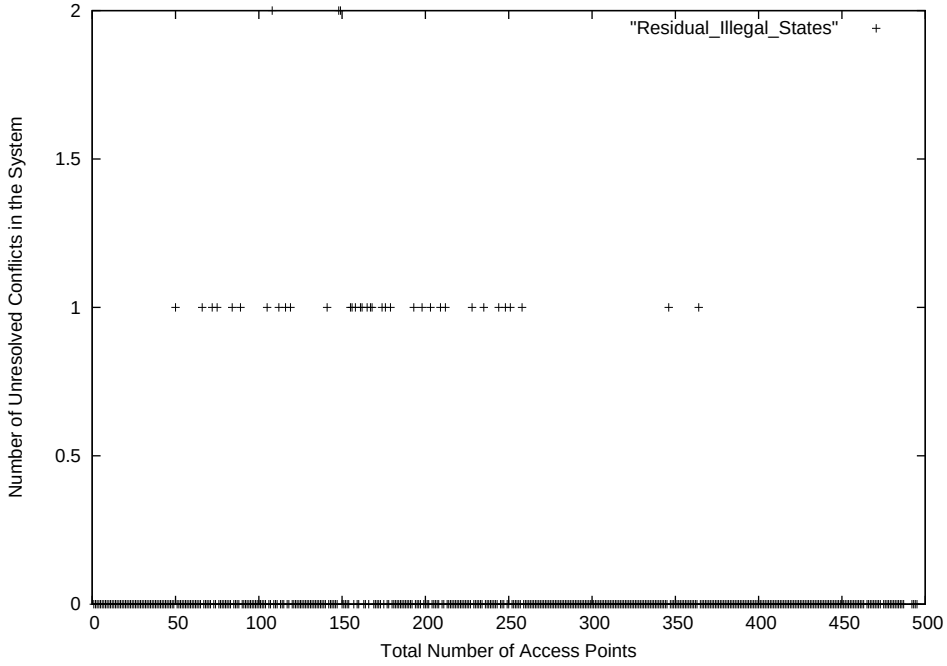

Figure 7-8: Number of Final Illegal States against Number of Nodes

7.6 Comparison between Stages of Access Allocation Protocol

As described in Chapter 4, the Access Allocation Protocol decreases the permitted load of the AP concerned when it cannot fit its requirements on any channel. However, this might possibly lead to nodes decreasing their throughput because of transient capacity deficiencies which can happen in many situations such as short-time microwave emissions, or when a node switches to a channel, forcing others to decrease load and then leaves itself for a better option. Hence, we performed simulations with a different version of the protocol where nodes search for a channel starting from their maximum requirement each time they switch channel. Hence, in such a case, a node might $increase$ its throughput when it switches channel, a case not allowed by the basic version. Figure 7-9 does show a minor increase in overall system performance,

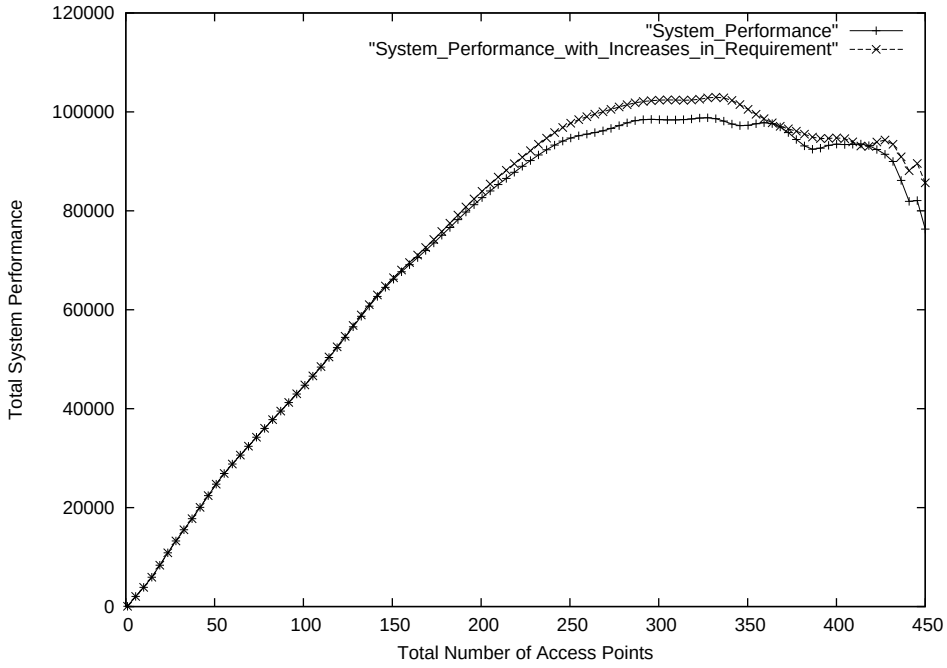

Figure 7-9: Comparison of System Performance for the Two Versions of the Access Allocation Protocol

however this comes at a premium, as now nodes might end up in conflicts. This is because APs have now chosen to be memoryless regarding the requirements of their neighbors, and reconcile only when they again get new information. Figure 7-10 compares the final illegal states in the system. It is clear that both versions are comparable in this regard. However, under the assumption of reliable message delivery, the algorithm with no increase in throughput at each switch during complete reallocation provides the maximal throughput with provably no conflicts in the system. This will be shown in Chapter 6.

The experiments done in this Chapter show that our algorithm is correct (because it gives minimal illegal states with unreliable message delivery), robust (shows a graceful increase in number of conflicts with increase in packet loss), stable (converges to a solution where nodes are not perturbed to more than 3 hops in partial reallocations) and maximal (provides excellent system performance without unresolved conflicts).

