ABSTRACT

Hui, Jie. QoS Provisioning in Wi-Fi Networks: Capacity Modeling and Resource Control. (Under the direction of Professor Michael Devetsikiotis).

The ubiquitous Wireless Fidelity (Wi-Fi) networks, and their increasing quality of service (QoS) requirements for emerging applications, motivate extensive studies of QoS provisioning in such networks. Two tasks, namely, **capacity modeling** and **resource control**, are crucial in solving the problem:

A performance model is first needed to predict the network QoS metrics from the network settings. We propose a new *unified analytical model* to study the saturation throughput and delay performance of 802.11e Enhanced Distributed Coordination Function (EDCA), which is easier to apply than the most current ones. In order to find usable mathematical models for most cases where analytical models are not feasible, we first advocate the application of *metamodeling* techniques to Wi-Fi performance studies and formulate a general metamodeling framework for such purpose. The results in three case studies support the validity of our methodology: our saturation capacity metamodel for 802.11 Distributed Coordination Function (DCF) displays an interesting log-linear relationship between capacity and number of users; our voice over Wi-Fi admission capacity metamodel gives a much tighter bound than bounds existing in the literature; and, finally, our throughput metamodel for a simple ad-hoc network, for the first time, characterizes the cross-layer effects between MAC and network layer schedulers. Our work, therefore, points out a new direction for future performance studies of Wi-Fi networks.

Then, based on the performance models we derive, resource *control* schemes of input parameters can be designed to achieve certain level of QoS outputs in some cases. For example, we are able to design a Weighted Round Robin (WRR) scheduler at the MAC layer to control the share of the radio resources, by applying our analytical model to a special case of EDCA configuration. Furthermore, based on our fitted metamodel for the capacity of voice over Wi-Fi, a more practical admission control scheme is composed.

QoS Provisioning in Wi-Fi Networks: Capacity Modeling and Resource Control

by

Jie Hui

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Doctor of Philosophy

Department of Electrical and Computer Engineering

Raleigh

2005

Approved By:

Dr. Arne Nilsson	Dr. Steve Roberts
Dr. Michael Devetsikiotis Chair of Advisory Committee	Dr. Wenye Wang

To my parents for their everlasting unconditional love for me, staying beside me in this past tough year and supporting me meal by meal;
To my husband for traveling overseas to accompany and support me;
To the light of my life, my dear son Mark. Part of this Ph.D. work was done when he was inside me and kicking me. The last piece was made when he called his daddy as mom and cried to his grandparents for milk. Thank him, I appreciate life more.
To my sister, wish her a happy birthday today.

献给我的父母，感谢他们给我有如滔滔江水永无止尽的爱；

献给我的丈夫，感谢他飘洋过海来陪我追寻我的梦；

献给我的生命之光，我的儿子小彬斌，感谢他让我更敬仰生命和心存感激；

献给我的姐姐，祝她今天生日快乐。

Biography

Jie Hui was born in Shanxi, China. She received the B.S. degree in Electrical Engineering from the Xi'an Jiaotong University, Xi'an, China, in 1996, and the M.S. degree in Electrical Engineering from Institute of Space Electronic Equipment, Shanghai, China, in 1999. From 2000 to July 2001, she worked with 3G R&D center of ZTE Corp. in Shenzhen China as a system and software engineer. From August 2001 to August 2005, she has been pursuing her Ph.D. degree in Computer Engineering specializing in wireless networking in the department of Electrical and Computer Engineering of North Carolina State University. Her research interests include QoS provisioning for multimedia over Wi-Fi, medium access control, cross-layer control and optimization, modeling, performance evaluation and efficient simulation of wireless networks.

Acknowledgements

First and foremost, I would like to acknowledge the unending love, support and trust my parents had given to me throughput the past of my life. It is my mother who cultivated my interest in math and trained my logical thinking abilities by solving geometrical problems together when we were walking after dinners. It is my father who taught me the altitude of life: to always try your best and not care too much about results. Without their taking care of the new baby, this Ph.D. work is simply impossible and I can barely survive.

My earliest dream is to be a mathematician mainly because the wonderful math teachers I met during my early schools. Among them are my mother, Xia Shurong and Yuan Chunxin. Then I switched my dream job to be a novelist when I totally devoted my spare time to reading novels and had an excellent Chinese teacher Song Yuanle. It is only now that I realized how important those fundamental education and skills are to my whole career life. But finally I ended up becoming an electrical engineer and thank to extraordinarily smart advisor Zhou Liqi during my master degree program, I was introduced to the mystic world of wireless communication. His profound knowledge of radio frequency circuit design, astronomy, electrodynamics, mechanics and thermodynamic amazed me and still remain unsurpassed. During my four years Ph.D program, my advisor Dr. Michael Devetsikiotis continuously encouraged me even when I was unsure myself. Thank both him and Dr. Wenye Wang for accommodating many of my special needs. When I most needed an powerful simulation tool for my research, Dr. Steve Roberts taught me Arena and systemical simulation methodology which proved to be very useful and also influenced me by his enthusiastic teachings.

I would also say thanks to my friend, Hongyan Lei, for her patient listening to my whingeing during my bad times and for her genuine celebrating with me during my good times.

The last, I want to say thanks to my husband, who gave up his career in China and followed me to this strange land for pursuing of our new dream.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wi-Fi, referring to any type of IEEE 802.11 ('a', 'b', or 'g' or the future ones, like 'n' or 's') Wireless Local Area Network (WLAN)[1], is becoming the standard network in homes, hot spots in public areas, and in the enterprise. Its rapid deployment has boosted an increasing demand for quality of service (QoS) in Wi-Fi networks. This increase in demand motivates researchers to study how to provide QoS in Wi-Fi, which differs from wired networks and licensed cellular networks, due to their shared medium. The medium access control (MAC) schemes, Distributed Coordination Function (DCF) and Enhanced Distributed Channel Access (EDCA), which regulate the rules to share the medium, thus, remain the core of this problem. Two major steps are required to achieve QoS provisioning. The first one is to understand how the network settings including DCF and EDCA MAC parameters affects the QoS performances- to *model* or predict the QoS performance outputs in functions of network inputs; then based on the performance model comes the second one - to *control* the network settings in order to achieve the desired QoS metrics.

In this chapter, we first introduce the background knowledge of QoS in networks including wired and wireless networks, then motivate the study of QoS provisioning in IEEE 802.11 WLAN. Our contributions in this area through modeling and control are briefly summarized before the outline of this dissertation.

[1]Thereafter, Wi-Fi and WLAN are used alternatively in this dissertation.

1.1 Background

To provide practical and efficient end-to-end QoS schemes is one of the biggest challenges facing networking researchers and vendors. End-to-end QoS can not be achieved even if a *single* network element (wired or wireless) does not provide QoS.

1.1.1 Definition of QoS

Quality of service (QoS) in the networking area refers to the capability of a network to provide service differentiation and resource assurance to different streams according to their requirements.

The following are some of the QoS parameters reflecting the service differentiation:

- Throughput: The Desired Bite Rate (bps) or bandwidth.
- End-to-end (ETE) Delay: Delay encountered by a packet, the sum of transmission delay, processing delays (includes router look-up), queueing delay etc.
- Delay Jitter: Variations in ETE delay.
- Packet Loss Rate: the percentage of lost packets due to channel error or queue overflow.

The most successful network, the Internet, did not offer QoS in the beginning, same as the legacy Wi-Fi networks. The Best Effort packet delivery is the only service it could support, which means traffic is processed as quickly as possible, but there is no guarantee as to timeliness or actual delivery.

QoS support, however, is essential for business and real-time applications such as VoIP and on-line multimedia services. Before we can say we already have effectively infinite bandwidth in the whole world, we still need QoS schemes to allocate limited resources to multi-users based on the service level agreements they purchase.

1.1.2 QoS in wired networks

In the Internet, there are a number of layer 3 routers in the backbone and also some layer 2 switches in the edge LAN. To support end-to-end QoS, all of them need to be QoS aware.

Layer 3 QoS: Two new architectures for resource allocation in layer 3, IntServ and DiffServ, are proposed in the Internet over the last decade.

The *IntServ* architecture is based on per-flow resource reservation. That is to reserve resource for every flow in every hop router like in telephone networks. A signaling protocol, RSVP, sets up and maintains the reservation path in the control plane. Then the data flow can go through the reserved path, which is classified according to the reservation index, and is scheduled using the reserved bandwidth in the data plane.

To avoid the high overhead of per-flow reservation in IntServ, *DiffServ* treats packets on a per-class basis. DiffServ pushes complexity to the network edge who will read per-flow information and classify, mark and police the traffic accordingly; and simplifies and speeds the operation in the core routers who need only forward packets differently according to up to 8 classes. The class information is carried in the DSCP or IP precedence fields in the IP packet header.

Layer 2 QoS: Layer 2 bridges and switches, the main connecting components of LANs, support class-based differentiated forwarding treatments at the MAC level. The basic forwarding treatments of packets in switches are very similar to those in DiffServ routers. The only difference is that switches could only see MAC frames and use CoS to classify traffic.

1.1.3 QoS in Wireless Cellular Networks

In recent years, with the increasing popularity of wireless communications, many researchers have made efforts to adapt wireline-inspired fair queueing algorithms to wireless cellular systems facing the big challenges of location-dependent and bursty channel errors. The common strategy of these schemes is to use compensation to remedy the fairness guarantees. The authors in [1] proposed an unified wireless fair queueing architecture and mapped those wireless fair scheduling algorithms (IWFQ [2], CIF-Q [3] and WFS [4]) to the unified architecture, then compared their properties. These scheduling algorithms differ in terms of the model they emulate, the swapping and compensation model they use.

1.1.4 QoS in Wireless LAN

Similar to wired or wireless cellular networks, wireless stations or wireless Access Points can support flow or class based priority treatment for applications which compete for the same output wireless port. But the difference is that there is no competition among output port resources in wired network or wireless cellular network, since the resources are reserved for each port (wire bandwidth in wired network and wireless frequency/time slot/code in wireless cellular network). However in IEEE 802.11 WLAN, all wireless stations and AP share the same wireless medium. Therefore, the QoS in wireless medium access is further required in order to achieve the end-to-end QoS.

QoS in AP and Wireless Station: AP is a layer 2 switch with WLAN at one side and wired Ethernet LAN at the other side. The wireless output port in AP and wireless station can provide the same class-based differentiated forwarding treatments as the normal wired output port does. Or it can provide flow-based treatment since the number of flows/users in a WLAN will not be too large.

QoS in Medium Access: In LAN, access to the link bandwidth is arbitrated by the MAC protocol in a decentralized fashion. 802.4 (Token Bus), 802.5 (Token Ring) support up to eight access priorities, but legacy 802.11 DCF MAC does not support differentiated access. In order to provide end-to-end QoS, a QoS enhancement over DCF, namely EDCA, is being standardized by the IEEE 802.11e group to add the priority access features.

Protocol Description of DCF and EDCA

A legacy DCF wireless station performs CSMA/CA with the following BEB procedures [5] to access the wireless medium (Fig. 1.1):

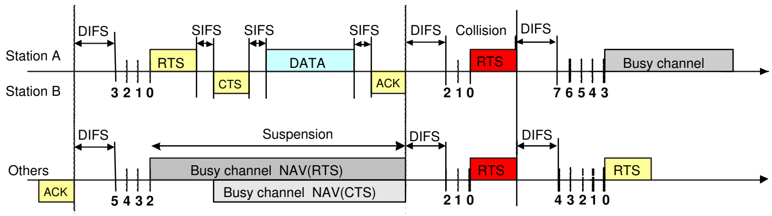


Figure 1.1: DCF Access Procedure.

Defer: A station with a pending packet has to wait for the channel to be idle for the duration of a Different Inter Frame Space (DIFS) before the transmission in order to give priority access to Polling or Control Messages. If the channel is sensed busy during this period, the station has to wait for another idle DIFS after the channel is idle again, then performs a random backoff. **Backoff:** Then, the station has to wait for an additional random backoff time, which is uniformly distributed between 0 and CW_{min} slots. If the channel is sensed to be busy during this period, the station suspends backoff until the channel is idle for DIFS again. **Handshaking:** If the packet size is bigger than a threshold, a two-way handshaking procedure is performed to further reduce the DATA collision probability, including a RTS and a CTS packet. **Data Transmission:** If the above procedures are successful, the DATA packet will be transmitted. **Confirmation:** Then, the station awaits an acknowledgement from the destination for confirmation. **Collision and Retransmission:** If more than one station begin their transmissions at the same time, a collision happens. The collided station will defer, backoff, and then retransmit with a new contention window size ($CW_{new=}CW_{old} * 2 + 1$) until CW_{max} is reached, then stays unchanged at CW_{max}. If the retransmission attempts reach a retry limit, the packet will be discarded.

To provide differentiated channel access, EDCA [6] supports up to four access categories (AC) in each QoS station, for packets belonging to eight user priorities (UPs) or frame types. AC values of 0, 1, 2 and 3 represent best effort, background, video and voice AC, respectively. The mapping between UP or frame type and AC can be found in the draft [6].

Comparing to the equal access of DCF contentions by using the same $DIFS$,

CW_{min} and CW_{max}, EDCA offers differentiated access through the EDCA parameter set $AIFS[AC]$, $CW_{min}[AC]$, $CW_{max}[AC]$ and $TXOPlimit[AC]$ for a corresponding AC (AC=$0, 1, 2, 3$). $AIFS[AC]$ is determined by $AIFS[AC] = SIFS + AIFSN[AC] * T_{slot}$, where $AIFSN[AC]$ is an integer indicating the number of slots after a SIFS duration a station should defer before either invoking a backoff or starting a transmission. The minimum value for $AIFSN$ is 2 for backoff entities of non-AP stations and 1 for back entities of QoS access points (QAP). Transmission opportunity (TXOP) is a new scheme to improve the efficiency of the protocol. A backoff entity can transmit multiple packets within one TXOP, of which the maximum length is $TXOPlimit[AC]$.

1.1.5 Packet Scheduling in QoS Schemes

Packet schedulers act in a key role in order to allocate the limited resources to competing flows/classes/stations with priorities in all of the QoS aspects above.

There are basically two kinds of scheduling mechanisms: Priority-based Scheduling (PQ) and Fair-share Scheduling (FS). The principle of PQ is to always serve the higher priority queue before the lower priority queue. The FS is trying to emulate the ideal Generalized Processor Sharing (GPS) in real packet-switched networks. Numerous fair queueing and scheduling algorithms have been proposed and analyzed. One category is to serve the queues in a round-robin fashion using frames or cycles, such as RR, WRR and DRR. The other category is to assign a service deadline (or time-stamp, or virtual finishing time) for each packet of different connection and serve them in the increased order of this deadline, such as WFQ, WFQ2, and SCFQ, etc. The main difference between various algorithms is the way to compute the virtual time.

Packet schedulers (PS) with the above scheduling algorithms are used to allocate resources in different places in the network. Routers use PS in layer 3 to allocate bandwidth of output port to different incoming flows. Switches use PS in layer 2 to do the same thing. Random Medium Access Control (MAC) Protocols use virtual schedulers controlled by the access algorithms to allocate shared medium bandwidth to the access entities.

1.2 Motivation: QoS Provisioning

As we conclude from the above section, QoS is akin to sharing limited resources, with priorities between competing parties. To share the same output port bandwidth, either wired or wireless, packet schedulers/controllers can be used to allocate resource either at network layer or MAC layer, either based on per-flow or per-class separation.

Among different output ports, there is no competition in wired or wireless cellular networks, since each output port is associated with an individual wire or wireless channel (frequency/time slot/code). However, in unlicensed 802.11 WLAN, no specific wireless resource is reserved for each port. All wireless stations compete for the same shared wireless medium. Therefore, how to share this scarce wireless bandwidth with priorities and how to restrict the number of competing parties are crucial issues in order to provide QoS.

The legacy 802.11 WLAN does not offer QoS access because its MAC function, DCF, treats all stations with the same access parameters. With the increasing demand of QoS in WLAN and its rapid deployment, a QoS enhanced MAC function over legacy DCF, so called EDCA, is being standardized by a IEEE 802.11e working group. EDCA differentiates the service among stations/applications by allocating different access parameters, such as contention window (CW) size or arbitrary inter frame space (AIFS). Therefore, how to provide the desirable QoS performance by adjusting or controlling those access parameters in a WLAN is a big problem facing network researchers, also the topic of our research and this dissertation.

Another urgent problem for VoIP applications over Wi-Fi networks is a good call admission control scheme (CAC) which determines when to accept or deny a call request in order to guarantee QoS level for existing users and the will-be admitted call. An accurate estimation of the VoIP capacity in a Wi-Fi network, i.e., the maximum number of VoIP calls can be supported, therefore is the major component of the answer, which also corresponds to a part of our research effort.

1.3 Contributions: Capacity Modeling and Resource Control

Both of our objectives, a scheduler of controlling EDCA access to provide predicable QoS performance and a CAC to support acceptable QoS for VoIP, require estimation of Wi-Fi capacities, under different settings. Our major contributions in this dissertation are done in capacity estimation (or capacity modeling) and can be categorized into: analytical modeling, simulation modeling and metamodeling. The control schemes to achieve the QoS provisioning goals are then rooted in the corresponding models naturally.

1.3.1 Methodology

A Wi-Fi network can be viewed as a generic *system*. The first step in studying a system is to define a vector of controllable variables (inputs) $X = \{x_1, x_2, ..., x_m\}$ and a vector of responses (outputs) $Y = \{y_1, y_2, ..., y_n\}$, where m and n are the sizes of inputs and output respectively. Here for QoS provisioning problems, inputs to a Wi-Fi network can be DCF and EDCA access parameters and number of admitted users, etc. And the outputs can be QoS throughput and delay performance of the Wi-Fi network.

The objective of the performance modeling, therefore, is to find the relationship between input X and output Y, $Y = F(X)$, or in other words, to construct the response surfaces [2] of $y_i = f_i(x_1, x_2, ..., x_m)$, where i=1,2,...,n.

The form of the true response function $F(\cdot) = \{f_1(\cdot), f_2(\cdot), ..., f_n(\cdot)\}$ of the system is unknown and perhaps very complicated, hence we must approximate it. The way to approach this approximation can either be based on measurements or modeling. *Measurements* of the real system although are perhaps accurate data, only offer lim-

[2] If there are two inputs in a system $X = \{x_1, x_2\}$, we may visualize the values of one response variable y_i as a surface lying above the (x_1, x_2) plane. It is this graphical perspective of the problem environment that has led to the term *response surface* [7]. Even if the 'surface' is not a true graphical two-dimensional surface when the number of inputs is not two ($m! = 2$), the term response surface is still used to represent generally the collection of response values.

ited sample points over the response surfaces due to the computation or sampling costs inherent in the experiments. Therefore, *abstract models* of the real system need to be built in order to exhaustively explore the response surfaces and obtain predictions and formulate strategies. Fig.1.2 shows different ways in which a system might be studied.

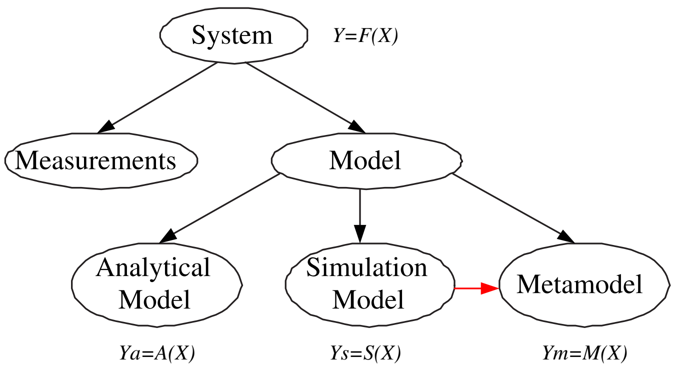


Figure 1.2: Ways to Study a System.