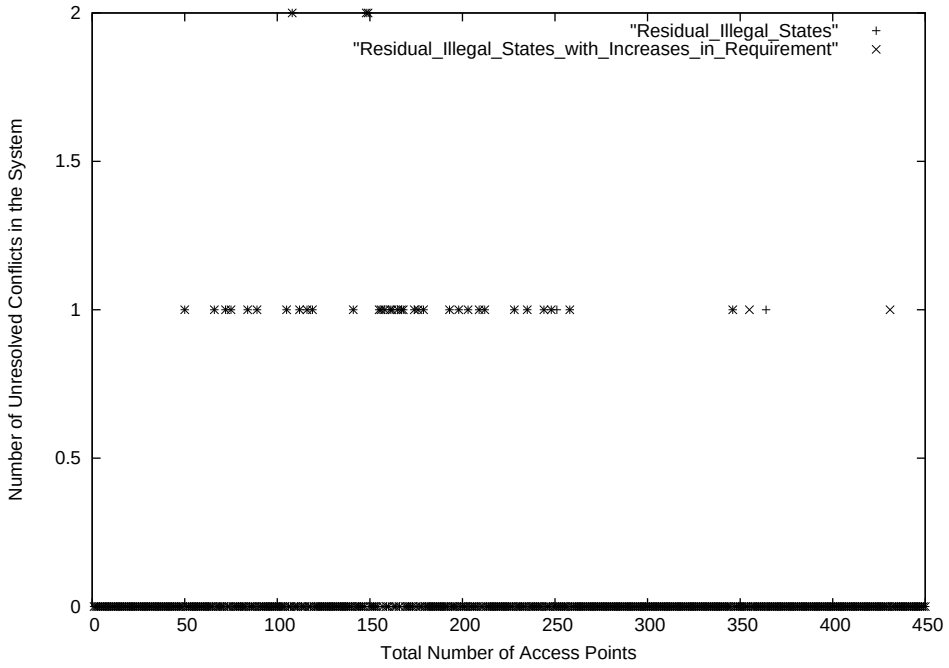

Figure 7-10: Number of Residual Illegal States for the Two Versions of the Access Allocation Protocol

Chapter 8

Performance Analysis for Multiple Colocated BSS

"We must never assume that which is incapable of proof."
-G. H. Lewes

In this Chapter, we analyze the throughput and goodput we can actually achieve for an environment where multiple APs are working to serve their clients. We detail a MAC level performance analysis below, first for two APs and then for a general case.

Besides helping APs decide what QoS can be assured to clients, our analysis vindicates the motivation for the Access Allocation Protocol: that without proper coordination, achievable system throughput falls sharply as the population of devices increases. By allocating channels well, we can reduce the effect of this phenomenon.

8.1 Throughput Analysis of IEEE 802.11 Protocol: Two APs

This Section aims to provide an analytical model to compute the *saturation throughput* for a simpler case of two APs, i.e. the condition when each station always has a packet to transfer. We derive our figures under the following assumptions:

- We ignore the effect of frame error due to bit errors induced by channel noise. Therefore frames are received in error (and without chance of recovery) only when they encounter collisions due to other simultaneous transmissions[1].
- We assume that the network consists of a finite and fixed number of APs and client stations. In real situations, when this is not true, the figure can be replaced by the average number of stations that contend for the channel at any time instant.
- We consider hidden stations under the assumption that when a station S_1 is heard by another station S_2, then it is also heard by the AP serving S_2. In reality this is not true. To address this, we make an imaginary mirror node for each node S_1 so that we can derive correct performance estimates. Hence the number of nodes each BSS has is a sum of real and such shadow nodes. The total system

[1]For actual system throughput on a particular frequency, the Bit Error Rate and Loss Model for that system should be added to the analysis presented below.

throughput thus calculated is then multiplied by the fraction of real nodes over sum of real and shadow nodes to get the real system throughput.

- We assume that the collision probability of a transmitted frame is constant and independent of the number of re-transmissions that this frame has experienced in the past.

Let there be two APs A and B in an area (Fig. 8-1) with n_A and n_B stations connected to them out of which a fraction of o_A and o_B nodes belong to an area (called the *overlap zone*) which is within the range of both APs. Hence, a successful transmission from the overlap zone requires $n_A + n_B - 1$ nodes to be quiet with a node transmitting, while a successful transmission from the non-overlap zone requires only the nodes in that BSS to be quiet with a node from that BSS transmitting. Let τ represent the probability that a station transmits in a randomly chosen time slot. τ thus decides the channel access requirements of the station concerned. The probability P_s that a transmission by a member of the BSS A is successful is given by the probability that exactly one station out of the two BSS transmits on the channel for the overlap zone nodes and exactly one station out of the that group transmits on the channel for the non-overlapping nodes.

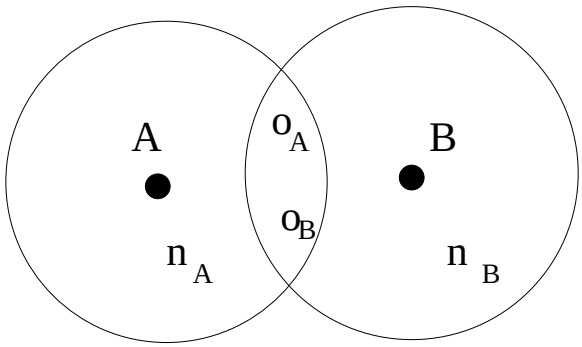

Figure 8-1: A System with Two Overlapping Access Points

$$\begin{aligned} P_s &= (\frac{o_A}{n_A})\tau(1-\tau)^{n_A+n_B-1} + (1-\frac{o_A}{n_A})\tau(1-\tau)^{n_A+o_B-1} \\ &= \tau(1-\tau)^{n_A-1}[\frac{o_A}{n_A}(1-\tau)^{n_B} + (1-\frac{o_A}{n_A})(1-\tau)^{o_B}] \end{aligned} \tag{8.1}$$