Types of Models

In general, we can build three types of models to abstract a system.

- *Analytical model:* When the behavior of the system is easy to analyze, an analytical model is a good way to characterize the rules regulating the system. The responses from the analytical model for inputs of X are Y_a with an approximation error ε_a,

$$Y_a = A(X) = Y + \varepsilon_a.$$

  An analytical model $A(\cdot)$ can be as simple as $s = vt$ in the example of a car system which asks about the distance of a car traverses during a period of time t with speed of v, and can sometimes be non-explicit and extraordinarily complex, requiring vast computing resources for finding solutions.

- *Simulation model*: For some complex systems, analytical models may not be possible to find or too complex to solve, therefore simulation models are used instead to capture the system behavior. The responses from the simulation model for inputs of X are Y_s with an approximation error ε_s,

$$Y_s = S(X) = Y + \varepsilon_s$$

An advantage of the simulation model over the analytical model is that it can simulate the system dynamics in great detail, thus providing a closer approximation to the system without the frequent need of assumptions required in the analytical model for facilitating the analysis. Quite often, simulation models also act as tools for validating the analytical models.

- *Metamodel:* An explicit closed-form response function is most wanted to explore the whole response surface and develop optimizing strategies. A simulation model $S(\cdot)$ is simply not such a function, since it is just a mechanism that turns input parameters into output performance measures. If the analytical model $A(\cdot)$ of a system is not available or can not offer such a closed-form function, people turn to another modeling method, metamodeling, to find the solution.

  A "metamodel" is defined as a mathematical closed-form model of the simulation model, or a 'model of the model' in [8]. The responses from the metamodel for inputs of X is Y_m with approximation error ε_m,

$$Y_m = M(X) = Y + \varepsilon_m$$

1.3.2 Contributions

In order to achieve QoS provisioning objectives in Wi-Fi networks, we make the following three contributions through the modeling methodology illustrated above and control schemes based on them:

Analytical Modeling of EDCA in Single-hop WLAN

The increasing QoS requirements for rapid deployed IEEE 802.11 WLAN networks motivate extensive performance evaluations of the upcoming 802.11e QoS-aware EDCA. Most of the analytical studies up-to-date have been based on one of the three major performance models in legacy DCF analysis, requiring a large degree of complexity in solving multi-dimensional Markov chains.

Here, we expose the common guiding principle behind these three seemingly different models. Subsequently, by abstracting, unifying and extending this common principle, we propose a new unified performance model and analysis method to study the saturation throughput and delay performance of EDCA, under the assumption of finite number of stations and ideal channel conditions in a single-hop WLAN. This unified model combines the strengths of all three models, and thus is easy to understand and apply; on the other hand, it helps increase the understanding of the existing performance analysis.

Despite its appealing simplicity, our unified model and analysis are validated very well by simulation results. And ultimately, by means of the proposed model, we are able to precisely evaluate the differentiation effects of EDCA parameters on WLAN performance in very broad settings, a feature which is essential for network design.

Centralized Control in Single-hop WLAN

An appropriate transfer function derived from modeling serves as the foundation for our control scheme. In a cooperative infrastructure WLAN, AP can control access priorities of stations through control messages. Therefore, we design a centralized priority controller in AP with the control function coming from the reverse function of the analytical model derived in our first contribution for a special configuration of EDCA.

Metamodeling Framework and Case Studies

Some of the easier performance problems of Wi-Fi networks can be solved by analytical modeling methods, but most of the complicated ones, involving too many factors from multiple layers, can only be answered through validated simulation models. However, an explicit mathematic model is always the most effective way to represent the system behavior and the most convenient basis for performance optimization.

Here, we first advocate the application of metamodeling techniques to performance studies of Wi-Fi networks, in order to find usable, if approximate, closed-form mathematical models. Subsequently, we formulate a general metamodeling framework for Wi-Fi networks.

Our results in three relevant case studies, after applying this framework, support the validity of our metamodeling methodology: our capacity metamodel for 802.11 Distributed Coordination Function (DCF) is validated by a well-known analytical model and displays an interesting log-linear relationship between capacity and number of users; our voice over Wi-Fi admission capacity metamodel gives a much tighter bound than bounds existing in the literature and composes a more practical admission control scheme; and, finally, our throughput metamodel for a simple ad-hoc network, for the first time, characterizes and quantifies the cross-layer effects between EDCA MAC layer and network layer through significant interactions in the metamodel. Our work, therefore, points out a new direction for future performance studies of Wi-Fi networks.

1.4 Outline

In the rest of the dissertation, these three accomplishments are presented in detail before summary and brief outline of future work.

Chapter 2 describes an analytical model of differentiated MAC access (802.11e EDCA) in a single-hop network. Based on a special case of this model, a centralized scheduler is introduced in Chapter 3 for single-hop cooperative WLANs. Chapter 4 advocates the application of metamodeling methodology and presents a metamodeling framework for Wi-Fi performance studies; three subcases based on this framework leverage and support the validity of our methodology. Finally, Chapter 5 summarizes this dissertation.

Chapter 2

Analytical Modeling of EDCA in Single-hop WLAN

2.1 Introduction

Throughput and delay analysis of contention-based random multiple access techniques, especially Carrier Sense Multiple Access (CSMA) and its variations, has long been a research focus in packet networks since 1970s. Accompanying the standardization and rapid deployment of IEEE 802.11 WLANs in 1990s, the performance analysis of its contention-based DCF MAC access function [5], a CSMA with collision avoidance (CSMA/CA) scheme with slotted binary exponential backoff (BEB), has been studied extensively by analytical or numerical means in recent years. Among those analytical studies, three major performance models have been proposed in parallel, in order to analyze the saturation throughput/capacity performance: Assuming a constant collision probability for each station, Bianchi [9, 10] proposed a Markov Chain to approximately model the behavior of CSMS/CA/BEB DCF, found the equilibrium packet transmission probability in a generic slot time by solving the Markov Chain, and finally obtained the saturation throughput by applying regenerative analysis to a generic slot time; Calì [11, 12] analyzed a p-persistent variant of DCF, with persistence factor p derived from the contention window (CW) in DCF, then found the capacity similarly using renewal theory; Tay [13] used instead an *average value* mathematical model, in order to calculate the packet collision probability, and solved the maximum throughput in terms of collision probability. A variation of Bianchi's model was proposed by Wu in [14] for the further consideration of retry limits.

Driven by the rapid growth of WLAN traffic volume and the different needs of applications, the IEEE 802.11 Task Group E has been working for several years to enhance the current best effort 802.11 MAC to support a QoS-aware WLAN. EDCA, one of the main and mandatory schemes in 802.11e [6], parameterizes DCF CSMA/CA scheme with prioritized Exponential Backoff (EB) to achieve differentiated QoS. In the recent years, the performance of EDCA has been explored by means of not only simulation [15, 16, 17, 18, 19, 20, 21, 22], and [23], but also analytical evaluations [24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34], and [23]. Most of the EDCA analytical studies are based on the modifications of DCF analysis mentioned above: [23] and [27] extends and parameterizes Calì's p-persistent DCF to accommodate different classes; with the exception [26], the others all modify or extend Bianchi's Markov Chain model [10] to accommodate the differentiation of Arbitration Inter Frame Space (AIFS) and/or CW. [24, 28, 31, 32] analyze the differentiation effects of only CW, while in the others, the differentiation effects of both AIFS and CW are considered. [30] and [29] varies transition rates on top of the original Markov Chains, while [25] and [34] enlarge the original bi-dimensional Markov Chain to tri-dimensional, and [33] enlarges it even to multi-dimensional. Other than multi-dimensional Markov chains, [26] provides a new rigorous analytical approach to model AIFS-based priority mechanisms but also with the weakness of high complexity.

To achieve more successful embedding of QoS-aware MAC in network schemes, such as call admission control and scheduling schemes, a performance model/analysis needs to be easier to understand and apply. The EDCA analysis mentioned above, which can evaluate differentiation effects of all EDCA parameters, all require high complexity.

Motivated by this need to *simplify* and to *unify*, we re-examine thoroughly the foundations of these EDCA analyses, including Bianchi's seminal Markov model, Calì's p-persistent CSMA model and Tay's average value model for DCF analysis.

We establish and expose a common guiding principle behind these seemingly different models: all of them assume saturation traffic conditions and homogeneous slots accessed with the same probability. The probability can be either the constant transmission probability (Calì) or constant collision probability (Bianchi and Tay); and they can be converted to each other. For example, the persistence factor p in Calì's model derived from CW using an iterative algorithm, actually, is the transmission probability calculated from Bianchi's Markov Chain model, and also can be converted from the collision probability in Tay's model which was solved by average value analysis. In other words, both Bianchi's and Tay's model imply a constant transmission probability, which agrees with the constant transmission probability resulting from a geometrically distributed backoff interval in Calì's model. This homogeneity, in turn, guarantees the application of regenerative analysis.

The differences among these methods consist of rather technical side issues, concerning diverse naming for transmission probability, different methods of finding the probabilities, and varied choices of renewal cycles. For instance, the renewal cycle in Calì's model was explicitly marked to be the time between two adjacent successful transmissions; implicitly, Bianchi picked a generic slot and Tay chose time between two transmissions to be their renewal cycles. But in the common assumption of slot homogeneity, these different renewal cycles are all valid and also can be transformed to each other.

Keeping these commonalities in mind, in this chapter we borrow strengths from these three models and compose a new unified performance model for EDCA analysis, without involving large complexity. We still use renewal theory to formulate the throughput performance, but assume constant packet transmission probabilities for different stations in different periods of time to account for differentiation effects of both AIFS and CWmin/CWmax in EDCA. The secondary assumption is that each packet transmission probability only depends on a unique collision probability. These transmission probabilities consist of a *two dimensional* persistence factor matrix, resulting in a p-persistent like CSMA/CA performance model. We denote this set of transmission probabilities by matrix $\mathbf{P}$, and thus name the unified model as the ***generalized*** $\mathbf{P}$***-persistent CSMA/CA*** *model* (called in the sequel $\mathbf{P}$-persistent, for short). In other words, we adopt a p-persistent like system in which persistence factors are time-dependent, while all the other pervious models used static persistence factors.

In order to solve for those transmission probabilities or the persistence factors in matrix $\mathbf{P}$, we provide extensions to both Bianchi's Markov Chain analysis and Tay's mean value analysis. This unified model, on one hand, reduces the complexity of Markov chains and is easy to apply; on the other hand, it increases the understanding of several efforts in the past on saturation throughput analysis of 802.11 MAC and allows better understanding the system behavior by exploiting the time-dependent persistence factors. Finally, the accuracy of the model and the analysis is well validated by simulation results.

The remainder of this chapter is organized as follows: First, we briefly review DCF and EDCA mechanisms in Section 1.1.4. Second, in Section 2.2, we propose our unified performance model, a $\mathbf{P}$-persistent CSMA/CA, for EDCA. How to derive $\mathbf{P}$, the key factor of the model, is illustrated in Section 2.2.3. Applying regenerative analysis to this model with knowledge of $\mathbf{P}$, we calculate the saturation throughput performance and delay performance of 802.11e EDCA separately in Section 2.3 and Section 2.4, in the assumption of finite number of stations and ideal channel conditions in a single-hop WLAN. We then establish the accuracy of this model by simulation results and study the differentiation effects of EDCA parameters in Section 2.6. Finally, in Section 2.7, we present our conclusions and describe our future directions.

2.2 Unified Performance Model

2.2.1 Assumptions and Configurations

Assumptions

According to the latest draft of the 802.11e standard, we make the following assumptions about the EDCA random access system which is going to be analyzed in the following section:

- Finite number of stations contend in a single-hop network;
- Each station has one access entity which belongs to one of the four ACs, and the number of stations deploying a AC is $N[AC]$;
- Ideal channel conditions: Propagation delay is zero, there is no channel error, no hidden node problem and no capture effects;
- Synchronized and slotted system: The time immediately following a busy medium and an idle AIFS[AC] is slotted and a station is allowed only to transmit at the beginning of each slot time (T_{slot}). Here we assume the slot boundaries in all stations are synchronized. This assumption together with ideal channel conditions implies a synchronous start of frame transmissions.
- Saturation traffic: all stations always have constantly back-logged queues;
- Constant successful transmission time T_s slots, including constant payload transmission time T slots [1], overhead and control message time: $T = PacketSize/DataRate/T_{slot}$, and $T_s = (PHY\&MACheader + RTS + CTS + ACK + 3 * SIFS)/T_{slot} + T$;
- Constant collision time $T_c = RTS/T_{slot}$ slots . In this chapter, we assume RTS/CTS access mode. However, the analysis can be easily applied to the basic access mechanism without RTS/CTS with the modification of T_c;
- TXOPlimit[AC]=0 for all ACs. We assume only one data frame is transmitted per EDCA TXOP.
- Infinite retry limit for a frame after collision as assumed in [10][2].

[1]Not affecting the validity of the model, we assume a constant packet payload size in this chapter; otherwise, T will be a function of the average payload size.

[2]The inclusion of a finite retry limit in the model is straightforward [14] and not discussed here in this chapter.

EDCA parameter sets

We also define the priority EDCA parameter set for a corresponding AC as $AIFS[AC]$, $CW_{min}[AC]$ and $CW_{max}[AC]$. For the ease of mathematic expressions in this chapter, we transform the standard EDCA parameters into the following: $d_i = AIFS[4-i]/T_{slot}$, $W_i = CW_{min}[4-i]+1$, $m_i = log_2^{\frac{CW_{max}[4-i]+1}{CW_{min}[4-i]+1}}$, and $n_i = N[4-i]$.

In other words, there are n_1 stations using voice access category (AC=3), ..., n_4 stations using best effort access category (AC=0). d_i can be interpreted as the length of $AIFS[4-i]$ measured in slots. From the draft [6], we know $AIFS[0] \geq AIFS[1] \geq AIFS[2] \geq AIFS[3]$. Therefore, $d_1 \leq d_2 \leq d_3 \leq d_4$. W_i is just an even number larger than the minimum contention window size by one; and m_i is the maximum backoff stage for $AC = 4 - i$.

Backoff Range

There are different conventions regarding the inclusion of the bounds in the range of the backoff slots. The uncertainties and changes in 802.11e drafts during the standardization process also reflect this.

In this chapter, we assume the range to be $[1, CW[AC]]$ inclusive. The reason why the backoff counter starts from 1 is because of the backoff suspension during a busy channel. For example, two backoff entities A and B contend for channel access. Entity A initiates a frame exchange at a particular slot, then B will defer from channel access upon detecting channel busy and suspend the decreasing of its backoff counter. After transmission, Entity A will randomly pick up a new backoff slot from zero to CW; and B will resume its backoff function upon detecting channel idle again. Before A can transmit again, B has to count down at least one more slot. This means the minimum backoff slot for A should not be zero. Therefore, a lower bound of 1 is a must for the backoff range. Thesis [23] also explains why backoff counter should start from one instead of zero.

2.2.2 Unified Performance Model

If several EDCA stations contend for a radio channel with the configurations as we assumed above, we will observe on the time axis an alternate sequence of idle periods (consisting of defer time and backoff slots) and transmission periods (successful or unsuccessful). An idle period or a delay period (D) and a following transmission period (TP) compose a cycle. An example of channel state with two stations contend for is shown in Fig. 2.1. To study the throughput and delay performance, like most of the classical CSMA analysis did, we need to find out the average length of the delay period, the transmission period and the useful message transmission time using regenerative analysis.

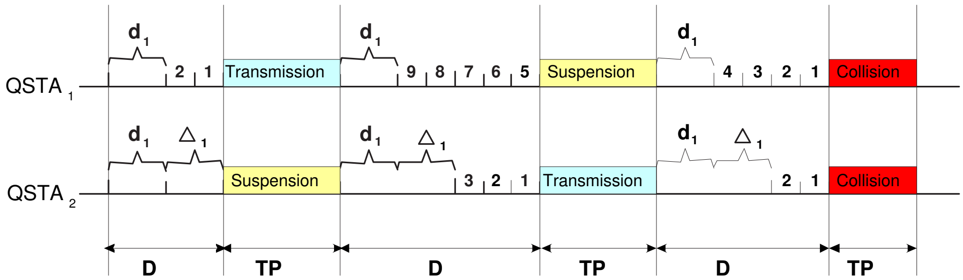


Figure 2.1: EDCA Channel States with Two stations ($d_1 = 2, d_2 = 4$).

However, the packet scheduling governed by EDCA BEB is not memoryless, since it depends on the transmission history (e.g., how many retransmissions the head-of-line packet has suffered). Therefore, those cycles are not strictly speaking *regenerative* cycles, and the average length of the delay periods and transmission periods, which depend on the backoff algorithm, can not be calculated easily.

Since we can not directly study the performance of EDCA system, we are going to construct a performance model to approximate the behavior of EDCA backoff algorithm, then study the performance of this model instead. If the model is appropriate enough based on good approximations, we should find good consistency between real system performances (from simulation) and model performances (from analysis).

Next, we construct such a performance model which abstracts, unifies and extends the common guiding principle behind the three previous DCF performance models by assuming constant transmission probabilities varying according to different stations and differing period of time. This key assumption, together with the memoryless packet scheduling regulated by this ***unified model*** as follows, makes the renewal analysis of throughput and delay performance possible.

- First, we divide the possible random delay period into 4 Backoff Sub-Periods ("BSP"s, for short, in the following) as shown in Fig.2.2. The jth BSP is defined as the period of time between $AIFS[4-j]$ and $AIFS[3-j]$ for $j = 1, 2, 3$; and the forth BSP is defined as the period of time greater than $AIFS[4]$. Thus, the length of BSP_j under the unit of slots is $\Delta_j = d_{j+1} - dj$ for $j = 1, 2, 3$.

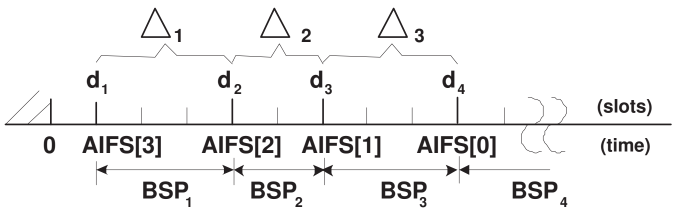


Figure 2.2: Random Delay Model.

- Second, we define the *slot transmission probability* p_{ij} as the transmission probability of a station belonging to AC$(4-i)$ in a slot boundary within the BSP_j, i.e., when the channel is idle, all n_i stations of AC$(4-i)$ will transmit in a slot within Δ_j with probability p_{ij}, or postpone the transmission by one slot with probability $1 - p_{ij}$, where $i = 1, 2, 3, 4$ and $i = 1, 2, 3, 4$.

- Third, there are two possibilities for transmission: If a station transmits and succeeds, the other stations have to wait for T_s slots until the transmission finishes, then repeat the same procedures. Otherwise, if more than one station tries to grab the channel in the same slot, a collision happens. All stations have to wait for T_c slots until the collision is detected, then repeat the above contention procedures.