Let n_i represent the total number of nodes in another BSS i which is overlapping partially by BSS A, and let o_{Ai} represent the nodes in the given BSS A which are overlapped by BSS i. Generalizing equation 8.1 for m APs neighboring each AP, we get:

$$P_s = \tau(1-\tau)^{n-1}[\sum_{i=1,i\neq A}^{i=m}(\frac{o_{Ai}}{n_{Ai}}(1-\tau)^{n_i}) + (1-\sum_{i=1,i\neq A}^{i=m}\frac{o_{Ai}}{n_{Ai}})(1-\tau)^{\sum_{i=1,i\neq A}^{i=m} o_{iA}})] \tag{8.2}$$

8.2 Throughput Analysis of IEEE 802.11 Protocol: Multiple APs

In a general setting, let each AP cover an area D, and let the stations be uniformly distributed with a density ρ. Then the number of stations in the range of an AP is $n = D\rho$. If we assume the average area of

overlap between two nodes is a fraction c of the total area D, then the number of stations of each BSS in overlap area with another BSS is $cD\rho$. Equation 8.1 then can be rewritten as:

$$\begin{aligned} P_s &= \tau(1-\tau)^{n-1}[\sum_{i=1,i\neq A}^{i=m}(cn(1-\tau)^n) + (n - \sum_{i=1,i\neq A}^{i=m} cn)(1-\tau)^{\sum_{i=1,i\neq A}^{i=m} cn}] \\ &= \tau(1-\tau)^{n-1}[mcn(1-\tau)^n + (n-mcn)(1-\tau)^{mcn}] \quad (8.3) \\ &= \tau(1-\tau)^{D\rho-1}[mcD\rho(1-\tau)^{D\rho} + (n-mcD\rho)(1-\tau)^{mcD\rho}] \quad (8.4) \end{aligned}$$

Figure 8-2 helps get an understanding of the function governing the probability of successful transmission. All the parameters used in our analytical model follow the parameters in paper [19] for DSSS and are summarized in table 8.1. Note that we assume the application data payload is 1000 bytes, IP header and UDP header are 20 and 8 bytes, so packet payload at MAC layer is 1028 bytes. The MAC Markov model equations discussed in appendix A are independent of the parameters. So the analysis of the appendix does not need to change at all for a different set of parameters. If parameters are changed, the analysis done in this Chapter also stands true; but in this case, the absolute values change.

Packet Payload	8224 bits
MAC header	224 bits
PHY header	192 bits
ACK	112 bits + PHY header
RTS	160 bits + PHY header
CTS	112 bits + PHY header
Channel bit rate	1 Mbps
Propagation delay	1 us
Slot time	20 us
SIFS	10 us
DIFS	50 us

Table 8.1: System Parameters for MAC and DSS physical layer

We can now express total throughput of each BSS as :

$$S = \frac{E[\text{payload information transmitted in a slot time}]}{E[\text{length of a slot time}]} \quad (8.5)$$

The average amount of payload information successfully transmitted in a slot time is $P_s E[P]$. Let P_q be the probability that all member nodes within range of BSS A remain quiet in the considered slot time, i.e.:

$$\begin{aligned} P_q &= 1-(1-\tau)^{n_A+o_B} \\ &= 1-(1-\tau)^{n_A+\sum_{i=1,i\neq A}^{i=m} o_{iA}} \\ &= 1-(1-\tau)^{n_A+mcA\rho} \quad (8.6) \end{aligned}$$

The average length of a slot time is obtained by considering that with probability P_q the slot time is free of any transmission, with probability P_s it contains a successful transmission, and with probability $1-P_q-P_s$,

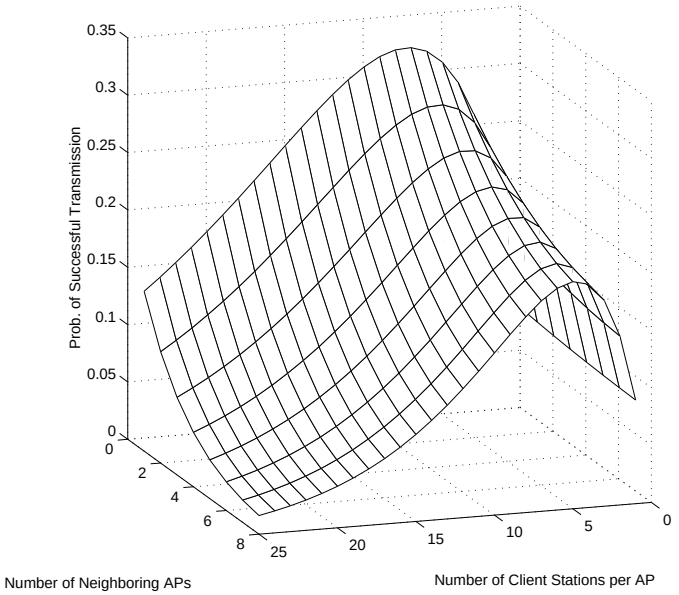

Figure 8-2: Probability of Successful Transmission for $c = 0.1, \tau = 0.05$

it contains a collision. Hence 8.5 becomes (Figure 8-3):

$$S = \frac{P_s E[P]}{P_q \sigma + P_s T_s + (1 - P_s - P_q) T_c} \tag{8.7}$$

Here, T_s is the average time the channel is sensed busy (i.e., the slot time lasts) because of a successful transmission, and T_c is the average time the channel is sensed busy by each station during a collision. σ is the duration of an empty slot time. To specifically compute the throughput for a given DCF access mechanism, is is now necessary only to specify the corresponding values T_S and T_C.

Let $H = PHY_{hdr} + MAC_{hdr}$ be packet header, and δ be the propagation delay. Here we will give the values only for RTS/CTS mechanism. For basic access and hybrid systems, the values of T_S and T_C can be found from [6]. For RTS/CTS Access Mechanism, collision can occur only in RTS frames (Fig. 8-4), hence

$$\begin{aligned} T_S^{rts} &= RTS + SIFS + \delta + CTS + SIFS \\ &\quad + \delta + H + E[P] + SIFS + \delta + ACK + DIFS + \delta \end{aligned} \tag{8.8}$$

$$T_C^{rts} = RTS + DIFS + \delta \tag{8.9}$$

A comprehensive analysis to provide the value of τ which considers frame retry limits as well has been

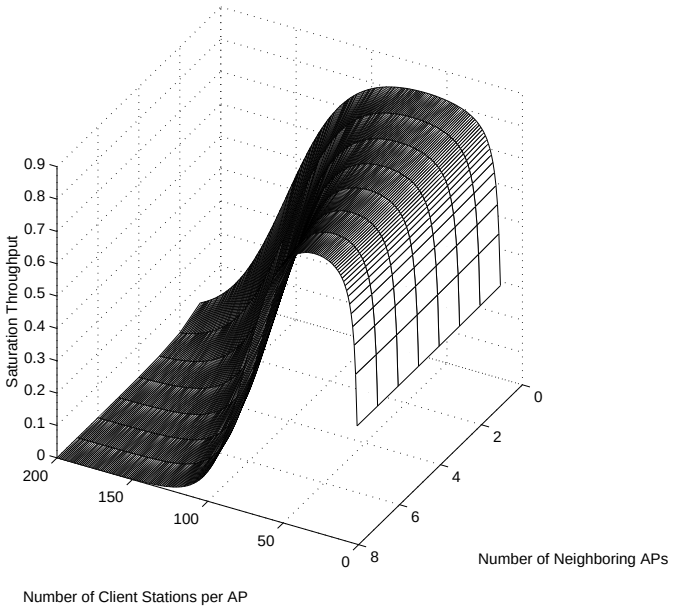

Figure 8-3: Saturation Throughput Curve for $c = 0.1, \tau = 0.05$

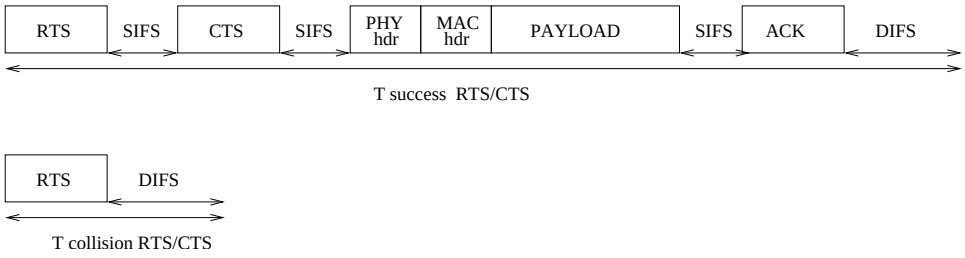

Figure 8-4: T_s and T_c for RTS/CTS mechanisms

provided in [65].

$$\tau = b_{0,0}.\frac{1-pm+1}{(1-p)} \tag{8.10}$$

$$p = 1 - \prod_{i=1, i\neq A}^{n} (1-\tau) \tag{8.11}$$

This results in a non-linear system that can be solved utilizing numerical methods and has a unique solution. We show in figure 8-5 how saturation throughput varies with τ and number of stations per AP.

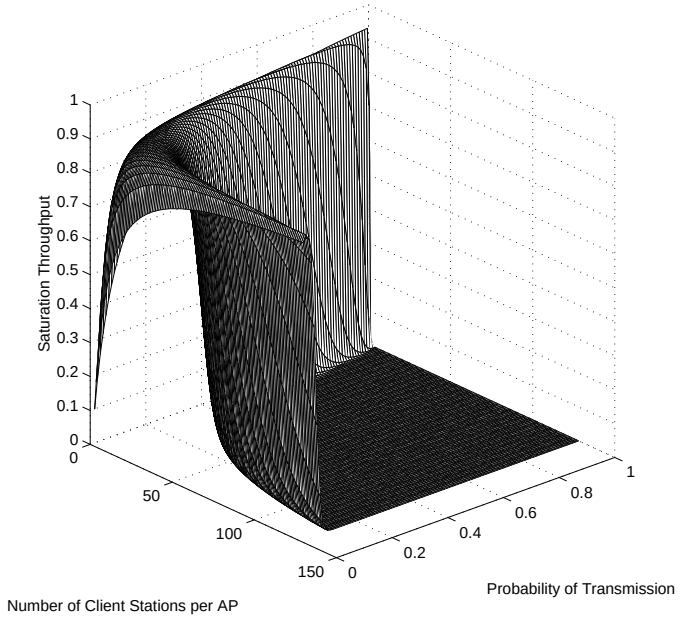

Figure 8-5: Saturation Throughput Curve for $c = 0.1, m = 2$

p is the probability that a transmitted frame encounters a collision, m is the maximum backoff stage and can have a value larger or smaller than m'. For DSSS (direct-sequence spread spectrum) physical layer in 802.11b, we have $m' = 5$, $b_{0,0}$ is a known function explained in appendix A, and σ is a the duration of an empty slot time. For maximum saturation throughput, τ should be approximately [6]:

$$\tau = \frac{1}{n\sqrt{\frac{T_c}{2.\sigma}}} \tag{8.12}$$

The analysis shown in this Chapter provides guidelines to decide the number of clients which can be serviced given the traffic requirements (represented by respective value of τ) of existing clients, number of neighboring access points and their clients. If a new client can be serviced, the analysis shows the channel access bounds for the new connection (given again by τ). This helps each AP to calculate the minimum goodput available to any station (which would happen in case of network saturation). Thus our analysis enables an AP to advertise an estimate of the QoS it can provide to new stations that may wish to join the AP.