We call this model the generalized **P**-persistent CSMA/CA, because all the slot transmission probabilities (p_{ij}) can actually be seen as persistence factors, assuming each station is using a classical p-persistent CSMA/CA. In this case, the persistence factors vary in different time periods, also according to different access categories. We insert all the persistence factors into a matrix, and name it the *slot transmission probability matrix* $\mathbf{P}$. As we can see, $\mathbf{P}$ is a 4×4 upper triangular matrix since i and j are both from 1 to 4 and stations of AC$(4-i)$ cannot transmit in BSP_j.

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ 0 & p_{22} & p_{23} & p_{24} \\ 0 & 0 & p_{33} & p_{34} \\ 0 & 0 & 0 & p_{44} \end{bmatrix}$$

In this model, the packet scheduling is *memoryless* because the transmission probabilities are independent and there are always $\sum_{i=1}^{4} n_i$ packets waiting in the beginning of a cycle (saturation traffic condition). Therefore, the time in which a transmission ends are *renewal points*. Fig. 2.3 shows the renewal cycles of the channel states. We denote by random variable D the duration of the random delay period[3], and by random variable TP the duration of a transmission period, which varies according to the failure or the success of the transmission.

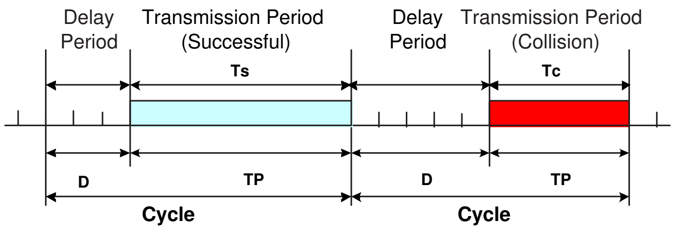


Figure 2.3: Channel Model.

Using the classical renewal property, we can calculate the saturation throughput S as the ratio of expected successful transmission time over the expected cycle length as in Eq. (2.1), where U represents the successful payload transmission time during a cycle. Furthermore, access delay can be directly derived from throughput using the

[3]The delay model is shown in Fig. 2.2.

relationship $AccessDelay = T/S$.

$$S = \frac{E(U)}{E(D) + E(TP)} \tag{2.1}$$

2.2.3 Calculation of P

In order to calculate the throughput and delay performance for the model, we first need to find the key factors, namely the slot transmission probabilities in matrix **P** from the EDCA backoff algorithm. We use the following three steps to calculate them.

Represent Slot Collision Probabilities c_{ij} as a function of p_{ij}

A transmission happens with probability p_{ij} for a station of AC$(4 - i)$ during a slot in BSP_j. It may succeed or collide. The slot collision probability c_{ij} is one minus the probability that all other stations do not transmit, including the other $(n_i - 1)$ stations of the same AC and all the other stations of different ACs.

$$c_{ij} = 1 - \frac{\prod_{k=1}^{4} (1 - p_{kj})^{n_k}}{1 - p_{ij}}, \qquad i, j = 1, 2, 3, 4. \tag{2.2}$$

Represent Slot Transmission Probabilities p_{ij} as a function of c_{ij}

Besides the key assumption, we further assume c_{ij}s are constant and independent of the backoff stage (secondary assumption) as assumed in other EB analyses [10, 35, 32, 13]. Thus, we can represent p_{ij} as a function of a unique c_{ij} using two methods here: Mean Value Analysis extended from [13] and Markov Chain Analysis extended from [10].

Mean Value Analysis

Given collision probabilities c_{ij}, the number of transmissions for a station with $AC = 4 - i$ to transmit a packet successfully in BSP_j is geometrically distributed with parameter $(1-c_{ij})$. When the backoff stage is r, the contention window size is updated to be $2^r * W_i - 1$ for $r = 0, 1, \ldots, m_i$. Since we only consider the contentions in BSP_j, which begins from the end of BSP_{j-1}, the average backoff slots in BSP_j is $\frac{2^r * W_i - \sum_{k=1}^{j-1} \Delta_k}{2}$, where $\sum_{k=1}^{j-1} \Delta_k$ represents the summation of all the past BSPs.

Subsequently, we can calculate the expectation of the number of backoff slots for a station of $AC = 4 - i$ in BSP_j conditioning on backoff stage r.

$$\begin{aligned}
\overline{w_{ij}} = & \quad (1-c_{ij})\frac{W_i - 1 - \sum_{k=1}^{j-1}\Delta_k}{2} \\
+ & \quad +c_{ij}(1-c_{ij})\frac{2*W_i - 1 - \sum_{k=1}^{j-1}\Delta_k}{2} + \ldots \\
+ & \quad (c_{ij})^{m_i}(1-c_{ij})\frac{2^{m_i}*W_i - 1 - \sum_{k=1}^{j-1}\Delta_k}{2} \\
+ & \quad (c_{i,j})^{m_i+1}\frac{2^{m_i}*W_i - 1 - \sum_{k=1}^{j-1}\Delta_k}{2} \\
= & \quad \frac{1-c_{ij}-c_{ij}*(2*c_{ij})^{m_i}}{1-2*c_{ij}} * \frac{W_i}{2} - \frac{1+\sum_{k=1}^{j-1}\Delta_k}{2}.
\end{aligned}$$

Therefore, p_{ij}, the slot transmission probability of a station with $AC = 4 - i$ in BSP_j, is the reciprocal of the average number of backoff slots plus one as in Eq. (2.3), where $i = 1, \ldots j$ and $j = 1, 2, 3, 4$. The other stations with AC values less than $4 - j$ transmit in this Δ_j period with probability of zero, implying $p_{ij} = 0$, when $i > j$.

$$p_{ij} = \frac{1}{\overline{w_{ij}}+1} = \frac{2}{W_i - \sum_{k=1}^{j-1}\Delta_k + c_{ij}*(W_i-1)*\frac{1-(2*c_{ij})^{m_i}}{1-2*c_{ij}}}. \tag{2.3}$$

Markov Chain Analysis

For each station $(AC = 4 - i)$ in BSP_j, we let $r_{ij}(t)$ represent the stochastic process of backoff stage $(0, \ldots, m_i)$, and let $b_{ij}(t)$ be the stochastic process of backoff timer $(1, 2, \ldots, W_i^{(m_i)} - 1)$ at time t. We can model $\{r_{ij}(t), b_{ij}(t)\}$ as a two-dimensional embedded Markov Chain shown in Fig. 2.4[4]. Similarly to the previous method, stations are not using the original contention window sizes, but the equivalent *smaller* contention

[4] In total, there are 10 Markov chains for different is and js, since $i = 1, \ldots j$ and $j = 1, 2, 3, 4$.

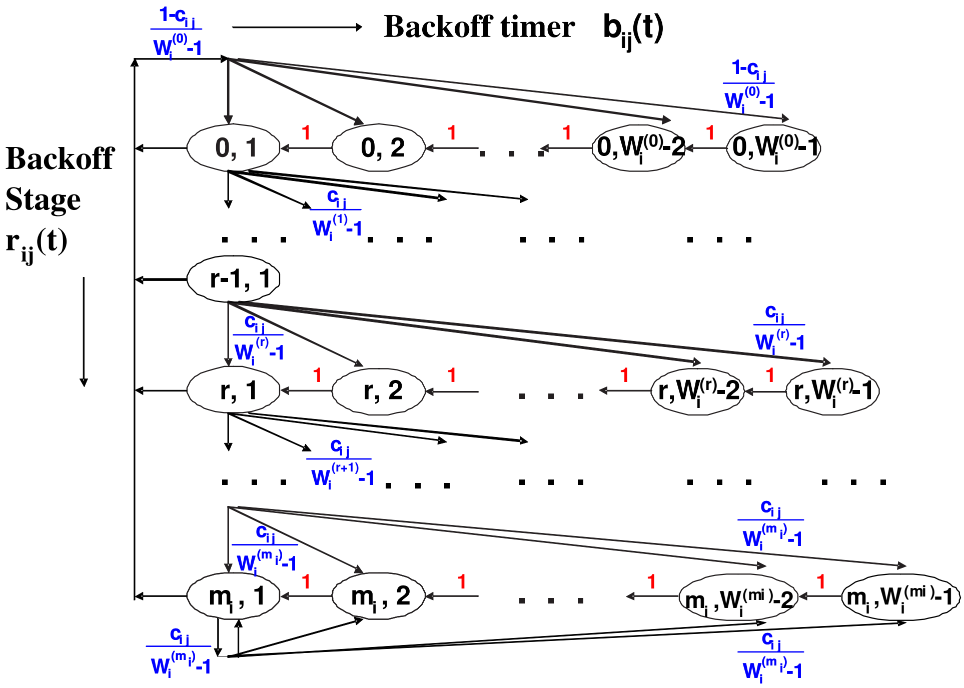


Figure 2.4: Markov Chain of $\{r_{ij}(t)$,$b_{ij}(t)\}$ for a station $(AC = 4 - i)$ in BSP_j.

window sizes $W_i^{(r)}$, subtracting all *past* BSPs from the CWs, $W_i^{(r)} = 2^r * W_i - \sum_{k=1}^{j-1} \Delta_k$, for backoff stage $r = 0, 1, \ldots, m_i$.

In this Markov chain $\{r_{ij}(t)$,$b_{ij}(t)\}$, the only non-null one-step transition probabilities are[5]:

$$\begin{cases} P_{ij}\{r,k|r,k+1\} = 1, & r \in [0, m_i], \\ & k \in [1, W_i^{(r)} - 2]; \\ P_{ij}\{0,k|r,1\} = \frac{1-c_{ij}}{W_i^{(0)}-1}, & r \in [0, m_i], \\ & k \in [1, W_i^{(r)} - 1]; \\ P_{ij}\{r,k|r-1,1\} = \frac{c_{ij}}{W_i^{(r)}-1}, & r \in [1, m_i], \\ & k \in [1, W_i^{(r)} - 1]; \\ P_{ij}\{m_i,k|m_i,1\} = \frac{c_{ij}}{W_i^{(m_i)}-1}, & k \in [1, W_i^{(m_i)} - 1] \end{cases}$$

[5]We adopt the short notation: $P_{ij}\{k_1, l_1|k_0, l_0\} = P_{ij}\{r_{ij}(t+1) = k_1, b_{ij}(t+1) = l_1|r_{ij}(t) = k_0, b_{ij}(t) = l_0\}$.

Whenever the backoff timer of a station is one, the station is going to transmit in the next slot. As stated in [32], for the consideration of backoff suspension stage and the consistency of Markovian states of different stations, we combine the backoff timer state 0 with state 1, and combine states before and during suspension to be a new state. Therefore, the slot transmission probability p_{ij} is just the summation of the stationary state probabilities of states with backoff timer value of one. Let $b_{ij}(r,k) = lim_{t->\infty} P\{r_{ij}(t) = r, b_{ij}(t) = k\}$ be the stationary distribution of the chain. Therefore, for $i = 1, ..., j, \;\; j = 1, 2, 3, 4,$

$$p_{ij} = \sum_{r=1}^{m_i} b_{ij}(r,1) = \frac{2}{W_i - \sum_{k=1}^{j-1} \Delta_k + c_{ij} * (W_i - 1) * \frac{1-(2*c_{ij})^{m_i}}{1-2*c_{ij}}} \tag{2.4}$$

As we can see, the result is the same as what was derived using Mean Value Analysis above. The derivation is similar to that in [10], thus not discussed here in detail.

Solve c_{ij} and p_{ij}

From the last two steps, we know c_{ij} is a function of n_i and $p_{1j}, ..., p_{jj}$, p_{ij} is a function of c_{ij}, W_i and m_i. Since n_i, W_i and m_i are all known, by solving the $2j$ dimensional nonlinear equations composed of equations of $c_{1j}, ..., c_{jj}$ from Eq. (2.2) and equations of $p_{1j}, ..., p_{jj}$ from Eq. (2.3) or (2.4), we can solve the values of p_{1j}, p_{2j}, ..., and p_{jj}. Repeating this procedure for all the BSPs ($j = 1, 2, 3, 4$), we can obtain the transmission probability matrix $\mathbf{P}$.

2.3 Saturation Throughput Analysis

By knowing $\mathbf{P}$, and then applying probabilistic analysis to the $\mathbf{P}$-persistent CSMA/CA performance model, we can calculate the average random delay $E(D)$, the average successful transmission time $E(U)$ and the average transmission period $E(TP)$, then obtain the generalized saturation throughput performance for 802.11 EDCA.

2.3.1 Average Random Delay $E(D)$

Before formulating $E(D)$, let us define two more transmission matrices besides $\mathbf{P}$.

System Slot Transmission Probability Matrix P_{tr}***:*** We already know p_{ij} represents the transmission probability of a ***station (***$AC = 4-i$***)*** at a ***slot*** boundary in BSP_j. For the whole system, it is possible that no one transmits in a slot. Then, we define the probability of the ***system*** to transmit at a ***slot*** boundary in BSP_j as *system slot transmission probability* $p_{tr\,j}$. It is related to p_{ij} by $p_{tr\,j} = 1 - \prod_{i=1}^{j} (1-p_{ij})^{n_i}$. Integrating all $p_{tr\,j}$ for $j = 1, 2, 3, 4$ into a matrix, we denote by P_{tr} the *system slot transmission probability matrix*,

$$P_{tr} = \begin{bmatrix} p_{tr\,1} & p_{tr\,2} & p_{tr\,3} & p_{tr\,4} \end{bmatrix}.$$

System BSP ***Transmission Probability Matrix*** P_Δ***:*** A BSP consists of many slots. In each slot, the system can be in transmission state with probability $p_{tr\,j}$. We define another matrix P_Δ as the *system* BSP *transmission probability matrix*, $P_\Delta = \begin{bmatrix} p_{\Delta_1} & p_{\Delta_2} & p_{\Delta_3} & p_{\Delta_4} \end{bmatrix}$. The jth elements of the matrix - p_{Δ_j} represent the probability that transmission begins in BSP_j. Thus, it is the product of the probability that no transmission happens in the past BSPs and the probability that at least one transmission happens in BSP_j:

$$\begin{aligned} p_{\Delta_j} &= [1-(1-p_{tr\,j})^{\Delta_j}] \prod_{k=1}^{j-1} (1-p_{tr\,k})^{\Delta_k}, j = 1, 2, 3 \\ p_{\Delta_4} &= \prod_{k=1}^{3-1} (1-p_{tr\,k})^{\Delta_k}. \end{aligned}$$

Then, we can calculate $E(D)$ using the following theorem.

Theorem: The average random delay can be computed by conditioning on the BSP in which transmission begins as

$$E(D) = d_1 + L \times P_\Delta^T = d_1 + \sum_{j=1}^{4} l_j \times p_{\Delta_j},$$

where L is the *conditional average backoff delay matrix* $L = \begin{bmatrix} l_1 & l_2 & l_3 & l_4 \end{bmatrix}$, and l_j represents the average backoff delay if transmission begins in BSP_j, with $l_j = \frac{1}{p_{trj}}$, $j = 1, 2, 3, 4$.

Proof. The random delay D is a random variable. It can take any value from the set $[d_1 + 1, ..., d_2, ..., d_3, ..., d_4, ...\infty)$[6]. For given system slot transmission probabilities, the tail-distribution function of D can be expressed as:

$$Pr(k) = Pr[D \geq k] =$$
$$\begin{cases} (1 - p_{tr1})^{k-d_1-1} & d_1 + 1 \leq k \leq d_2 \\ (1 - p_{tr2})^{k-d_2-1}(1 - p_{tr1})^{\Delta_1} & d_2 + 1 \leq k \leq d_3 \\ (1 - p_{tr3})^{k-d_3-1} \prod_{j=1}^{2} (1 - p_{trj})^{\Delta_j} & d_3 + 1 \leq k \leq d_4 \\ (1 - p_{tr4})^{k-d_4-1} \prod_{j=1}^{3} (1 - p_{trj})^{\Delta_j} & d_4 + 1 \leq k < \infty \end{cases}$$

Then, the expected delay can be represented as a function of $Pr(k)$ as:

$$E(D) = \sum_{k=d_1+1}^{\infty} k[Pr(k) - Pr(k+1)] = d_1 + 1 + \sum_{k=d_1+2}^{\infty} Pr(k) = d_1 + \sum_{k=d_1+1}^{\infty} Pr(k)$$

By inserting the values of $Pr(k)$, we obtain

$$\begin{aligned} E(D) = d_1 &+ \sum_{k=d_1+1}^{d_2} (1 - p_{tr1})^{k-d_1-1} + \sum_{k=d_2+1}^{d_3} (1 - p_{tr2})^{k-d_2-1}(1 - p_{tr1})^{\Delta_1} \\ &+ \sum_{k=d_3+1}^{d_4} (1 - p_{tr3})^{k-d_3-1} \prod_{j=1}^{2} (1 - p_{trj})^{\Delta_j} \\ &+ \sum_{k=d_4+1}^{\infty} (1 - p_{tr4})^{k-d_4-1} \prod_{j=1}^{3} (1 - p_{trj})^{\Delta_j} \\ = d_1 &+ \frac{1}{p_{tr1}}[1 - (1 - p_{tr1})^{\Delta_1}] + \frac{1}{p_{tr2}}[1 - (1 - p_{tr2})^{\Delta_2}](1 - p_{tr1})^{\Delta_1} \\ &+ \frac{1}{p_{tr3}}[1 - (1 - p_{tr3})^{\Delta_3}] \prod_{j=1}^{2} (1 - p_{trj})^{\Delta_j} + \frac{1}{p_{tr4}} \prod_{j=1}^{3} (1 - p_{trj})^{\Delta_j} \\ = d_1 &+ \sum_{j=1}^{4} l_j \times p_{\Delta_j} = d_1 + L \times P_{\Delta}^{T}. \end{aligned}$$

[6]Since the minimum value of backoff slots is 1 as explained in Section 2.2.1, the minimum random delay is $d_1 + 1$.

where l_j, p_{Δ_j} for $j = 1, 2, 3, 4$ are defined as above. □

2.3.2 Average Successful Transmission Period $E(U)$

A transmission can be successful or not. In order to calculate the average successful transmission period $E(U)$, we first need to find out the probability for a transmission to succeed.

We define P_s as the successful transmission probability for the whole system. That is the probability for a certain transmission in a cycle to be successful. It is the summation of the successful transmission probabilities of each station $P_s = \sum_{i=1}^{4} n_i * p_{si}$, where, we denote by p_{si} the probability that a transmission is successful for one of the n_i stations with AC equal to $4 - i$.

We introduce another quantity p_{sucij}, which represents the probability that the transmission is successful for a station $(AC = 4 - i)$ in BSP_j, given that there is at least one transmission in BSP_j. It is related to p_{ij} and p_{trj} by $p_{sucij} = \frac{p_{ij}}{1-p_{ij}} \times \frac{1-p_{trj}}{p_{trj}}$.

Similarly as in the previous section, we can calculate p_{si} conditioning on the BSP in which the transmission begins: $p_{si} = \sum_{j=1}^{4} p_{sucij} * p_{\Delta_j}$.