Chapter 9

Protocol Implementation

"Life grants nothing to us mortals without hard work."
-Horace

In this Chapter we describe the implementation details of the protocol. We are currently implementing the protocol at the MAC layer. We have already developed an IP layer implementation. With the help of our protocol implementation, we plan to test our simulation results in real conditions. Our work will also serve as a prototype for the technical framework of the Personal Router Project.

9.1 Hardware Details

We use the commercial D-Link Wireless PCI Adapter (model DWL-520) [20]. The card implements IEEE 802.11b protocol and works within the 2.4GHz DSSS sanctioned for home and office environments. The specified operating range is 230 feet. We are attaching them to desktops in our laboratory. We plan to create a reasonable sized testbed, which will be useful not only for our future experiments, but also for other research work under the auspices of the Personal Router project.

9.2 Software Environment

We use the open source Host AP device driver [49] for our experiments. No changes in the device driver are required for IP layer execution of the Access Allocation Protocol. However, for a MAC level implementation, the driver is being rewritten.

For direct communication between two APs, we need to establish a bridge between them. This is done by using the open-source bridge utilities [10].

The operating system for the development platform is Redhat Linux 8.0 with kernel version 2.4.18-27.8.0 on i686 architecture. However, any platform with standard socket programming facility could be used as a platform.

9.3 IP Level Protocol Implementation

We employ simple IP datagrams for message passing between participating APs. Nodes broadcast "AP Advertisements" for a specific port address over a subnet (18.26.255.255 in our case). When stations have not heard an advertisement for a while and they need a new advertisement (because they are new on a channel or they plan to switch from their previous AP), they send an "AP Solicitation". This can repeated a

fixed number of times. The protocol logic in the implementation remains the same as described in chapter 4 and Chapter 5. Stations send advertisement digests to their AP and APs coordinate their access time with other APs by using the interference table and the neighbor list information. For more details, please see the pseudo-codes 5-1, 5-2,6-1,6-2.

9.4 MAC Level Protocol Implementation

We intend to have an implementation which is compatible with the IEEE 802.11 standard. This places significant restrictions and does not let us follow the most appropriate approach of adding extra information to beacons[1] of AP and Stations for AP Advertisements and Solicitations. Had we done that, our devices would be incompatible with network devices not following the Personal Router protocols, an undesirable situation which could hinder phased deployment of the PR project.

Hence, to do AP solicitations, devices (other APs or stations) have to associate with an AP and only then, can they get the advertisements. For AP Advertisements to be heard, we change the normal procedure of dropping packets of other networks by device drivers to a snoopy approach. This allows devices to hear broadcasts from other APs. AP broadcasts are sent as a different message than the MAC beacons, to ensure compatibility with the standard.

Except for the communication methodology, the protocol as such remains the same as in the IP level implementation.

If contacted, the authors will make the software available for free to interested readers.

[1] Beacons are used for device discovery in most communication networks.

Chapter 10

Conclusion and Future Work

"Try as hard as we may for perfection, the net result of our labors is an amazing variety of imperfectness. We are surprised at our own versatility in being able to fail in so many different ways."
-Samuel McChord Crothers

In this thesis, we present the technical framework to enable multiple co-located wireless access points to co-exist so that they can provide a variety of network services to the users in that region. We use channel access time as the means of resource allocation, while keeping the system modular so that any other unit can also be used without changing the protocol. We provide a secondary protocol to ensure robustness of our main algorithm for access allocation.

We also propose a model to measure transport level goodput with the help MAC level parameters. Although simple estimates based on current internet conditions can be made already, our work helps us estimate much more accurately the QoS which can be assured to clients by also taking into account the physical environment characteristics like number of other stations, APs and their clients. This is a result which has its independent benefits as well.

Our simulations show that our protocol is robust. We also provide formal proof that our algorithm is correct and that it quickly converges to a stable solution which provides maximal overall throughput.

To extend this work beyond this thesis, we would look into the case when the players of the protocol choose to act in self-interest. We would also like to build the prototype of the personal router in which all the relevant research done in our group would find its deployment.

We look forward to entrepreneurs utilizing our technical framework to make the Personal Router Vision a success.

Appendix A

System Performance Indicator Background

The analysis shown below lays the foundations for the work reported in chapter 8. For complete literature, please refer to [65].

Let $b(t)$ be the stochastic process representing the backoff time counter and $s(t)$ be the stochastic process representing the backoff stage for a given station at slot time t. the bidimensional process $\{b(t), s(t)\}$ can be modelled with a discrete-time Markov chain depicted in figure A-1.

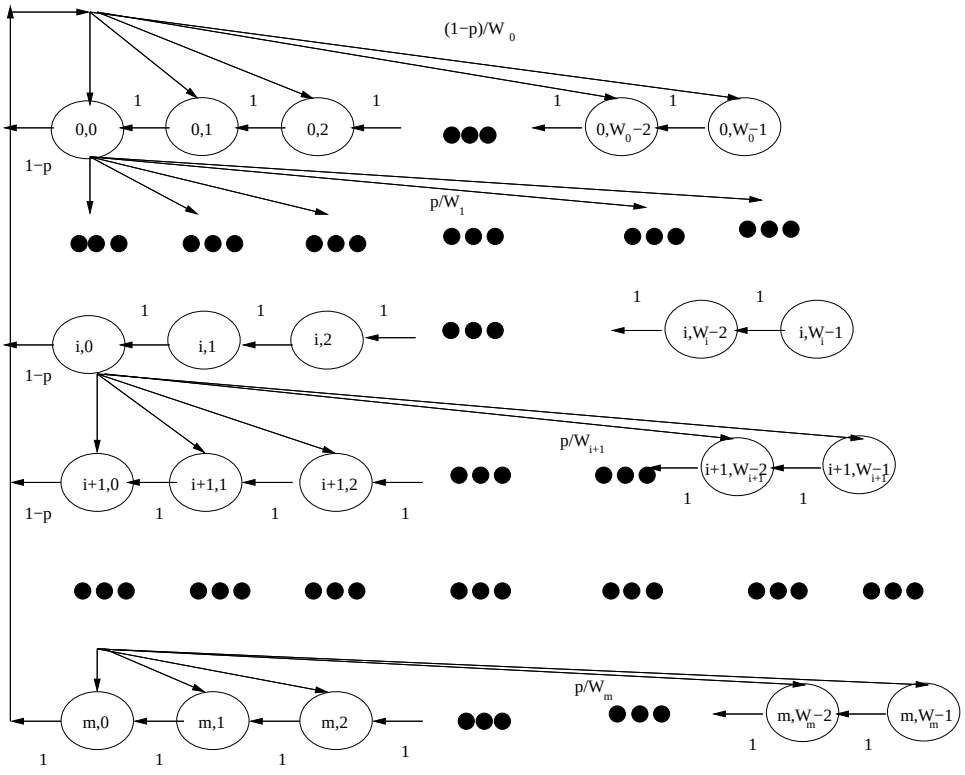

Figure A-1: Markov Chain Model

If $b_{i,k} = lim_{t\to\infty} P\{s(t) = i, b(t) = k\}$, where $i \in [0, m], k \in [0, W_l - 1]$ is the stationary distribution

of the Markov chain, then we can calculate the probability $b_{i,k}$. We have the following relations for b.

$$b_{i,0} = p.b_{i-1,0}, \quad 0 < i \leq m \tag{A.1}$$

$$= p^i.b_{i-1,0}, \quad 0 < i \leq m \tag{A.2}$$

Let m represents the station short retry count and is equal to 7 according to standard [47]. Here m is also the maximum backoff stage and can have a value larger or smaller than M. For the DSSS physical layer in 802.11b, we have M=5. Similarly the following relations represent the values of the contention window W:

$$W_i = 2^i.W, \quad i \leq M \tag{A.3}$$

$$= 2^M.W \quad i > M \tag{A.4}$$

As the chain is regular, for each $k \in [0, W_i - 1]$ we have:

$$\begin{aligned} b_{i,k} = & \quad \frac{W_i - k}{W_i}.(1-p).\sum_{j=0}^{m-1} b_{j,0} + b_{m,0} \quad , i = 0 \\ = & \quad \frac{W_i - k}{W_i}.p.b_{i-1,0} \quad , 0 < i \leq m \end{aligned} \tag{A.5}$$

Using A.2, A.5 can be simplified as:

$$b_{i,k} = \frac{W_i - k}{W_i}.b_{i,0} \quad , 0 \leq i \leq m \tag{A.6}$$

Equations A.2 and A.6 express all $b_{i,k}$ values as a function of $b_{0,0}$ and of collision probability p. If the normalization condition is imposed, we have:

$$1 = \sum_{k=0}^{W_i-1} \sum_{i=0}^{m} b_{i,k} = \sum_{i=0}^{m} b_{i.0} \sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} = \sum_{i=0}^{m} b_{i,0}.\frac{W_i + 1}{2} \tag{A.7}$$

By means of A.4, $b_{0,0}$ is given by A.8.

$$b_{0,0} = \begin{cases} \frac{2.(1-2p).(1-p)}{W.(1-(2p)^{m+1}).(1-p)+(1-2p).(1-p^{m+1})} & , m \leq M \\ \frac{2.(1-2p).(1-p)}{W.(1-(2p)^{M+1}).(1-p)+(1-2p).(1-p^{m+1})+W.2^M.p^{M+1}.(1-2p).(1-p^{m-M})} & , m > M \end{cases} \tag{A.8}$$

Now, we can represent transmission probability τ that a station transmits a frame in a randomly chosen slot time, in terms of $b_{0,0}$. As a station transmits only when the backoff counter reaches the value of zero, τ can be represented as:

$$\tau = \sum_{i=0}^{m} b_{i,0} = \sum_{i=0}^{m} p^i.b_{0,0} = b_{0,0}.\frac{1 - p^{m+1}}{(1-p)} \tag{A.9}$$

The probability p that a transmitted frame encounters a collision, is the probability that at least one of the $n-1$ remaining stations within interference range transmit in the same time slot. If we assume that all stations see the system in steady state and transmit with probability τ, the collision probability p is given by:

$$p = 1 - (1-\tau)^{n-1} \tag{A.10}$$

Equations A.9 and A.10 form a nonlinear system with two unknown τ and p. This can be solved utilizing

numerical methods and has a unique solution [6].

Bibliography

[1] I. Aad and C. Castelluccia. Introducing service differentiation into IEEE 802.11. In *Proceedings of the Fifth IEEE Symposium on Computers and Communications*, 2000.

[2] I. Aad and C. Castelluccia. Differentiation mechanisms for IEEE 802.11. In *Proceeding of IEEE INFOCOM*, pages 209–218, 2001.

[3] C. W. Ahn, C. G. Kang, and Y. Z. Cho. Soft reservation multiple access with priority assignment (srma/pa): a novel MAC protocol for QoS-guaranteed integrated services in mobile ad-hoc networks. In *IEEE Transactions on Vehicular Technology*, volume 2, pages 942–947, 2000.

[4] B. Berger and J. Rompel. A better performance guarantee for approximate graph coloring. In *Algorithmica*, volume 5, pages 459–466, 1990.

[5] V. Bhargavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LANs. In *ACM SIGCOMM Computer Communication*, volume 24, pages 212–225, 1994.

[6] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. In *IEEE Journal on Selected Areas in Communication*, volume 18, 2000.