Thus, $E(U)$, the expected time during a cycle that the channel is used without a collision is the product of payload transmission time and the probability of a successful transmission:

$$E(U) = P_s \times T.$$

2.3.3 Mean Transmission Period $E(TP)$

By conditioning on the success of the transmission, we can calculate the mean transmission period as

$$E(TP) = P_s T_s + (1 - P_s) T_c.$$

2.3.4 Saturation Throughput

System Saturation Throughput S: Finally, we can obtain the total saturation throughput for the system by plugging $E(U)$, $E(D)$ and $E(TP)$ into Eq. (2.1) as

$$S = \frac{P_s \times T}{d_1 + \sum_{j=1}^{4} l_j * p_{\Delta_j} + P_s T_s + (1 - P_s) T_c}. \tag{2.5}$$

Station Saturation Throughput S_i: The saturation throughput for a station of $AC = 4 - i$ is the total throughput multiplied by p_{si}/P_s for $j = 1, 2, \ldots, N$.

$$S_i = \frac{p_{si} \times T}{d_1 + \sum_{j=1}^{4} l_j * p_{\Delta_j} + P_s T_s + (1 - P_s) T_c}. \tag{2.6}$$

Saturation Throughput Ratio: The n_i stations belonging to the same AC $(4-i)$ receive the same amount of saturation throughput. The saturation throughput ratio among stations of different ACs can be expressed as the ratio of successful transmission probabilities:

$$S_1 : S_2 : S_3 : S_4 = p_{s1} : p_{s2} : p_{s3} : p_{s4}.$$

2.4 Access Delay Analysis

We define the access delay as the time duration from the packet becoming Head Of Line (HOL) on the sender's side, until it receives acknowledgment from the receiver. The contributors to access delay include: 1) *Medium Access Delay*: The time between the packet becoming HOL and it beginning transmission. It further includes the backoff time, deferring periods and retransmission time due to collisions; 2) *Successful Transmission Delay:* The transmission time of useful data and overhead.

The calculation of access delay is straightforward following the throughput analysis. For one of the n_i stations with $AC = 4 - i$ $(i = 1, 2, 3, 4)$,

$$AccDelay_i = \frac{T_{cycle}}{p_{si}} = \frac{T}{S_i}. \tag{2.7}$$

The mean access delay is given by the packet transmission time T divided by the throughput. Similarly, for the whole system, the mean access delay given by

$$AccDelay = \frac{T_{cycle}}{P_s} = \frac{T}{S}. \tag{2.8}$$

2.5 Special Cases of EDCA

From the unified model, we can easily analyze the throughput of legacy 802.11 DCF and other special cases. The results totally agree with those in [10] and [32], in which the analysis is carried out in different ways.

2.5.1 Legacy 802.11 DCF

In legacy 802.11 DCF, all stations use the same MAC parameters, including $AIFS = d * T_{slot}$, CW_{min}, and CW_{max}, meaning there is only one AC, and one BSP. If we assume there are total n stations and let $W = CW_{min} + 1$ and $m = log_2^{\frac{CW_{max}+1}{CW_{min}+1}}$, in a slot within BSP, all n stations transmit with a same probability p, and succeed with a same successful probability p_{suc}:

$$\mathbf{P} = [p], \quad P_{suc} = [p_{suc}].$$

By solving the nonlinear equations $p = \frac{2}{W + c*(W-1)*\frac{1-(2*c)^m}{1-2*c}}$ and $c = 1-(1-p)^{n-1}$, p can be found; and thus $p_{suc} = \frac{p(1-p)^{n-1}}{1-(1-p)^n}$.

From $\mathbf{P}$, we can derive $P_{tr} = [p_{tr}]$, where $p_{tr} = 1-(1-p)^n$; $P_\Delta = [p_\Delta]$, where $p_\Delta = 1$; and the average random delay $E(D) = d + \frac{1}{p_{tr}}$.

Subsequently, each station receives the same saturation throughput. The normalized system saturation throughput is

$$\begin{aligned} S &= \frac{P_s \times T}{d + \frac{1}{p_{tr}} + P_s T_s + (1-P_s) T_c} \\ &= \frac{p_{tr} \times P_s \times T}{(1-p_{tr})*1 + p_{tr} P_s (T_s + d + 1) + p_{tr}(1-P_s)(T_c + d + 1)}. \end{aligned}$$

where $P_s = n * p_{suc} = \frac{np(1-p)^{n-1}}{1-(1-p)^n}$. This expression is the same as that in [32], which was derived using the concept of generic slot introduced in [10].

2.5.2 EDCA (Same AIFS, Different CW)

In this special case, only the contention window sizes are used to differentiate the services for stations of different ACs, while the $AIFS = d*T_{slot}$ are kept the same for all stations, thus resulting in different $W_i = CW_{min}[4-i]+1$ and $m_i = log_2^{\frac{CW_{max}[4-i]+1}{CW_{min}[4-i]+1}}$. We also assume there are n_i stations for AC of $4-i$, where $i = 1, 2, 3, 4$. This differentiation scheme is very commonly used practically.

Stations only compete in BSP_4 with length of Δ_4, since $\Delta_1 = \Delta_2 = \Delta_3 = 0$. Similarly, only the 4^{th} columns of matrixes $\mathbf{P}$ and P_{suc} are non-zero with different values.

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & p_1 \\ 0 & 0 & 0 & p_2 \\ 0 & 0 & 0 & p_3 \\ 0 & 0 & 0 & p_4 \end{bmatrix}, \; P_{suc} = \begin{bmatrix} 0 & 0 & 0 & p_{suc1} \\ 0 & 0 & 0 & p_{suc2} \\ 0 & 0 & 0 & p_{suc3} \\ 0 & 0 & 0 & p_{suc4} \end{bmatrix},$$

From $\mathbf{P}$, we can derive $P_{tr} = [0\;0\;0\;p_{tr}]$, where $p_{tr} = 1 - \prod_{i=1}^{4}(1-p_i)^{n_i}$; $P_\Delta = [0\;0\;0\;p_\Delta]$, where $p_\Delta = 1$. p_1, p_2, p_3, p_4 can be solved from the nonlinear Equations $p_i = \frac{2}{W_i + c_i*(W_i-1)*\frac{1-(2*c_i)^{m_i}}{1-2*c_i}}$ and $c_i = 1 - \frac{\prod_{k=1}^{4}(1-p_k)}{(1-p_i)}$. The successful probability of a station of $(AC = 4-i)$ can be computed as $p_{si} = \frac{p_i}{1-p_i} * \frac{1-p_{tr}}{p_{tr}}$, and the total $P_s = \sum_{i=1}^{4} n_i * p_{si}$.

Similarly, the average random delay and the total throughput take the same value as those in the general case. The saturation throughput ratio among stations of different ACs is expressed as $S_1 : S_2 : S-3 : S_4 = \frac{p_1}{1-p_1} : \frac{p_2}{1-p_2} : \frac{p_3}{1-p_3} : \frac{p_4}{1-p_4}$, also the same as derived in paper [32].

2.6 Simulation Validation and Discussion

2.6.1 Simulation Model

We programmed a discrete-event simulation of a single-hop static WLAN. For each station, a traffic generator feeds traffic packets into a MAC queue. The packet stays in the MAC queue until it reaches HOL and wins the contention of the medium access. After the packet departs from MAC queue, it is transmitted by a transmitter. The receiver in the destination station accepts the packet and sends it to a sink after collecting statistics.

Traffic Model

The traffic generator generates packets according to the distribution of packet inter-arrival time and packet size. The distribution of packet inter-arrival time can be any distribution in our simulator. Because we study the network under saturation traffic in this chapter, we use constant packet inter-arrival time and set the traffic rate larger than the capacity of the network in order to make it saturated. We use in our simulation a constant payload size of 1500 bytes.

Radio Channel Model

As assumed in the analysis, we implement an ideal radio channel in the simulation: that means that the propagation delay is zero, and there is no channel error and no exposed or hidden node problem.

MAC Implementation

In order to minimize the number of events and speed up the simulation, we did not simulate the behavior of backoff entity in each station individually. Instead, we simulated the behavior of the 802.11e EDCA medium access as a virtual scheduler [7]. For example, the scheduler observes how many packets are in HOL position, then

[7] An ideal channel condition and the synchronized system make this simulation method feasible.

compares their $AIFS[AC] + backoff$ values. It tells the stations with the smallest value to transmit the packet, and tells the other station to hold the packet and retransmit, then advances the simulation time by $AIFS[AC]+backoff$ of the winner station plus the packet transmission time. If only one station has the smallest delay value, the packet transmission is a success and packet transmission time is T_s; if more than one station has the same smallest delay value, collision happens and the packet transmission time is T_c.

The parameters of 802.11 MAC and PHY deployed in the simulation as well the comparative analysis are shown in Table 2.1.

Table 2.1: 802.11 MAC/PHY Simulation Parameters

Parameter	**Value**	**Parameter**	**Value**
RTS	0.352 ms	SIFS	0.01 ms
CTS	0.304 ms	PHY/MAC header	0.328 ms
ACK	0.304 ms	Tslot	0.02 ms

2.6.2 Simulation Results

Simulation Validation: Experiment 1, 2 and 3

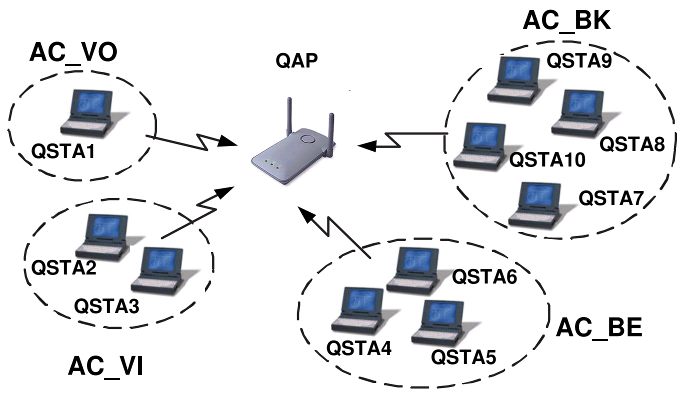


Figure 2.5: Simulation Scenario for Experiment 1.

The WLAN scenario we simulated in experiment $1, 2, 3$ is shown in Fig. 2.5. In this WLAN, ten stations send traffic to QAP. Each station deploys one backoff entity of one AC to contend for the channel. Among these ten wireless stations (WS), there is one backoff entity per AC_VO (WS1), two backoff entities per AC_VI (WS2 and WS3), three per AC_BE (WS4, WS5 and WS6) and four per AC_BK (WS7, WS8, WS9 and WS10).

The EDCA parameter sets of the three experiments are (see Table 2.2):

- Experiment 1: default setting from draft [6];
- Experiment 2: vary AIFS from the default setting;
- Experiment 3: vary CW from the default setting.

Table 2.2: 802.11e EDCA Parameter Sets for Experiment $1, 2, 3$

Experiment	**AC**	**CWmin**	**CWmax**	**AIFSN**
1	AC_BK	31	1023	7
(default)	AC_BE	31	1023	3
	AC_VI	15	31	2
	AC_VO	7	15	2
2	AC_BK	31	1023	7
	AC_BE	31	1023	5
	AC_VI	31	1023	3
	AC_VO	31	1023	2
3	AC_BK	31	1023	2
	AC_BE	31	1023	2
	AC_VI	15	31	2
	AC_VO	7	15	2

In each experiment, we simulate ten scenarios for this WLAN: progressively from scenario 1 to scenario 10, we add WS1 to WS10 to the system one at a time. Then, we collect the aggregate saturation throughput and access delay for each AC and the whole system, and compare the results from simulation with the results from our analysis in Fig. 2.6, 2.7 and 2.8. Lines represent analytical results, while the markers near the lines represent the corresponding simulation ones. We use subscript 'A' to represent analysis and 'S' to represent simulation in the legend. Each simulation result is averaged from twenty simulation replications, and each simulation replication lasts for 1000000 transmission cycles. The 95% confidence interval (CI) is shown as

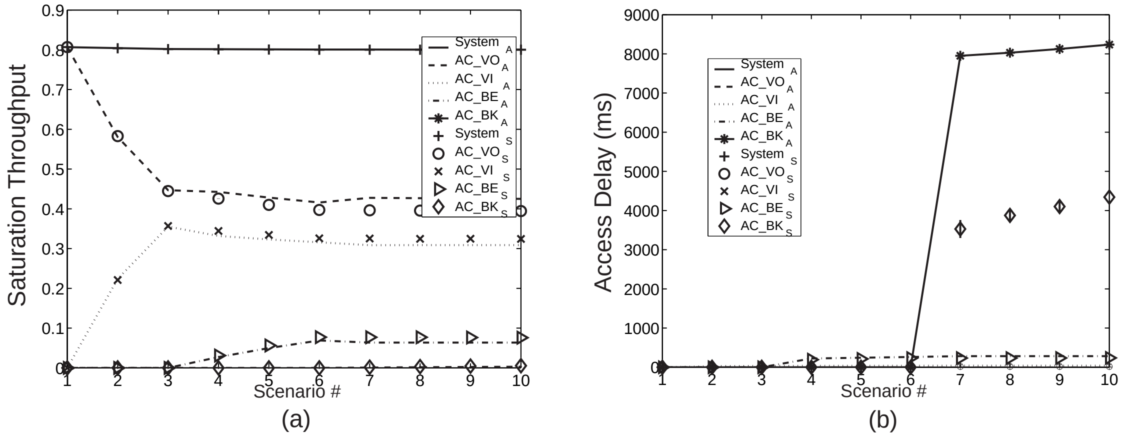


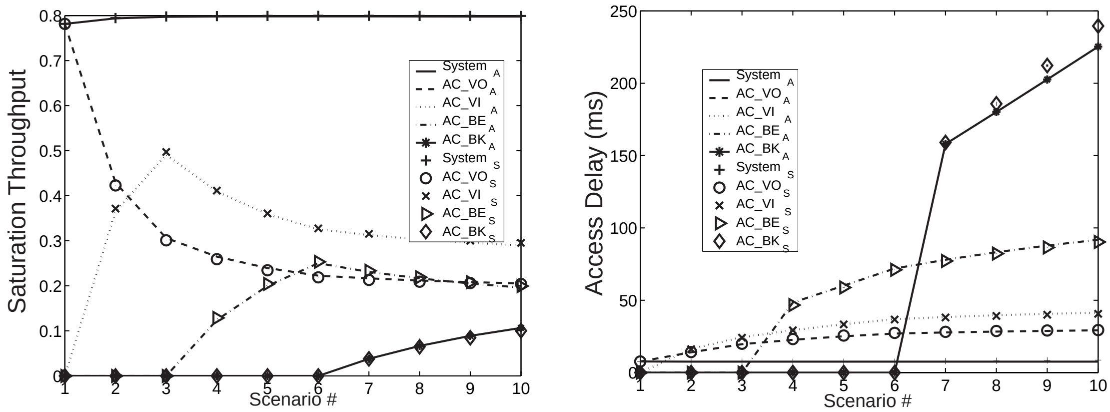
Figure 2.6: Experiment 1: Performance Using Default EDCA Parameter Set

a bar around each simulation result. For most of the results, the CIs are too small to be visible except the last four for access delay of AC_BK in each figure.

The agreement between analysis and simulation is remarkable. The only exception is the access delay for AC_BK in Fig. 2.6. The reason is that the throughput for a station of AC_BK is very small (of the order of 10^{-3}) and the access delay for a AC_BK station is very large (of the order of 10^3). In our analysis, the access delay is inversely proportional to the throughput. Therefore, a tiny difference between simulation and analysis result in throughput will result in a huge gap between access delay simulation and analysis results.

From experiment 1 (Fig. 2.6), we can also gain another insight: Although only one AC_VO station competes with two AC_VI stations, three AC_BE stations and four AC_BK stations, the maximum (saturation) throughput per AC_VO pumped into the network is still the largest, the maximum (saturation) throughput per AC_BK pumped into the network is the smallest and nearly zero. The reason is that the default EDCA parameter set differentiates among the four ACs very effectively through the combined effects of AIFS, CWmin and CWmax. AC_BK stations are almost starved in this experiment due to the long AIFS and big CWmin and CWmax, comparing to the very small AIFS and CW of AC_VO.

By varying the EDCA parameter set from the standard default values, we can



(a) (b)

Figure 2.7: Experiment 2: Performance by Varying Only AIFS

study the individual differentiation effects of AIFS and CW separately. In experiment 2, we keep CWmin and CWmax the same for all ACs, and just vary the AIFSN by two units between ACs; in experiment 3, we keep AIFSN all the same, but vary CWmin and CWmax. Fig. 2.7 and Fig. 2.8 show the results from these two experiments. The agreement between simulation and analysis is still very good. Beside that, by comparing these three figures (Fig. 2.6, 2.7 and 2.8), we can tell that the combined effects surely are greater than each individual one. But among them, the relationship is not simply additive. The difference between AC_VO and AC_VI is mainly due to the non-overlapping CW ranges. However the starvation of AC_BK is not due to each individual but the combined effects of both AIFS and CW differentiation.

Differentiation Effects: Experiment 4, 5 and 6

Another important factor not considered in the above experiments is the number of stations per AC. Realizing the correlation among the differentiation effects of all EDCA parameters, we believe that a formal sensitivity analysis over the whole response surfaces will be ideal to carry out a thorough study of parameter effects. However, such a study is beyond the scope of this chapter.

In the following, we perform three more experiments of specific settings and attempt to gain more understanding about the differentiation effects of AIFS, CWmin

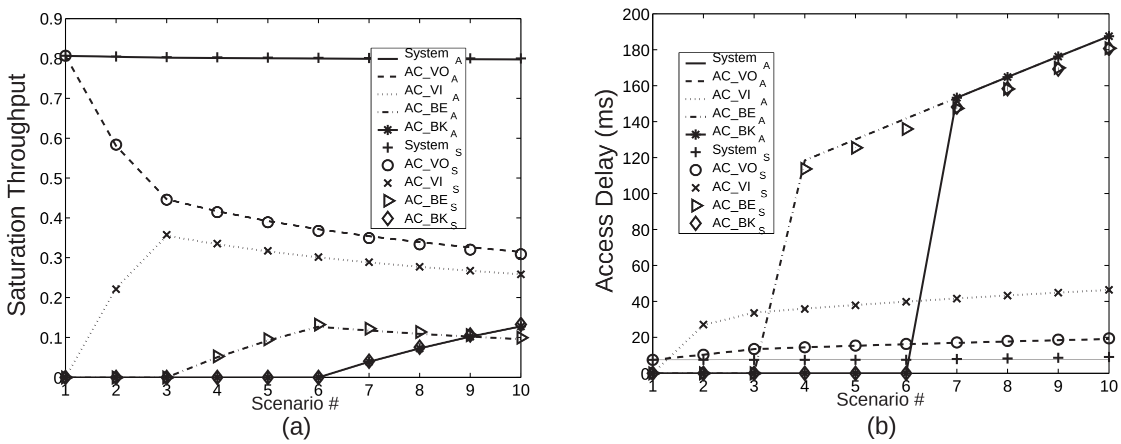


Figure 2.8: Experiment 3: Performance by Varying Only CW

and make some further observations. To filter out the effect of the number of stations and CWmax, we fix the number of stations for each AC to be equal to 3 and $CWmax = (CWmin + 1)^5 - 1$. The EDCA parameter sets are (see Table 2.3):

- Experiment 4: Only differentiate AIFS but keep CWmin and CWmax equal to 31 and 1023 respectively. The AIFSN differences increase from 0 to 4.

- Experiment 5: Only differentiate CWmin and CWmax but keep the AIFS equal to 2. The CWmin difference increases from 0 to 4.

- Experiment 6: Differentiate both AIFS and CWmin, CWmax by combining experiments 1 and 2.