[7] G. Bianchi, L. Fratta, and M. Oliveri. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. In *Proceedings of PIMRC*, pages 392–396, Taipei, Taiwan, 1996.

[8] F. Box. A heuristic technique for assigning frequencies to mobile radio nets. In *IEEE Transactions on Vehicular Technology*, volume 27, pages 57–64, 1978.

[9] D. Brelaz. New methods to color vertices of a graph. In *Communications of the ACM*, volume 22, pages 251–256, 1979.

[10] L. Buytenhek. Linux ethernet bridging. November 2001.

[11] F. Cali, M. Conti, and E. Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoritical limit. In *IEEE/ACM Transactions on Networking*, volume 8, 2000.

[12] M. Carlsson and M. Grindal. Automatic frequency assignment for cellular telephones using constraint satisfaction techniques. In *Proceedings of the Tenth International Conference on Logic Programming*, pages 647–663, 1993.

[13] M. Chams, A. Hertz, and D. De Werra. Some experiments with simulated annealing for coloring graphs. *European Journal of Operations Research*, 32:260–266, 1987.

[14] W. T. Chen, S. H. Chen, and J. C. Liu. An efficient QoS guaranteed MAC protocol in wireless ATM networks. In *Proceedings of ICOIN*, pages 785–792, 2001.

[15] W. T. Chen, T. Y. Fann, and W. T. Lin. A MAC protocol with quality of service guarantee for wireless ATM networks. In *Proceedings of ICOIN*, pages 105–110, 2001.

[16] W. T. Chen, B. B. Jian, and S. C. Lo. An adaptive retransmission scheme with QoS support for the IEEE 802.11 MAC enhancement. In *IEEE Transactions on Vehicular Technology*, volume 1, pages 70–74, 2002.

[17] H. S. Chhaya and S. Gupta. Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol. In *Wireless Networks*, volume 3, pages 217–234, 1997.

[18] D. D. Clark and J. Wroclawski. The personal router whitepaper. *MIT Technical Report*, 2000.

[19] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai. IEEE 802.11 wireless local area networks. In *IEEE Communication Magazine*, 1997.

[20] Inc. D. Link Systems. D-link systems inc. dwl-520 d-linkair 2.4ghz wireless pci adapter. 2003.

[21] S. Deering. ICMP router discovery messages. In *RFC 1256*, Xerox PARC, 1991.

[22] P. Faratin, J. Wroclawski, G. Lee, and S. Parsons. The Personal Router: An agent for wireless access. In *Proceedings of American Association of Artificial Intelligence Fall Symposium*, pages 13–21, 2002.

[23] N. Funabiki, N. Okutani, and S. Nishikawa. A three-stage heuristic combined neural network algorithm for channel assignment in cellular mobile systems. In *IEEE Transactions on Vehicular Technology*, volume 49, pages 397–403, March 2000.

[24] N. Funabiki and Y. Takefuji. A neural network parallel algorithm for channel assignment problems in cellular radio networks. In *IEEE Transactions on Vehicular Technology*, volume 41, pages 430–437, 1992.

[25] A. Gamst. Some lower bounds for a class of frequency assignment problems. In *IEEE Transactions on Vehicular Technology*, volume 35, pages 8–14, 1986.

[26] M. R. Garey and D. S. Johnson. Computer and intractability: A guide to the theory of np-completeness. W.H. Freeman, 1979.

[27] S. Garg, M. Kappes, and M. Mani. Wireless access server for quality of service and location based access control in 802.11 networks. In *Proceedings of International Symposium on Computers and Communications*, pages 819–824, 2002.

[28] G. R. Grimmet and C. J. H. McDiarmid. On colouring random graphs. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 77, pages 313–324, 1975.

[29] B. Hadzi-Velkov and B. Spasenovski. Capture effect in IEEE 802.11 basic service area under influence of rayleigh fading and near/far effect. In *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 1, pages 172–176, 2002.

[30] W. K. Hale. Frequency assignment: Theory and applications. In *Proceedings of the IEEE*, volume 68, pages 1497–1513, 1980.

[31] J. K. Hao, R. Dorne, and P. Galinier. Tabu search for frequency assignment in mobile radio networks. In *Journal of Heuristics*, volume 4, pages 47–62, 1998.

[32] A. Hertz and D. De Werra. Using tabu search techniques for graph coloring. In *Computing*, volume 39, pages 345–351, 1987.

[33] T. Ho and K. Chen. Performance analysis of IEEE 802.11 CSMA/CA medium access control protocol. In *Proceedings of PIMRC*, pages 407–411, 1996.

[34] K. C. Huang and K. C. Chen. Interference analysis of nonpersistent CSMA with hidden terminals in multicell wireless data networks. In *Proceedings IEEE PIMRC*, pages 907–911, September 1995.

[35] S. Hurley, D. H. Smith, and S. U. Thiel. FASoft: A system for discrete channel frequency assignment. In *Radio Science*, volume 32, pages 1921–1939, 1997.

[36] J. Jiang and T. H. Lai. An efficient approach to support QoS and bandwidth efficiency in high-speed mobile networks. In *International Conference on Communications*, volume 2, pages 980–984, 2000.

[37] D. S. Johnson, C. R. Aragon, L.A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation; part II, graph coloring and number partitioning. In *Operations Research*, volume 39, pages 378–406, 1991.

[38] A. Juhasz, F. Ulrich, B. Eged, and F. Kubinszky. Analysis of WaveLAN systems' performance. In *Hradstechnika*, April 2002.

[39] P. Karn. MACa: A new channel access method for packet radio. In *Proceedings of the 9th ARRL Computer Networking Conference*, Canada, 1990.

[40] K. Kim, S. Shin, and K. Kim. A novel MAC scheme for prioritized services in IEEE 802.11a wireless LAN. In *Proceedings of ICATM*, pages 196–199, 2001.