From the result comparison in Fig.2.9, we can clearly see that a larger difference in AIFS or CW result in a larger difference in throughput and delay performance among different ACs. Furthermore, for this specific base setting $\{AIFSN, CWmin, CWmax\} = \{2, 31, 1023\}$, we can infer the following:

- AIFSN has a larger differentiation effect on the performance than CW for the same variation from 1 to 4;

- The combined differentiation effects of AIFSN and CW are bigger than both individual ones.

Table 2.3: 802.11e EDCA Parameter Sets for Experiment $4, 5, 6$

Exp	**AC**	**CWmin**	**AIFSN**
4	AC_VO	31	2, 2, 2, 2, 2
	AC_VI	31	2, 3, 4, 5, 6
	AC_BE	31	2, 4, 6, 8,10
	AC_BK	31	2, 5, 8,11,14
5	AC_VO	31,31,31,31,31	2
	AC_VI	31,32,33,34,35	2
	AC_BE	31,33,35,37,39	2
	AC_BK	31,34,37,40,43	2
6	AC_BK	31,31,31,31,31	2, 2, 2, 2, 2,
	AC_BE	31,32,33,34,35	2, 3, 4, 5, 6,
	AC_VI	31,33,35,37,39	2, 4, 6, 8,10
	AC_VO	31,34,37,40,43	2, 5, 8,11,14

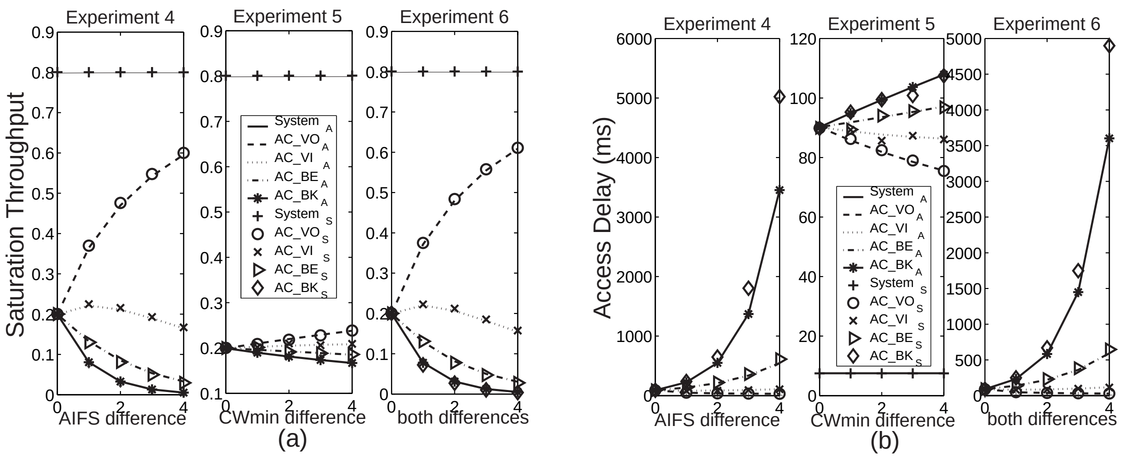


Figure 2.9: Experiment $4, 5, 6$: Comparison of Differentiation Effects

- By keeping all other parameters unchanged, a larger AIFS or CW results in a lower throughput and a longer access delay.

Special Case with very small CWmin: Experiment 7

For some special settings with very small CW values (like CWmin=1,2,3), the CWmin variations can show dramatic effects in performance differentiations. Also, a bigger CW may result in a higher throughput and a shorter access delay. The reason is that a very small CW value causes a high collision probability, therefore, an increase from the this small CW can help reduce the collision probability, thus resulting in a

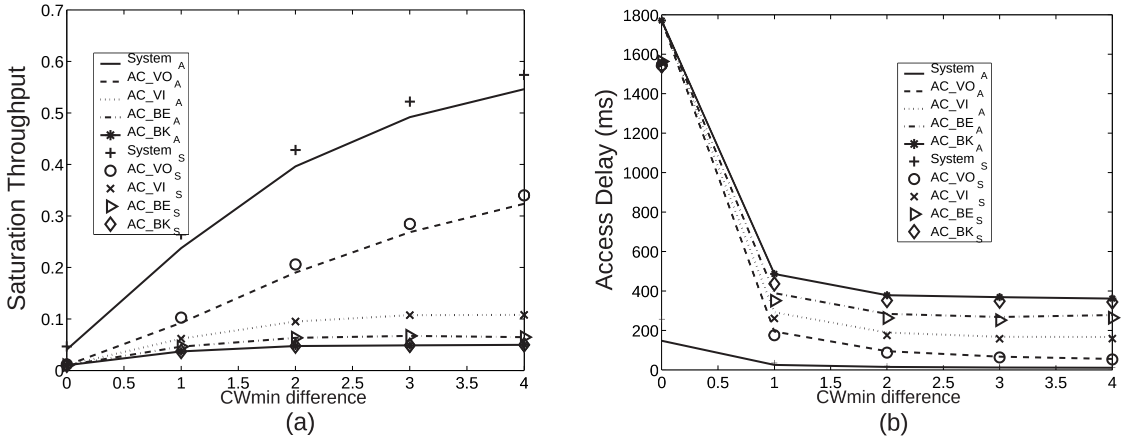


Figure 2.10: Experiment 7: Special Case of Very Small CWmin

better performance. Experiment 7 (see Table 2.4) explores such a special case. The results in Fig. 2.10 support the above observations.

Table 2.4: 802.11e EDCA Parameter Sets for Experiment 7

Experiment	**AC**	**CWmin**	**CWmax**	**AIFSN**
7	AC_BK	3,3,3,3,3	3,3,3,3,3	2
	AC_BE	3,4,5,6,7	3,4,5,6,7	2
	AC_VI	3,5,7,9,11	3,5,7,9,11	2
	AC_VO	3,6,9,12,14	3,6,9,12,14	2

From the above experiments and discussions, we surmise that the differentiation effectiveness around different parameter settings can vary from or even contradict each other. The number of stations, AIFS, CWmin and CWmax for each AC all affect the throughput and delay performance, and they are also correlated with each other. Therefore, as mentioned earlier, a formal sensitivity analysis has to be conducted to complete a more thorough study of the parameter effectiveness. This will be left as our future work.

Sensitivity with respect to number of stations: Experiment 8

To study how our model performs with the changing number of stations, we conduct experiment 8: the EDCA parameter set is as in experiment 1; the number of

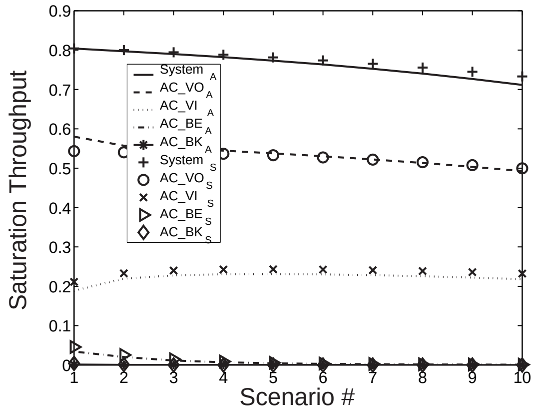


Figure 2.11: Experiment 8: Sensitivity with Respect to Number of stations

stations per AC increases from 1 to 10 ($N(0) = N(1) = N(2) = N(3) = 1, 2, ..., 10$).

From the saturation throughput results in Fig. 2.11, we observe that the predictive capability of our model remains high even with increasing number of stations. Furthermore, another effect is that the saturation throughput drops due to more frequent collisions, as the number of stations increases (greater than 2 per AC).

Comparing to the previous performance evaluations about both AIFS-based and CW-based differentiation approaches, and although we do not offer a rigorous sensitivity analysis, our work does provide the evaluation of four ACs with the standard 802.11e EDCA parameter setting. Most previous work only evaluates a WLAN with two or three classes. The good agreement between simulation and our analysis proves the accuracy of our model, and provides a useful tool for network design and analysis.

2.7 Conclusions

EDCA is introduced in 802.11e for QoS improvements over legacy 802.11 DCF. The understanding of how the EDCA parameters affect the performance of WLAN is a crucial prerequisite for the design of any QoS scheme using EDCA.

Our main contributions in this chapter are three fold: First, we abstract and unify a common guiding principle behind three major performance models, thus increasing the understanding and applicability of these efforts. Second, we propose a unified performance model and analysis method for 802.11e EDCA by taking elements of all three models, while maintaining their common principles. In our model, the memory effects of backoff counter and backoff stage are still accounted for by using a bi-dimensional state Markov Chain as in [10] or mean value analysis as in [13]; in a novel manner, in order to account for the effect of different AIFS values, we did not introduce further dimension(s) to the state space as in [25, 28, 34, 33], but we used multiple bi-dimensional chains or multiple average value analysis in separate backoff subperiods under the main assumption of time-dependent p-persistence behavior. This new model is easy to apply by reducing the complexity of Markov Chains and offering an alternative mean value analysis method to compute persistence factors. For another aspect, this model also allows better understanding of the system behavior by exploiting the concept of Backoff Sub-Periods, and by using the persistence factors matrix $\mathbf{P}$. Third, simulation results validate our model and analysis, showing that our model will be a helpful tool for 802.11e network designers.

All of the analyses and simulations in this chapter are performed with the assumption of ideal channel conditions, saturation traffic and a single-hop network environment. The study of throughput and delay performance of EDCA with a more realistic wireless channel model, different traffic models, and even in multi-hop networks, are included in our future plans. Another area of research can be to find closed form solutions for a simplified performance model and then perform a formal sensitivity analysis of each parameter, finally providing a dynamic turning of parameters according to the desired QoS level.

Chapter 3

Centralized Control in Single-hop WLAN

In a cooperative infrastructure mode WLAN, AP can control the sharing of medium resource of stations by emulating a packet scheduler at MAC layer through proper setting of EDCA parameters. In order to design such a scheduler, a inverse transfer function is needed in order to calculate EDCA parameters from the desired throughput allocation.

For the general case, this inverse function is difficult to find, But for a special case of EDCA configuration (same AIFS, different CW), we show in this chapter that an analytical reverse function can be found, and that ultimately, we can design a WRR scheduler at MAC layer to control the share of the radio resource. Our results show that the MAC WRR which was designed using our analytical model can achieve larger utilization, bandwidth ratio as expected and smaller average delay than the network layer model.

3.1 Related Work

For several years the IEEE 802.11 Task Group E has been working to enhance the current best effort 802.11 MAC [5] to support QoS for different applications.

EDCA, one of the main and mandatory schemes in 802.11e [6], parameterizes DCF to achieve differentiated QoS. In the last two years, the performance of EDCA has been explored by use of numerical evaluation [15, 16, 18]. Similar conclusions have been drawn, specifically that smaller Contention Window (CW) or smaller Arbitration Inter Frame Space (AIFS) will support a higher priority level, like a higher throughput and a shorter delay.

By applying these backoff based differentiation methods, a few distributed MAC layer schedulers have been proposed to emulate ideal centralized schedulers. Kanodia *et al.* designed a distributed priority scheduler [36] to emulate Earliest Deadline First (EDF) and Virtual Clock (VC); they also proposed Distributed Wireless Ordering Protocol (DWOP) to emulate FIFO by mapping the priority tag to backoff window size. Vaidya *et al.* proposed a Distributed Fair Scheduling (DFS)[37] to emulate Self-Clocked Fair Queueing (SCFQ) by setting backoff interval inverse proportional to the flow weights. They use simulation results to argue that the bandwidth allocation is proportional to the weights, something we will address in this chapter. Luo *et al.* proposed new packet scheduling models [38] to approximate WFQ with Modified WRR for a multihop wireless network, by assigning each backoff interval equal to a flow degree. Their model ensures fairness while seeking to maximize spatial reuse. Banchs *et al.* proposed Assured Rate MAC Extension (ARME) in [39] to provide soft throughput guarantees in line with the Assured Rate PDB for DiffServ by changing the CW dynamically. Barry *et al.* designed a whole system of QoS schemes, including differentiated backoff-based MAC access and radio channel monitoring algorithms which will help make correct admission decisions. Qiao *et al.* [24] proposed a priority-based fair medium access control (PMAC) protocol to achieve fairness and maximization of utilization simultaneously by modifying the DCF backoff scheme, which is also a part of 802.11e EDCA QoS schemes.

3.2 Throughput analysis of a special case of 802.11e EDCA

In this special case, only the contention window sizes are used to differentiate the services for stations, while the AIFS are kept the same for all stations. This differentiation scheme is very commonly used in practice.

All stations compete in the same BSP since they use the same AIFS, therefore, we denote the transmission probability for station i as p_i and collision probability as c_i. Different from Chapter 2, here in order to consider a general scenario, we assume there are n stations (n can be larger than 4).

3.2.1 Multi-stations in Uplink

The scenario assumed in this section is that n stations are competing for a single WLAN channel with the same DIFS but different contention window size $CWmin_i$.

By applying the methodology of performance model we developed in Chapter 2, we can derive p_i and c_i as:

$$p_i = \sum_{i=0}^{m} b_{i,1} = \frac{2}{W_i + p_i(W_i - 1)\left\{\frac{1-(2*c_i)^{m_i}}{1-2*c_i}\right\}} \tag{3.1}$$

$$c_i = 1 - \frac{\prod_{i=1}^{n}(1-p_i)}{(1-p_i)} \tag{3.2}$$

We then use numerical techniques to solve these $2n$ dimensional nonlinear equations.

The probability that at least one station transmits is

$$P_{tr} = 1 - \prod_{i=1}^{n}(1-\tau_i) \tag{3.3}$$

The probability of successful transmission of each station P_{s_i} and the probability of total successful transmission $P_s$can be calculated as:

$$\begin{aligned} P_{s_i} &= \frac{p_i}{1-p_i}\prod_{i=1}^{n}(1-p_i) \\ P_s &= \sum_{i=1}^{n} p_{s_i} = \sum_{i=1}^{n}\left\{\frac{p_i}{1-p_i}\prod_{i=1}^{n}(1-p_i)\right\} \end{aligned} \tag{3.4}$$

Normalized system saturation throughput S can be computed as in equation (3.5). We can then derive the allocated bandwidth ratio (saturation throughput ratio) as in equation (3.6). Here T_s is calculated as same as in [10], but we consider the RTS timeout in $T_c = RTSTimeout + EIFS$.

$$S = \sum_{i=1}^{n} S_i = \frac{P_s E[P]}{(1-P_{tr})T_{slot} + P_s(T_s+T_{slot}) + (P_{tr}-P_s)(T_c+T_{slot})} \tag{3.5}$$

$$\begin{aligned} S_1 : S_2 : ... : S_n &= P_{S_1} : P_{S_2} : ... : P_{S_n} \\ &= \frac{\tau_1}{1-\tau_1} : \frac{\tau_2}{1-\tau_2} : ... : \frac{\tau_n}{1-\tau_n} \end{aligned} \tag{3.6}$$

3.2.2 Multi-flows in Downlink

The scenario assumed in this section is similar to that in the previous section except that the competing stations are replaced by flows in a station. The slot transmission probability p_i remains as same as in the multi-station case, described by equation (3.1). However, taking into account the priority access during virtual collision, the collision probability of flow i would be calculated differently as in equation (3.7), and the successful transmission probability as in equation (3.8) and (3.9).

$$c_i = 1 - \prod_{k=1}^{i-1}(1-\tau_k) \tag{3.7}$$

$$P_{s_i} = p_i \prod_{k=1}^{i-1}(1-\tau_k) \tag{3.8}$$

$$P_s = \sum_{i=1}^{n} p_{s_i} = P_{tr} \tag{3.9}$$

P_s equals P_{tr} means that there will be no unsuccessful transmission since no real collision occurs. Then, the normalized system throughput S and the allocated bandwidth ratio can be expressed as followings:

$$S = \frac{P_s E[P]}{(1 - P_{tr})T_{slot} + P_s(T_s + T_{slot})} \tag{3.10}$$

$$\begin{aligned} S_1 : S_2 : ... : S_n &= P_{S_1} : P_{S_2} : ... : P_{S_n} \\ &= \tau_1 : (1 - \tau_1)\tau_2 : ... : (1 - \tau_1)...(1 - \tau_{n-1})\tau_n \end{aligned} \tag{3.11}$$

3.3 Design of a MAC Layer WRR Scheduler

As stated in [40], the reason that packet schedulers are implemented above the MAC layer in LANs is simply because the IEEE LAN standards - 802.X MAC and LLC do not provide the necessary mechanisms for a controlled bandwidth sharing. However, from the section above, we know that 802.11E MAC does provide thus a mechanism in WLAN. The total throughput and the throughput ratio can be changed by varying the EDCA access parameters.

In this section, we will show how to design a MAC layer WRR packet scheduler using an analytical method we derived above.

WRR: Weighted Round Robin is the simplest approximation of Generalized Processor Sharing (GPS) [41] for packet based networks. Every flow has an integer weight w_i corresponding to the expected service share. Based on those weights, a server with the total rate r pre-computes a service schedule (frame), which serves session i at a rate of $\frac{W_i}{\sum_i W_i} r$.

MAC Layer WRR: To emulate a WRR in MAC layer, we can assign the EDCA parameters in the following way (Fig. 3.1):

$$\begin{aligned} AIFS_{j+1} &= AIFS_j = DIFS, \\ CWmin_{j+1} &> CWmin_j (j = 0, 1, 2, \ldots, 7) \end{aligned} \tag{3.12}$$

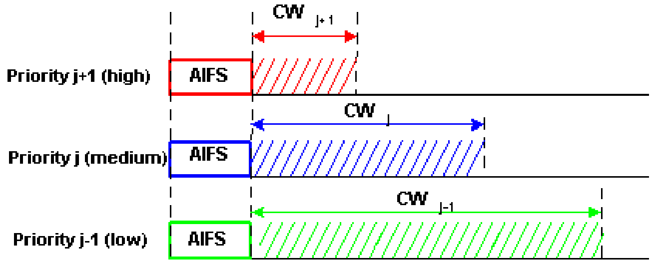


Figure 3.1: EDCA configurations for a MAC WRR.

3.3.1 Design of a MAC WRR

Our new analysis model in Section 3.2 can very closely estimate the saturation throughput performance for given specific station/flow contention window sizes. And vice versa, contention window sizes can also be derived for given expected total throughput and throughput ratio in multi-station and multi-flow cases in the following procedures.

Given Conditions: Normalized Throughput S; Bandwidth Ratio $ratio = [r_1, r_2, ..., r_n]$; Persistence factors $PF_1, PF_2, ..., PF_n$; MAC/PHY as in Table 2.1.

Uplink (Multi-stations)

From equation (3.5), we can represent P_s as a function of S and P_{tr} as:

$$P_s = \frac{P_{tr}T_c + T_{slot}}{(K/S - 1)T_s + T_c} \tag{3.13}$$

where K is the maximum throughput $K = \frac{\frac{AveragePacketSize}{DataRate}}{T_s}$.

Then, the slot transmission probability of station i can be derived from equation (3.4) as:

$$p_i = \frac{\frac{r_i}{\sum_{i=1}^{n} r_i} P_s}{\frac{r_i}{\sum_{i=1}^{n} r_i} P_s - P_{tr} + 1} \tag{3.14}$$

p_i, P_{tr} and P_s can be derived by solving the $n+2$ dimensional nonlinear equations consisting of equation (3.13), (3.14) above and (3.3) for P_{tr}.