[41] P. Kim. Deterministic service guarantees in IEEE 802.12 networks–part I: the single-hub case. In *IEEE/ACM Transactions on Networking*, volume 6, pages 645–658, 1998.

[42] S. Kim and S. L. Kim. A two-phase algorithm for frequency assignment in cellular mobile systems. In *IEEE Transactions on Vehicular Technology*, volume 43, pages 542–548, 1994.

[43] L. Kleinrock and F. A. Tobagi. Packet switching in radio channels: Part I. In *IEEE Transactions on Communication*, volume 23, pages 1400–1416, 1975.

[44] L. Kucera. The greedy coloring is a bad probabilistic algorithm. In *Journal of Algorithms*, volume 12, pages 674–684, 1991.

[45] D. Kunz. Channel assignment for cellular radio using neural networks. In *IEEE Transactions on Vehicular Technology*, volume 40, pages 188–193, 1991.

[46] IEEE Local and Metropolitan Area Network Standards Committee. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. In *IEEE Std 802.11-1997*, New York, New York, 1997. The Institute of Electrical and Electronics Engineers.

[47] IEEE Local and Metropolitan Area Network Standards Committee. IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications. In *ISO/IEC 8802-11:1999(E)*, New York, New York, 1999. The Institute of Electrical and Electronics Engineers.

[48] IEEE Local and Metropolitan Area Network Standards Committee. Draft supplement to standard for telecommunications and information exchange between systems - LAN/MAN specific requirements - part 11: Wireless medium access control (MAC) and physical layer (PHY) specifications: Medium access control (MAC) enhancements for quality of service (QoS). In *IEEE Std 802.11-1999 Edition*, New York, New York, May 2002. The Institute of Electrical and Electronics Engineers.

[49] Jouni Malinen. Host ap driver for intersil prism2/2.5/3. 2003.

[50] D. G. Matula. Bounded color functions on graphs. In *Networks*, volume 2, pages 29–44, 1972.

[51] J. M. Peha. Spectrum management policy options. In *IEEE Communications Surveys*, 1998.

[52] R. J. Pennotti and R. R. Boorstyn. Channel assignment for cellular mobile telecommunications systems. In *Proceedings of National Telecommunications Conference*, pages 16:5–1–16:5–5, 1976.

[53] D. P. Satapathy and J. M. Peha. Etiquette modifications for unlicensed spectrum: Approach and impact. In *IEEE Transactions on Vehicular Technology*, 1998.

[54] D. P. Satapathy and J. M. Peha. A novel co-existence algorithm for unlicensed fixed power devices. In *Proceedings of IEEE Wireless Communications and Networking Conference*, 2000.

[55] S. Sharma, K. Gopalanr, N. Zhu, P. De, G. Peng, and T. C. Chiueh. Quality of service guarantee on 802.11 networks. In *Hot Interconnects*, volume 9, pages 99–103, 2001.

[56] S. T. Sheu and T. F. Sheu. DBASE: A distributed bandwidth allocation/sharing/extension protocol for multimedia over IEEE 802.11 ad hoc wireless LAN. In *Proceedings of IEEE INFOCOM*, volume 3, pages 1558–1567, 2001.

[57] K. N. Sivarajan, R. J. McEliece, and J. W. Ketchum. Channel assignment in cellular radio. In *IEEE Transactions on Vehicular Technology*, pages 846–850, 1989.

[58] D. H. Smith, S. Hurley, and S. U. Thiel. Improving heuristics for the frequency assignment problem. In *European Journal of Operational Research*, pages 76–86, 1998.

[59] H. L. Tzeng and C. Chen. Performance comparison of two MAC protocols for wireless ATM networks. In *Proceedings of International Conference on Information Networking*, pages 805–810, 2001.

[60] J. Weinmiller, H. Woesner, J.P. Ebert, and A Wolisz. Analysing the RTS/CTS mechanism in the dfwMAC media access protocol for wireless LANs. In *IFIP TC6 Workshop on Personal Wireless Communications*, Czech Republic, 1995.

[61] J. Weinmiller, H. Woesner, J.P. Ebert, and A Wolisz. Analysing and tuning the distributed coordination function in the IEEE 802.11 DFWMAC draft standard. In *Proceedings of MASCOT*, 1996.

[62] D. J. A. Welsh and M. B. Powell. An upper bound on the chromatic number of a graph and its applications to timetabling problems. In *Computer Journal*, volume 10, pages 85–86, 1967.

[63] A. Wigderson. Improving the performance guarantee of approximate graph coloring. In *Journal of the Association for Computing Machinery*, volume 30, pages 729–735, 1983.

[64] D. C. Wood. A technique for coloring a graph applicable to large scale time-tabling problems. In *The Computer Journal*, volume 3, pages 317–319, 1969.

[65] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma. Performance of reliable transport protocol over IEEE 802.11 wireless LAN: Analysis and enhancement. In *Proceedings of IEEE INFOCOM*, volume 2, pages 599–607, 2002.

[66] J. Y. Yeh and C. Chen. Support of multimedia services with the IEEE 802.11 MAC protocol. In *IEEE International Conference on Communications*, volume 1, pages 600–604, 2002.

[67] M. Yokoo, E. H. Durfee, Toru Ishida, and Kazuhiro Kuwabara. The distributed constraint satisfaction problem: Formalization and algorithms. In *IEEE Transactions on Knowledge and DATA Engineering*, volume 10, September 1998.

[68] L. Zhao and C. Fan. M-PCF: Modified IEEE 802.11 PCF protocol implementing QoS. In *Electronics Letters*, volume 38, November 2002.

[69] G. Zhijun, W. Youzheng, and Z. Jian. QoS guaranteed wireless LAN-wireless access to ATM. In *Proceeding of International Conference on Communication Technology*, 1998.