The collision probability of station i can be calculated as

$$c_i = 1 - \frac{1 - P_{tr}}{1 - p_i} \tag{3.15}$$

Finally, the contention window sizes CW_i can be derived from equation (3.1) inversely:

$$CW_i = round\left\{ \frac{\frac{2}{p_i} - 1}{1 + c_i \frac{1-(2*c_i)^{m_i}}{1-2*c_i}} \right\} \tag{3.16}$$

Downlink (Multi-flows)

Recalling equation (3.10) about the normalized throughput and the maximum throughput K above, we can calculate the transmission probability P_{tr} as:

$$P_{tr} = \frac{T_{slot} \times S}{T_s \times (K - S).} \tag{3.17}$$

The successful transmission probability P_{s_i} of flow i can be calculated from the bandwidth ratio and P_{tr} using the relation $\sum_{i=1}^{n} P_{s_i} = P_{tr}$ as:

$$P_{s_i} = \frac{r_i}{\sum_{i=1}^{n} r_i} P_{tr} \tag{3.18}$$

Thus, the slot transmission probability of flow i can be derived from equation (3.8) as:

$$p_i \quad = \quad \frac{P_{s_i}}{\prod_{k=1}^{i-1} (1 - \tau_k)} \tag{3.19}$$

Then we can derive the collision probability of flow i easily by using equation (3.7). The final calculation of the contention window sizes CW_i would be same as equation (3.16).

3.3.2 Validation via Simulation

In this section, a simple experiment is conducted to evaluate our analysis method. For the given expected throughput and bandwidth ratio, first, we calculate the contention window sizes using our theoretical analysis. Second, the contention window CW is entered as input into a simulation to compute the throughput S and bandwidth ratio numerically, and then compared with our expected values.

Figure 3.2 compares the expected saturation throughput of the whole system and each station/flow with the simulation results in the multi-station and multi-flow cases (*Note: The confidence intervals are too small to be viewed in the graph*). Table 3.1 shows the values. We conclude from the comparison that our method provides a feasible way to design the MAC layer WRR.

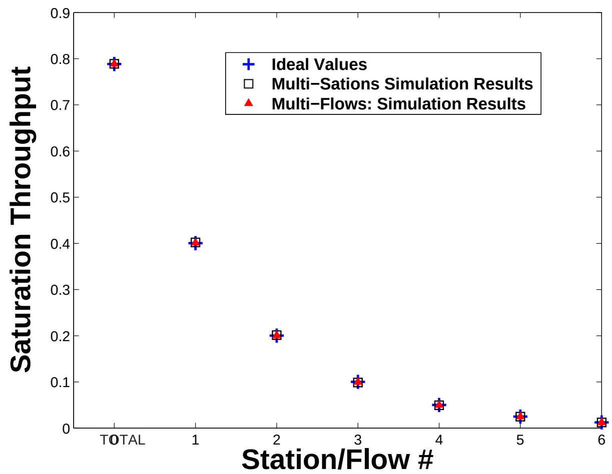


Figure 3.2: Design a MAC WRR: ideal versus designed saturation throughput.

Table 3.1: Experiment Results: n=6

Analysis Input	$S = 0.7885$ $r = [1\,0.5\,0.25\,0.125\,0.0675\,0.03375]$
Analysis Output	$CW = [35\,67\,130\,257\,474\,944]$ (MS) $CW = [47\,86\,166\,325\,595\,1182]$ (MF)
Simulation Input	$CW = [35\,67\,130\,257\,474\,944]$ (MS) $CW = [47\,86\,166\,325\,595\,1182]$ (MF)
Simulation Output	$S = 0.78893 \pm 0.00006$ (MS) $r = [1\,0.5011\,0.2458\,0.1231\,0.0617\,0.0303]$ $S = 0.78875 \pm 0.00003$ (MF) $r = [1\,0.5006\,0.2482\,0.1237\,0.0621\,0.0310]$

3.3.3 Advantages of MAC Layer WRR

Before the development of the 802.11e protocol, WRR schedulers would be implemented in the network layer of Access Point (AP) of a WLAN. For uplink scheduling, the AP uses polling based PCF access protocol to poll stations. For downlink scheduling, it polls its queues sequentially. In this case, why do we need to design a MAC layer scheduler instead? This question should be answered in two scenarios.

Uplink

A distributed EDCA MAC scheduler is the only way to prioritize access of different stations, assuming no polling protocol is used.

Downlink

A MAC layer scheduler provides higher throughput and smaller average access delay. The following section gives the simulation comparison between these two.

Simulation Configurations: The ideal WRR scheduler is to achieve the bandwidth ratio of six flows as 1 : 0.5 : 0.25 : 0.125 : 0.0625 : 0.03125, as previously. All of the MAC/PHY simulation parameters are the same as in Table 2.1.

For the *Network Layer WRR*, we assign different weights to flows, namely [32 16 8 4 2 1]. This network layer scheduler is on top of a legacy DCF MAC which is using $CWmin = 31$. The server rate of WRR is assumed equal to the capacity of the DCF.

For the *MAC Layer WRR*, we use the method in the previous section to design the scheduler as $CW_{min} = [47\,86\,166\,325\,595\,1182]$.

Throughput Performance Comparison: As shown in Table 3.2 , the network layer can control the link sharing exactly as expected; but, it can not achieve as much throughput as a MAC layer scheduler can. The reason is that the competition between multiple access instances helps reduce the time wasted in backoff in the latter case than the former one, which has only a single access instance in the link layer (using legacy DCF). Furthermore, the bandwidth ratio obtained from a MAC WRR is still acceptably close to the desired value.

Table 3.2: Throughput Performance of Network and MAC WRR.

	Network Layer WRR + DCF	**MAC Layer WRR + EDCA**
Throughput	0.78145±0.00001	0.78875±0.00003
BW1/BW1	1.0000	1.0000
BW2/BW1	0.4998	0.5002
BW3/BW1	0.2499	0.2479
BW4/BW1	0.1250	0.1238
BW5/BW1	0.0625	0.0619
BW6/BW1	0.0312	0.0307

Delay Performance Comparison: Another important QoS factor concerns the delay and delay jitter. From the simulation comparison in Table 3.3, we can see that flow 1 with the highest priority is treated preferably in both cases.

Table 3.3: Access delay Performance of Network and MAC WRR.

	Flow ID	**Ave (ms)**	**Std**	**95% CI**	**MIN (ms)**	**MAX (ms)**
L3 WRR	**1**	22.8	41.4	0.114	7.41	254
	2	37.9	87.3	0.34	15	377
	3	68.1	140	0.768	15	438
	4	129	196	1.53	15	469
	5	250	234	2.58	15	484
	6	491	0.268	0.00417	483	492
MAC WRR	**1**	15	7.98	0.022	7.37	74.6
	2	29.9	19.2	0.0748	7.37	606
	3	60.7	45.4	0.251	7.37	1180
	4	121	97.1	0.76	7.37	2280
	5	224	188	2	7.37	5750
	6	454	390	5.91	7.37	7910

A *MAC layer scheduler* can support smaller access delay with smaller deviation than a *network layer* scheduler does. But the maximum delay of the other low priority flows in a MAC scheduler is much higher than in the network layer scheduler. The main reason is that there will be more retransmissions for low priority flows because of the virtual collision resolution policy. This may restrict the use of MAC WRR in real time applications because the worst case delay may be too large to be acceptable.

3.4 Conclusions

Our main contribution in this work is to show how to design a MAC layer WRR scheduler by doing the analysis of 802.11e EDCA saturation throughput performance of a special EDCA configuration inversely. The MAC layer scheduler designed in this dissertation has certain advantages over a network layer scheduler, both for uplink and downlink communication. The MAC scheduler in the uplink helps us exempt the polling overhead. In the downlink case, simulation results showed that the MAC layer scheduler obtains better delay performance and larger and variable total throughput. Furthermore, it can also achieve the targeted throughput ratio between different flows.

Chapter 4

Metamodeling Wi-Fi Networks - a framework and three case studies

4.1 Introduction

Appropriate non-trivial performance models are required to study network performance of Wi-Fi networks. Some of the simpler problems can be solved by analytical modeling. The feasibility of analysis in DCF capacity case [10, 11, 13] and EDCA capacity problem [42] is due to the exclusion of other layers in the problem, since the capacity is defined as the asymptotic limit of MAC layer throughput, meaning traffic from the application layer is assumed to be saturated. But still, these analytical models are non-explicit, highly complicated and require numerical techniques for their solution. For other, more complex problems, when it is required to take into consideration *multiple factors* from other layers as well as QoS constraints, analytical models typically are difficult to derive and, hence, simulation-based modeling becomes the most applicable approach.

Let us look, for instance, at the question of *VoIP capacity over Wi-Fi*, namely, calculating the maximum number of VoIP phone calls supportable in a Wi-Fi network. Not only do the traffic characteristics of VoIP from the application layer need to be considered (which will affect the MAC access dynamics dramatically), but also the queueing performance such as delay and packet loss rate need to be included, in order to obtain satisfiable QoS constraints.

Although there are already quite a few analytical studies on the capacity of VoIP over Wi-Fi networks [43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54], our thorough review has led us to believe that they actually estimate the capacity in way which is overly optimistic, and thus offer rather loose upper bounds on the number of VoIP calls. This is because of two reasons: first, most of them do not consider the asymmetry of traffic conditions [1] and the correlation between traffic load and MAC access dynamics, therefore they adopt the methodology of $capacity\ of\ VoIP\ calls = \frac{capacity\ of\ a\ Wi-Fi\ network}{resource\ reuqested\ by\ one\ VoIP\ call}$. where the capacity in the numerator is mistakenly chosen as the capacity of Wi-Fi with one flow under saturation; second, no queueing performance metrics such as delay and packet loss constraints are taken into account. However, we argue that such capacity will never be reached in practice because a much lower limit will be encountered first, due to the non-negligible collisions and QoS constraints. This argument is also confirmed by simulation results.

Then, how to find a better estimate of the capacity of VoIP over Wi-Fi, one leading to a tighter bound, remains a key problem for a good design of call admission control (CAC) policies. The CAC based on an overly optimistic capacity estimation will result in admitting more VoIP phone calls than a Wi-Fi network can support, thus degrading the call qualities of all users. Hence, until a comprehensive analytical model considering all the factors mentioned above becomes available, simulation models are still the most reliable way to find the right answer.

In a Wi-Fi network, things will be more complicated when it is in ad-hoc mode instead of infrastructure mode, because a station in ad-hoc networks can act as a router and forward packets on behalf of its peers. The forwarding scheduler at the network layer is very important in arbitrating the resource between own generated traffic and forwarding traffic, and it posts another multi-layer complication on the system performance. There are some schedulers [55, 56, 57, 58, 59, 60] designed for stimulation of cooperation among possibly selfish parties in ad-hoc networks. But an

[1] For VoIP over Wi-Fi, the access point has a larger load than the stations because it talks with multiple stations in the downlink.

analytical model is not feasible due to the large dimensionality of the design space across multiple layers in such Wi-Fi settings, therefore simulation models are still the only practical way to evaluate the system performance.

However, an explicit mathematical model is always the most effective way to represent the system behavior and the most convenient way to facilitate embedding of QoS-aware MAC in network schemes, such as call admission control and scheduling schemes. Simulation models, although able to capture the system dynamics very closely, are time costly and only provide snapshots of the system over particular simulation configurations. Even the analytical model example from the Wi-Fi capacity problem does not offer a true closed form solution.

Therefore we are led to investigate an alternative method - namely, a metamodeling technique [61, 62, 63], in order to find an approximate mathematical representation of system performance in Wi-Fi networks. *Metamodeling*, described at times as a 'model of a model '[8], has been used by the simulation community to study the behavior of computer simulations for over thirty years and applied to many fields including manufacturing, industrial engineering, operation research, queueing models [64]. However, there are really few applications of metamodeling to the field of computer networks. Shaw [65] discussed the simulation and the metamodeling methodologies applied to the parallel processing environment. Barrett *et al.* [66, 67] and Vadde *et al.* [68] are the first two to use experiment design and statistical analysis to characterize the factor interactions among routing, MAC, mobility and load, in mobile ad-hoc network (MANET). But both works try to view the problem more from an engineering point of view, only capturing the capability of Response Surface Methodology (RSM) in interpolating the factor interactions of the model, without an awareness or mention of the superset of metamodeling methodologies in the background.

Our key methodological contribution in this Chapter is that we first advocate the application of metamodeling techniques to performance studies of Wi-Fi networks and build a framework of metamodeling network performance evaluation for Wi-Fi networks. Under this framework, we apply metamodeling techniques to three important Wi-Fi network case studies which need closed-form mathematical performance models, from the simplest one only involving MAC access dynamics, to the most complex one requiring interactions from application layer, network layer and MAC layer in an ad-hoc setting.

Our results in these three subcases prove the validity of our metamodeling methodology and provide useful insights into understanding Wi-Fi performance: our capacity metamodel of 802.11 DCF is validated by the well-known analytical model and shows an interesting log-linear relationship between capacity and number of users; our voice over Wi-Fi admission capacity metamodel give a much tighter bound than the existing bounds in the literature and composes a more practical admission control scheme; and the throughput metamodel, in a simple ad-hoc network, for the first time characterizes and quantifies the cross-layer effects between EDCA MAC layer and network layer. Our work, therefore, points out a new direction for the future performance studies of Wi-Fi networks.

The remainder of this chapter is organized as follows: First, in Section 4.2, we briefly introduce the background of metamodeling and formulate a general framework for metamodels of Wi-Fi network performance, in various settings. By applying these metamodeling techniques to three subcases, namely DCF capacity problem, VoIP capacity over Wi-Fi problem and the cross-layer scheduling in ad-hoc network, we show how metamodeling works and the validity of this novel application to the performance study of Wi-Fi networks, respectively in Section 4.3, Section 4.4 and Section 4.5. Finally, in Section 4.6, we present our conclusions.

4.2 Metamodeling Methodology

4.2.1 Metamodeling and Response Surface Methodology

As discussed in the introduction, we can build three types of performance models to study a Wi-Fi network. When the behavior of the system is easy to analyze, an analytical model is a good way to characterize the rules regulating the system. For some complex systems, analytical models may not be possible to find or too complex to solve, therefore simulation models are used instead to capture the system behavior.

An advantage of the simulation model over the analytical model is that it can simulate the system dynamics in great detail, thus providing a closer approximation to the system without the frequent need of assumptions required in the analytical model for facilitating the analysis. However, an explicit closed-form response function is most wanted to explore the whole response surface and develop optimizing strategies. A simulation model $S(\cdot)$ is simply not such a function, since it is just a mechanism that turns input parameters into output performance measures. If the analytical model $A(\cdot)$ of a system is not available or can not offer such a closed-form function, people turn to another modeling method, metamodeling, to find the solution.

Metamodel is defined as a mathematical closed-form model of the simulation model, or a 'model of the model' in [8]. The responses from the metamodel for inputs of X is Y_m with approximation error ε_m,

$$Y_m = M(X) = Y + \varepsilon_m$$

Metamodeling, the way of building a metamodel, involves: (1) simulation experiment design for generating data; (2) choosing a mathematical model to represent the data; (3) fitting the model to the observed data.

Experiment design:

An experiment design represents a sequence of experiments to be performed, expressed in terms of *factors* (design variables X) set at specified *levels* (predefined values) [8]. The most common experimental design is a 2^k full factorial design for k factors at 2 levels, which is also used for our metamodel. The other experimental designs include fractional factorial designs, orthogonal arrays, etc.

Model Choice:

After performing the necessary computer simulation runs according to the selected experimental design (a set of of values from the input X) and collecting output data (a set of Y_m) from the simulations, the next step is to choose a model $M(\cdot)$ to approximate the functional relationship between inputs X and outputs Y.

The metamodel can correspond to a response surface, or a neural network, or induction learning and Kringing, among others. The most prevalent models in the metamodeling literature are response surfaces and neural networks. Response surface models are normally first order, first order with interactions and second order polynomials as followings. The neural networks model presents in nonlinear format of $y = \frac{1}{1+e^{-(\beta+\sum w_i x_i)/T}}$.

$$y = \beta_0 + \sum_{i=1}^{m} \beta_i X_i \tag{4.1}$$

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \sum_{i=1}^{m} \sum_{j=1, i<j}^{m} \beta_{ij} X_i X_j \tag{4.2}$$

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \sum_{i=1}^{m} \beta_{ii} (X_i)^2 + \sum_{i=1}^{m} \sum_{j=1, i<j}^{m} \beta_{ij} X_i X_j \tag{4.3}$$

Model Fit:

To fit a model is to find the parameters of the model, e.g., βs in the response surface model. The fitting method varies according to different choice of models. To fit a response surface model, least square regression is used; while back-propagation can be applied to building a neural network model.

Response Surface Methodology (RSM):

The whole metamodeling methodology is further specified to fall under the *Response Surface Methodology*, or RSM, when the selected model is a response surface model. Myers *et al* [7] define RSM as 'a collection of tools in design or data analysis that enhance the exploration of a region of design variables in one or more responses'. The general RSM approach includes three steps: experiment design, build response surface model and least squares regression analysis. RSM consists of a group of techniques used in the empirical study of relationships between (one or more) response variables and many input variables. The techniques have been used to answer the key question of what values of the input variables (setting) will yield a maximum for the response variable.

In this chapter, we choose to apply RSM as our metamodeling methodology due to the reasonable number of factors in the model and to the well-established theory and techniques of response surface methodology.

4.2.2 Related Work

Although the term of metamodel was originated in 1987 by Kleijnen [69], the simulation community has used metamodeling techniques to study the behavior of computer simulations for over thirty years. Winter simulation conference proceedings provide good resources for tutorial and survey of issues in metamodeling, such as Barton series [62, 70, 71], Kelton series [72, 73, 61] and Sargent [74]. The most recent one can be seen in [8] by Simpson.

RSM is the most prevalent metamodeling methodology. But the technique itself was invented far earlier in 1951 by Box and Wilson [75] for study of a real non-simulation system. Due to the availability of cheap computing resources, vast simulations became feasible experiments, therefore resulting in the wide application of RSM to simulations in various disciplines, ranging from industrial engineering, operation research to aerospace and mechanical engineering. The methodology of RSM matured and advanced gradually, and the associated techniques consist of the topic for whole books [7, 76] or book chapters [77, 78]. A comprehensive review of RSM developments and applications from 1966-1988 is given in Myers *et al* [79]. Recent surveys can be seen in [80] and [81].

However, there are really few applications of metamodeling to the field of computer networks. Experiment design and analysis is presented only as alternative to simulation in [82] to study performance of computer systems, the book itself is a good resource for various performance analysis methods including experiment design, measurement, simulation and queueing modeling, though. Shaw [65] discussed the simulation and the metamodeling methodologies applied to the parallel processing environment.

Barrett *et al* [66, 67] are the first to use experiment design and statistical analysis to characterize the factor interactions among routing, MAC, mobility and load, in MANET. Subsequently, Vadde *et al* [68] studied the throughput and delay performance of MANET in the prospective of factor interactions among QoS architecture, routing protocol, MAC, load and mobility extensively. But both works try to view the problem more from an engineering point of view, only capturing the capability of RSM in interpolating the factor interactions of the model, without an awareness or mention of the superset of metamodeling methodologies in the background.

4.2.3 Formulation of Wi-Fi Metamodels

To enable increased application of metamodeling, including RSM, to the study of computer network performance, and to release its vast "horsepower", the core of our work is a key methodological contribution: that is, to first build a framework of metamodeling network performance evaluation for Wi-Fi networks. Under this framework, many problems can be formulated and studied systematically, for example, the interactions among different protocol layers, as in [66, 67, 68], and also in our case study III (Section 4.5), as well as performance problems such as capacity of the network in our case study I (Section 4.3) and case study II (Section 4.4).

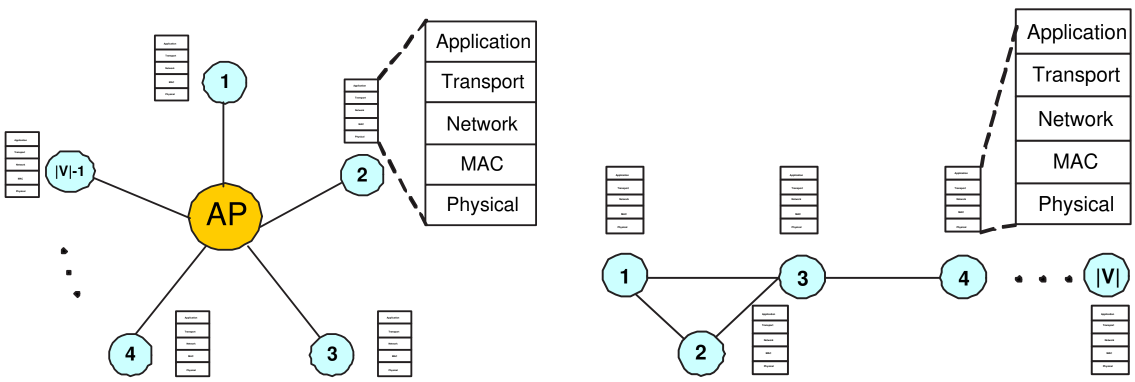
The system we are interested here is not a manufacturing workshop, an aircraft, or a computer processor, but a *Wi-Fi network*. Before we jump to the start of the design of experiments, choosing and fitting metamodels, which are standard procedures for every metamodeling problem, it is worth to first define the scope of the problem, including problem statement, network model, design variables (inputs) and performance metrics that we are interested in (outputs). Then the last step would be to evaluate the model and draw conclusion from the model which will help the understanding of Wi-Fi networks.

The complete procedure can be done sequentially as follows:

- **Define the scope of the problem**
  - **Problem statement:** clearly specify the question to be answered and

assumption to be made. The problem will determine the network model, what to input into the network and what to collect from the network. For example, if the problem is the capacity of DCF MAC in single hop infrastructure Wi-Fi with physical layer of 802.11a, 802.11b or 802.11g, the network model is going to be an infrastructure Wi-Fi, and we only care about the design variables in DCF MAC and physical layer without consideration of other higher layer parameters;

– **Network model**: a Wi-Fi network can be either in infrastructure mode or ad-hoc mode [5] and represented by a graph $G = \{V, E\}$. The number of nodes in the network is the number of vertices $|V|$ in the graph and the connections among nodes are represented in edges E. In infrastructure network, all traffic have to go through access point (AP) therefore results in only one hop in wireless domain; whereas in ad-hoc network, each node can route traffic for other nodes and each flow may pass multiple hops. Fig. 4.1 shows the network model examples for both modes.



(a) Infrastructure Mode (b) Ad-hoc Mode

Figure 4.1: General Network Model of Wi-Fi Networks

– **Inputs:** the design or controllable variables of each node $X = \{C_1, C_2, ..., C_{|V|}\}$, where C_i is not a scalar but a vector representing all protocol parameter set for node i.

$$C_i^T = \begin{bmatrix} application\ \ layer\ \ parameter\ \ set_i \\ transport\ \ layer\ \ parameter\ \ set_i \\ network\ \ layer\ \ parameter\ \ set_i \\ MAC\ \ layer\ \ parameter\ \ set_i \\ PHY\ \ layer\ \ parameter\ \ set_i \end{bmatrix}$$

* *Application layer*: traffic profiles defined in transmission specification (TSPEC) such as peak transmission rate, average transmission rate, etc.
* *Transport layer*: congestion window size etc.
* *Network layer*: routing protocol and scheduler schemes
* *MAC layer*: DCF access parameter like contention window size (CW) or EDCA access parameters such as arbitrary inter-frame space (AIFS).
* *Physical layer*: data transmission rate, overhead, *etc.*

If the system is symmetric, the input vector can be simplified to be just one parameter set $X = \{C\}$, which is shared by every node. In some problem, only a subset of layers are the focus of study, therefore, C_i^T may only include the parameter sets from these layers since parameters from other layers are fixed.

– **Outputs:** the desired performance metrics from the network, such as throughput, delay, and packet loss. $Y = \{P_1, P_2, ..., P_{|V|}\}$, where P_i is not a scalar but a vector representing all performance metrics of node i.

$$P_i^T = \begin{bmatrix} throughput_i \\ delayt_i \\ packet\ \ loss\ \ rate_i \\ ... \end{bmatrix}$$

- **Design of experiments**
- **Choose and fit the metamodel**

- **Evaluate the metamodel**

To summarize, the most crucial step in the framework is the first step, in which the scope of the problem is defined and the network model, inputs and outputs, are clearly specified. The subsequent design of experiments, and the choosing and fitting of models can be done following standard statistical approaches. The final step of evaluating the metamodel and the application of the metamodel is the most useful procedure which sheds light on the understanding and the design of Wi-Fi networks. We show in the following three sections how we apply this framework and methodology to three relevant sub-case studies of Wi-Fi performance evaluation, and the insights network designers can obtain from using the metamodels.

4.3 Case Study I: DCF Saturation Capacity

4.3.1 Case Background

Capacity, defined as the maximum throughput, is a fundamental characteristic of a network, which affects the network design, planning, and cost. Therefore, it remains a research focus of the study of access networks.

802.11 DCF MAC (CSMA/CA with BEB) is the random multiple access scheme a Wi-Fi network uses for radio access, which distinguishes Wi-Fi (802.11) from other access networks, such as Ethernet (802.3), and Token Ring (802.5). Hence, to study the capacity of Wi-Fi network requires performance analysis of 802.11 DCF MAC.

Accompanying the standardization and rapid deployment of IEEE 802.11 WLANs in 1990s, the performance analysis of its contention-based DCF MAC access function [5] has been studied extensively by analytical or numerical means in recent years. Among those analytical studies, three major performance models have been proposed in parallel, in order to analyze the saturation throughput/capacity performance: Assuming a constant collision probability for each station, Bianchi [10] proposed a Markov Chain to approximately model the behavior of CSMS/CA/BEB DCF, found the equilibrium packet transmission probability in a generic slot time by solving the Markov Chain, and finally obtained the saturation throughput by applying regenerative analysis to a generic slot time; Calì [11, 12] analyzed a p-persistent variant of DCF, with persistence factor p derived from the CW in DCF, then found the capacity similarly using renewal theory; Tay [13] used instead an *average value* mathematical model, in order to calculate the packet collision probability, and solved the maximum throughput in terms of collision probability. A variation of Bianchi's model was proposed by Wu in [14] for the further consideration of retry limits.

Bianchi's model in [10] is so widely adopted that most subsequent analytical models of EDCA (QoS enhanced MAC over DCF) and many adaptive backoff schemes are built on top of it. But one disadvantage of this model limits its use in practice: that is no close form solution is provided. Solving of nonlinear equations is required in derivation, which is time consuming and non-feasible for real time use.

The need of an approximate closed-form mathematical model of DCF capacity motivates us to apply metamodeling methodology in this case. The reason we put it as the first case study is due to the availability of validation through a well know analytical model.

4.3.2 Define the Scope of the Problem

Problem Statement

The question to be answered can be stated as the capacity or the maximum throughput of DCF MAC in an infrastructure Wi-Fi network. Since we only want to study the maximum capacity of the MAC layer, also defined as the asymptotic limit of DCF, we make assumptions typical in the literature related to DCF capacity [10, 12, 13] as follows:

- There are N nodes and one AP in an infrastructure mode Wi-Fi network. The N nodes and the AP are identical, they have the same type of traffic source, the same MAC access function and the same physical layer transmission rate. There are N flows coexisting, either in uplink from wireless station to AP, or in downlink from AP to wireless station.

- Traffic source is most aggressive and always keep sending packets. The packet size is fixed as a constant.
- There is one queue in MAC layer, always backlogged. The packet at the head of line uses DCF to access the radio channel.
- The channel is ideal. There is no channel error, no exposed or hidden node problem, and no propagation delay.
- There is no consideration of queueing overflow and QoS requirement. In other words, this capacity is a theoretical limit which will not be achieved in realistic situations.

Network Model

According to the problem statement, the network is an infrastructure Wi-Fi network with N nodes and one AP as shown in Fig. 4.2. Since our objective of the study is the capacity of 802.11 MAC/PHY, MAC and physical layer in the protocol stack are modeled together with a simplified application layer. The transport layer and network layer are excluded from the problem.

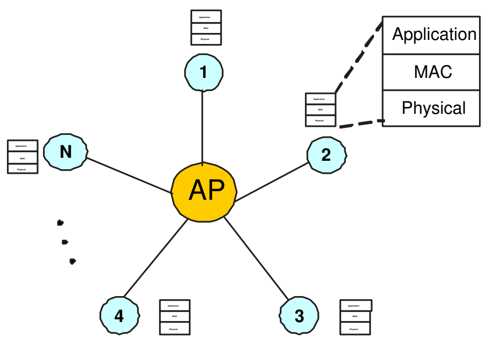


Figure 4.2: Network Model of Case Study I: an infrastructure Wi-Fi network.

Application Layer - Saturated Traffic: N identical traffic sources for N flows. For each traffic source, the packet size is a constant *packet_size* in units of bytes, the inter-arrival time between two packets is set small enough to achieve a saturation effect.

MAC Layer - IEEE 802.11 DCF: A DCF access backoff entity is implemented in MAC layer of each node to compete for radio channel. We assume RTS/CTS is used. Because the contention window size is the most important parameter in the backoff procedure, we are going to vary CW and study its effect on capacity in the experiments.

Physical layer (PHY) - IEEE 802.11b [5]: In the physical layer, we use the most popular 802.11b DSSS with a transmission rate of 11Mbps [2].

Inputs

The controllable or design variable is the value of the parameter minimum value of CW (CWmin) of DCF MAC layer in each station. Because all stations are identical (symmetric), we can further simplify the input to CWmin and the number N of stations in the network.

$$X = [N \;\; CWmin]$$

The other non-controllable or fixed inputs in the network are shown in Table 4.1. The data packet MAC layer sends to PHY layer is an MAC Protocol Data Unit (MPDU). The length of the MPDU depends on the type of frame this MPDU carries and also the length of MAC Service Data Unit (MSDU) frame the MAC layer receives from the higher layer.

Outputs

The output responses that we are interested in is the capacity of the network, i.e., the total throughput of all stations. It can either be normalized to the transmission rate, or in units of Mbps.

$$Y = S_{total} = N * S_{user}$$

[2]Other physical layer modes such as different rates of 802.11a and 802.11g can be easily substituted by changing the physical layer parameters including data transmission rate, Short Inter Frame Space (SIFS), Different Inter Frame Space (DIFS), a slot time (Tslot), physical layer overhead and the way MAC MPDU transmission time is calculated.

Table 4.1: Constants in Network of Case I

Application Layer		
	packet_size	1500 Bytes
MAC Layer		
	MAC_MPDU_TxTime	$\frac{8*MPDUlen}{TxRate}$ μs
	MPDUlen_Data	$34 + packet\_size$ Bytes
	MPDUlen_RTS	20 Bytes
	MPDUlen_CTS	14 Bytes
	MPDUlen_ACK	14 Bytes
Physical Layer		
	TxRate	11Mbps
	SIFS	10 μs
	DIFS	50 μs
	Tslot	20 μs
	PHYoverhead	192 μs

4.3.3 Design of Experiments

The throughput changes as the contention window size (CW) of each station changes and the total number of stations (N) competing for the medium varies. Therefore, we vary the values of CW and N as followings shown in Table 4.2. Note here we change both of CWmin and maximum value of CW (CWmax), but keep the number of backoff stage $\log_2^{(CWmax+1)/(CWmin+1)}$ constant as 5. Therefore we only show the change of CWmin in the table.

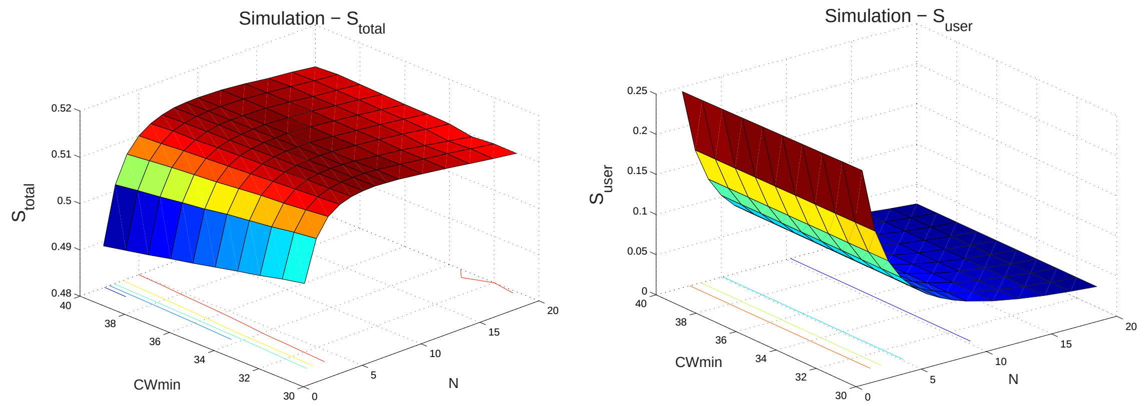
Table 4.2: Experiment Design Parameters for Case I

Factors	**Levels of Variation**	**Level Values**
$CWmin$	10	31,32,33,34,35,36,37,38,39,40
N	12	2,3,4,5,6,7,8,9,10,12,14,16

The levels of variation of CWmin and N are 10 and 12 respectively, resulting in 120 simulations. For each simulation setting, we run 10 replications in order to get satisfactory confidence intervals of the simulation results. The simulation model is implemented in ARENA and has been validated in our early work [83].

4.3.4 Choose and Fit the Metamodel

By observing the data we collect from the simulations for the settings shown in the table above, we can see that although S_{total} in Fig.4.3(a) changes with both CWmin and N, $S_{user} = S_{total}/N$ in Fig.4.3(b) is not affected by CWmin much.



(a) Normalized Total Throughput

(b) Normalized User Throughput

Figure 4.3: Response Surfaces of Throughput Values

But still S_{user} appears convex. By taking a logarithmic transformation, we can clearly see the linear trend of $ln(S_{user})$ over $ln(n)$ in Fig.4.4, which also shows the insignificance of CWmin. Therefore, we choose a metamodel based on a first order response surface model:

$$\ln(S_{user}) = \beta_0 + \beta_1 * \ln(N)$$

By running the SAS GLM [84] program over the transformed data set, we obtain $\beta_0 = -0.7013$ and $\beta_1 = -0.9852$, therefore the fitted metamodel is represented in Equation (4.4). The Analysis of Variance (ANOVA) statistics for the model and each factor is shown in Table 4.3. The high R-Square value of 0.999860 supports the goodness-of-fit of this model in interpreting the given data set.

$$\ln(S_{user}) = -0.7013 - 0.9852 * \ln(N) \tag{4.4}$$

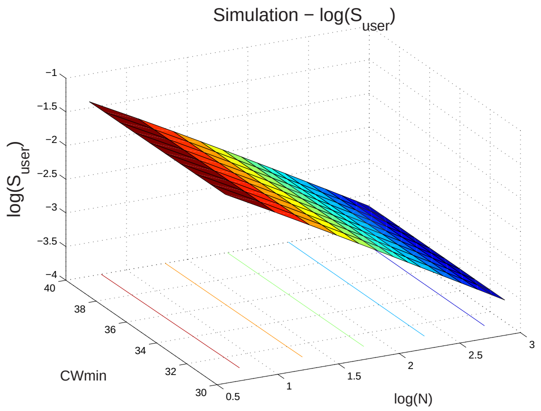


Figure 4.4: Response Surface after Transformation: $Log(S_{user})$

Table 4.3: ANOVA Table for $\ln(S_{user})$

Source	**Degrees of Freedom**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr>F**
Model	1	42.60659141	42.60659141	844225	<.0001
Error	118	0.00595526	0.00005047		
Corrected Total	119	42.61254667			
R-Square	**Coeff Var**	**Root MSE**	**Mean**		
0.999860	-0.274330	0.007104	-2.589622		
Parameter	**Estimate**	**Standard Error**	**t Value**	**Pr > $\|t\|$**	
Intercept	-0.7012883640	0.00215507	-325.41	<.0001	
logN	-0.9851761722	0.00107222	-918.82	<.0001	

Even after choosing a full model with $CWmin$ and the interaction term as in following equation, the newly fitted model only gives a tiny improvement over the previous one, with an R-square increment of 0.000015 from 0.999860 to 0.999875, but with at the expense of two more items.

$$\ln(S_{user}) = \beta_0 + \beta_1 * \ln(N) + \beta_2 CWmin + \beta_3 CWmin * \ln(N),$$

Therefore, we are content with the simple model in Equation (4.4), which is sufficiently accurate. By taking an exponential and transforming the normalized throughput back to absolute throughput in units of Mbps, we can obtain the final fitted capacity model for 802.11b 11Mbps DCF as

$$\begin{aligned} S_{user} &= 11 * e^{-0.7013-0.9852\ln(N)} = 5.4553 \cdot N^{-0.9852} \ \ (Mbps) => \\ S_{total} &= 11N * e^{-0.7013-0.9852\ln(N)} = 5.4553 \cdot N^{0.0148} \ \ (Mbps) \end{aligned} \tag{4.5}$$

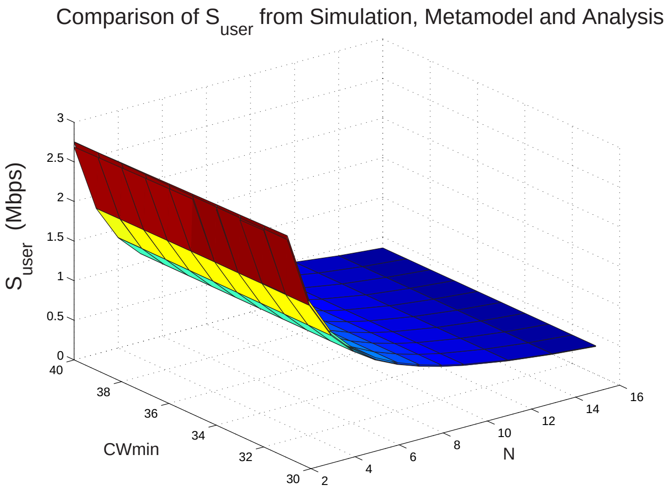
4.3.5 Evaluate the Metamodel: log-linear relationship

Validation

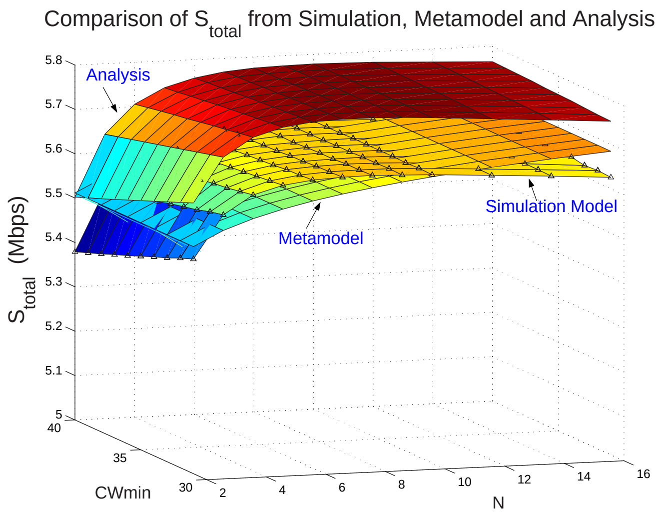
In order to validate the fitted model, we compare the throughput results from the metamodel to the simulation model and the analytical model, which is a revised model [32] over Bianchi's model [10], after correcting a small state consistency problem. From Fig. 4.5(a), we can see that the user throughput S_{user} response surfaces from the three models lie almost on top of each other and, in general, agree with each other very well.

A comparison of the total throughput S_{total}, which is the performance metric we want to study, is shown in Fig. 4.5(b). Although we can tell the difference among the three surfaces, the three models still agree with each other very well in a large scale, by noting that the scale of the graph is very small and ranges from 5 to 5.8 Mbps. Another insight we can draw from this graph is that the analytical model overestimates the capacity but characterizes the curvature of the response surface accurately. Our metamodel in some areas overestimates and in other areas underestimates the capacity, but overall, it captures the main relationship of S_{total} vs. N. But it does not model the curvature of the surface over the area of small number of users N very well due to its simplicity.

For further quantification of the predictability of the metamodel, we plot the residuals and prediction error rate in Fig. 4.6(a) and Fig. 4.6(b), respectively. These two graphs again confirm the insights we derived above. The difference between the analytical model and the simulation model is almost constant positive (constant means good surface curvature agreement and positive means overestimation). The difference between the metamodel and the simulation model is small and has both positive and negative values. And surely, the difference between the metamodel and



(a) Comparison of S_{user}



(b) Comparison of S_{total}

Figure 4.5: Comparison of S_{user} and S_{total} between the Metamodel, Simulation and Analysis

the analytical model is bigger than it with simulation model. But generally, all differences are small since the largest residual capacity is only 0.15 Mbps. The relative prediction error rate shown in Fig. 4.6(b) better proves the goodness of the metamodel with a maximum error rate of 3%, which means that in the worst case, the metamodel can approximate the simulation model with 97% accuracy.

Evaluation

An interesting message from this model is when the change of CW is small, the throughput is not affected much. In other words, the effect of CW is very mild. Therefore CWmin is ignored from the metamodel, and we will only analyze the sensitivity of the number of users in the network N in the network on throughput performance.

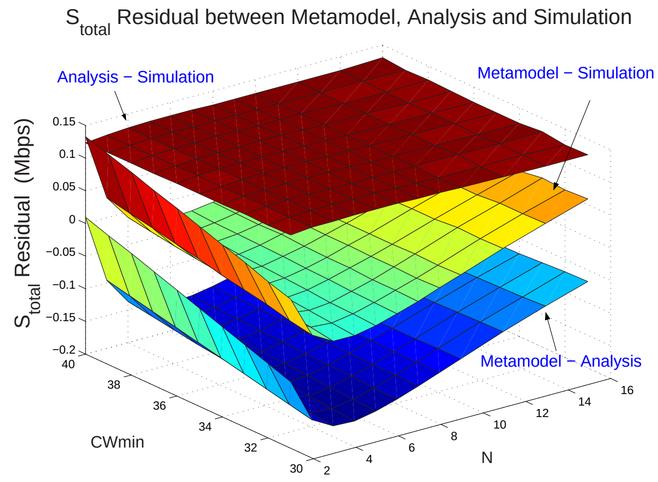
From the model of Equation (4.5), we see that S_{user} decreases with the rate of $r_1 = k^{-0.9852}$ and S_{total} increases with the rate of $r_2 = k^{0.0148}$, when N is increased by k times.

$$
\begin{aligned}
N' &= k * N => \\
S'_{user} &= 5.4553 \cdot (k * N)^{-0.9852} = S_{user}/k^{0.9852} = S_{user}/r_1 \;\; (Mbps) \\
S'_{total} &= 5.4553 \cdot (k * N)^{0.0148} = S_{total} \cdot k^{0.0148} = S_{total} * r_2 \;\; (Mbps)
\end{aligned}
$$

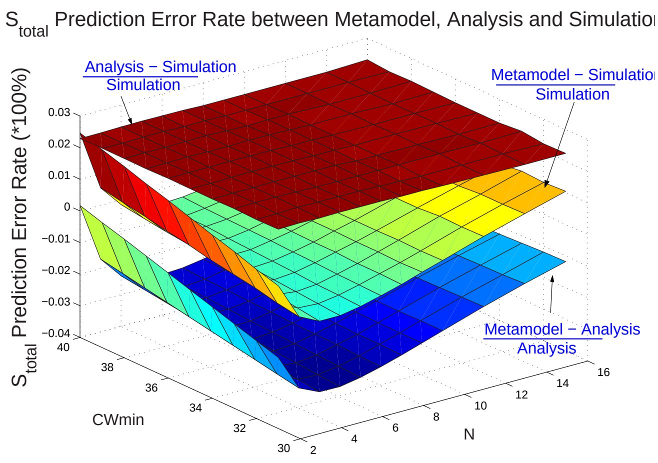
For the cases of $k = 2, ..., 10$, the decreasing rate r_1 and the increasing rate r_2 are calculated in Table 4.4. For example, when we double the number of users N, the throughput each user can obtain will be a little bit more than half of original one (divided by 1.9796), and the capacity of the network or the total throughput will only be multiplied by 1.0103, almost the same as the original one. In summary, the network capacity will increase slightly when the number of users increases because of the less wasted time on backoff caused by more intense competitions, and the share of each user will decrease just a little bit slower than the curve of $1/x$.

Engineering Approximation

The sensitivity analysis above intriguers us to explore the possibility of further approximation. Since $r_1 = k^{0.9852} \approx k$, and $r_2 = k^{0.0128} \approx 1$, we try to see if the



(a) Residual



(b) Prediction Error Rate

Figure 4.6: Prediction Error of Metamodel of S_{total}

Table 4.4: Sensitivity Analysis of Metamodel

Multiplier k	**Divisor** $r_1 = k^{0.9852}$	**Multiplier** $r_2 = k^{0.0148}$
2	1.9796	1.0103
3	2.9516	1.0164
4	3.9188	1.0207
5	4.8823	1.0241
6	5.8430	1.0269
7	6.8013	1.0292
8	7.7575	1.0313
9	8.7120	1.0331
10	9.6650	1.0347

following math model can approximate the throughput well or not.

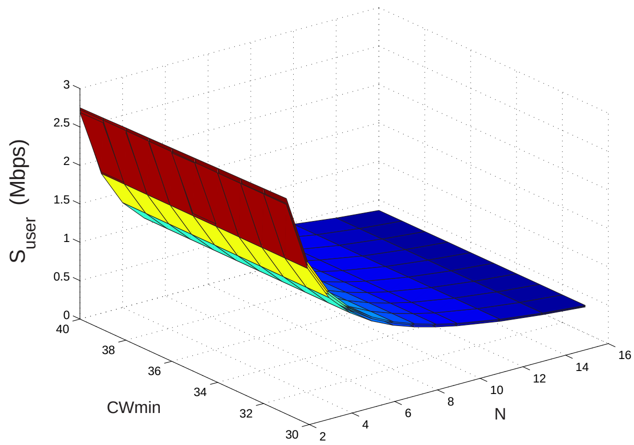
$$\begin{aligned} \overline{S_{user}} &= 5.4553 \cdot N^{-0.9852} \approx 5.4553/N \ \ (Mbps) \\ \overline{S_{total}} &= 5.4553 \cdot N^{0.0148} \approx 5.4553 \ \ (Mbps) \end{aligned} \tag{4.6}$$

Comparisons of the throughput results between the new model, the simulation model and the analytical model are shown in Fig. 4.7. The close match of S_{user} among the three surfaces is still as good as in the previous one. Although the S_{total} from the new model is a constant plane, it still approximates the simulation model well. The reason behind this phenomenon is that the capacity increases with N so lightly that we can approximate the network capacity to be a constant as the throughput when there is only one user, then the throughput each user can get will just be simply this constant shared by the number of users.

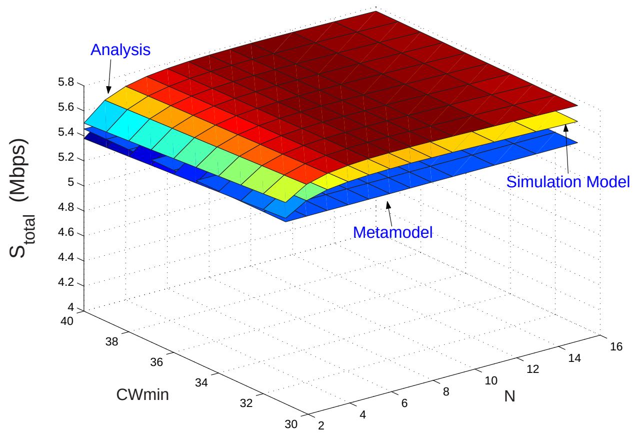
4.4 Case Study II: VoIP Admission Capacity

4.4.1 Case Background

Voice over IP (VoIP) has been witnessing massive growth over the past two years. Some people project that VoIP over Wi-Fi is going to be the next major disruptive application (or "killer application"). The market statistics confirm this: worldwide Wi-Fi VoIP handset revenue totaled 54.7 million in 2004 and units totaled $143,000$,



(a) Comparison of S_{user}



(b) Comparison of S_{total}

Figure 4.7: Comparison of S_{user} and S_{total} from NEW Metamodel, Simulation and Analysis

and strong growth is expected at least through 2009 as steady adoption of voice over Wi-Fi continues, according to Infonetics Research's latest report.

However, the delivery of solid end-to-end voice quality remains a big challenge for network designers and researchers. Nowadays, the only practical VoIP network planning method is over-engineering, which means to provide over-abundant bandwidth, thus underutilizing the network. But a well designed call admission control (CAC) scheme can customize the network bandwidth according to usage model, therefore saving the network resources and .

Therefore, the study of VoIP capacity over Wi-Fi access networks, constitutes a main question to answer before designing a CAC, and has been a research topic for the last few years. Although there are already quite a few analytical studies related to the capacity of VoIP over Wi-Fi networks [43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54], after thoroughly reviewing them, we find that they actually estimate the capacity in an overly optimistic way and offer rather loose upper bounds of the number of VoIP calls. This is because of two reasons: first, most of them do not consider the asymmetry of traffic conditions and the correlation between traffic load and MAC access dynamics, therefore they adopt the methodology of $capacity\ of\ VoIP\ calls = \frac{capacity\ of\ a\ Wi-Fi\ network}{resource\ reuqested\ by\ one\ VoIP\ call}$. where the capacity in the numerator is mistakenly chosen as the capacity of Wi-Fi with one flow under saturation; second, no queueing performance metrics such as delay and packet loss constraints are taken into account. However, we argue that such capacity will never be reached in practice because a much lower limit will be encountered first, due to the non-negligible collisions and QoS constraints. This argument is also confirmed by simulation results.

Under-utilization is not good, over-utilization is even worse, because the CAC based on overly optimistic capacity estimation will result in admitting of more VoIP phone calls than a Wi-Fi network can support thus degrading the call qualities of all users. Hence, how to find a better estimation of the capacity of VoIP over Wi-Fi with a tighter bound remains a key problem for a good design of call admission control (CAC) policies.

Before a comprehensive analytical model considering all the factors mentioned above becomes available, simulation models are still the most reliable way to find the right answer, but with limited power in formulating the solution. Therefore, it again motives us to apply a metamodeling methodology to find and characterize tighter VoIP bounds on the VoIP capacity over Wi-Fi.

4.4.2 Define the Scope of the Problem

Problem Statement

The question to be answered is to calculate the maximum number of VoIP calls that can be supported in an infrastructure Wi-Fi network with satisfactory QoS performance of one way delay less than $100ms$ and a packet loss rate less than 2%. This question can also be stated in another way, namely, 'Can I add this new VoIP call?' as in the title of paper [51]. If this new VoIP call receives acceptable QoS after being admitted into the network but also does not affect the QoS of existing calls, while adding one more call will result in the opposite, we will then say this call can be added and the number of existing calls plus this one is the maximum number of VoIP calls supportable, and is the VoIP capacity of the network.

Network Model

The network model in this case is similar to the one in Case I, and shown in Fig. 4.8. The network is an infrastructure Wi-Fi network with one AP and multiple nodes. VoIP is the application carried in the network, therefore, application layer, simple RTP/UDP/IP layers which add the VoIP headers, and 802.11 MAC and PHY layers are models in the protocol stack of each station and AP. The number of nodes in the network N is not fixed, since the maximum number of nodes, denoted as N^*, is the parameter we want to know.

Application Layer - G711 VoIP Traffic: There are N pairs of VoIP bidirectional calls. Each node carries an uplink VoIP flow to AP, and the AP carries N VoIP downlink flows for N stations. For each call, we use the ITU G711 64kbps codec where frames are sent out every time interval of $VoicePktIntvl$ ms. The packet size will vary according to the packet interval to keep the data rate constant at $64Kbps$.

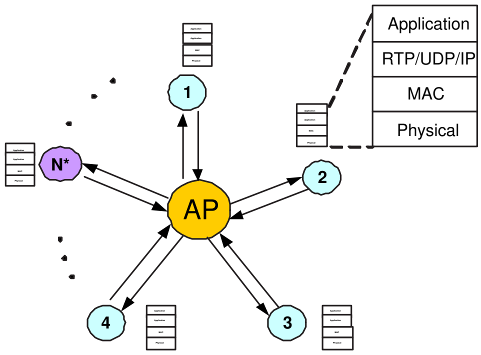


Figure 4.8: Network Model of Case Study II: an infrastructure Wi-Fi network with VoIP Traffic.

For example, if the packet interval $VoicePktIntvl = 20ms$, the packet size will be $64k * 20ms = 1280bits = 160Bytes$. G711 provides the best quality among the often used VoIP codecs, among them G711, G729 and G723.1, since it occupies the largest data rate. But due to its widest use in VoIP networks, we will illustrate our VoIP capacity experiment with G711 traffic without loss of generality. For more detailed VoIP codec algorithms the reader is referred to book [85].

RTP/UDP/IP Layer - VoIP headers: The standard method of transporting voice samples through an IP based network requires the addition of three headers. These headers are IP, UDP and RTP headers. An IPv4 IP header consists of 20 bytes; a UDP header has 8 bytes and a RTP header has 12 bytes, therefore the total length of the header is 40 bytes.

MAC Layer - IEEE 802.11 DCF: A DCF access backoff entity is implemented in the MAC layer of each node to compete for radio channel. We assume the basic access mode is used without RTS/CTS overhead, since the packet size of VoIP traffic is small enough and below the threshold for activation RTS/CTS scheme. In this experiment, all DCF parameters include DIFS and CWmin are fixed.

PHY layer - IEEE 802.11b: In physical layer, we use the 802.11b DSSS with variable data transmission rates of 1, 2, 5.5, and 11Mbps.

Inputs

The controllable or design variables of interest are two: the VoIP voice sample interval and the PHY data rate.

$$X = [VoicePktIntvl \;\; DataRate]$$

The other non-controllable or fixed inputs in the network are shown in Table 4.5.

Table 4.5: Constants in Network of Case II

Application Layer		
	G711 traffic_rate	64 kbps
RTP/UDP/IP Layers		
	VoIP header	40 bytes
MAC Layer		
	MAC_MPDU_TxTime	$\frac{8*MPDUlen}{DataTxRate}$ μs
	MAC header	34 Bytes
Physical Layer		
	CWmin	31
	CWmax	1023
	SIFS	10 μs
	DIFS	50 μs
	Tslot	20 μs
	PHYoverhead	192 μs

Outputs

The output responses that we are interested in is the VoIP capacity of the network, i.e., the maximum number of VoIP calls that can be supported in a Wi-Fi network with satisfactory QoS metrics, i.e., one way delay less than 100 ms and packet loss rate less than 2%.

$$Y = N^*$$

4.4.3 Design of Experiments

The output capacity will vary as the controllable input variables change. Table 4.6 shows the parameters, voice sample interval ($VoicePktIntvl$) and the data transmission rate ($DataTxRate$), for our designed experiments. For the G711 codec, the normal used sample duration is tens of milliseconds, and the data transmission rate of 802.11b in the standard can only be 1, 2, 5.5 and 11 Mbps.

Table 4.6: Experiment Design Parameters for Case II

Factors	**Levels of Variation**	**Level Values**
$VoicePktIntvl$ (ms)	10	$10, 20, 30, 40, 50, 60, 70, 80, 90, 100$
$DataTxRate$ (Mbps)	4	$1, 2, 5.5, 11$

The levels of variation of $VoicePktIntvl$ and $DataTxRate$ are 10 and 4 respectively, resulting in 40 simulations. For each simulation setting, we run 20 replications in order to obtain satisfactory confidence intervals of the simulation results. And the simulation duration of each run is set to 3 minutes in order to emulate the normal calling time. We set the queue size of each MAC queue to be 30 packets.

The simulation model is implemented in ARENA. For each design combination of $VoicePktIntvl$ and $DataTxRate$, we actually run a series of simulation scenarios with the number of nodes or VoIP call pairs incrementally set to be from 1 to 20. Then we find the maximum number of VoIP pairs N^* as the output, for which the simulated packet loss rate is less than 2% and maximum one way delay is less than $100ms$. The data set we collected from the simulations is shown in Table 4.7.

4.4.4 Choose and Fit the Metamodel

First, we choose a first order polynomial RSM model with interactions as Equation (4.7). The ANOVA results from SAS GLM show that the R-square of the fitted model is only 0.864488.

$$N^* = \beta_0 + \beta_1 * VoicePktIntvl + \beta_2 DataTxRate + \beta_3 VoicePktIntvl * DataTxRate \quad (4.7)$$

Table 4.7: Simulation Outputs N^* for Case II

$VoicePktIntvl$ **(ms)**	**1Mbps**	**2Mbps**	**5.5Mbps**	**11Mbps**
10	2	2	4	4
20	2	4	7	8
30	3	5	9	11
40	3	6	11	12
50	3	6	12	13
60	4	7	12	14
70	4	7	12	15
80	4	7	13	15
90	4	7	13	15
100	4	8	14	15

Inspired by the logarithmic transformation in Case I, we also try to fit the seconde model as shown in Eq. (4.8), for which every variable is logarithmically transformed. The new model results in a much higher R-square value of 0.950476. The ANOVA table is shown in Table 4.8.

$$\begin{aligned} ln(N^*) &= \beta_0 + \beta_1 * ln(VoicePktIntvl) + \beta_2 ln(DataTxRate) \\ &+ \beta_3 ln(VoicePktIntvl) * ln(DataTxRate) \end{aligned} \tag{4.8}$$

Although the item $log(DataTxRate)$ seems insignificant since its t test is larger than 0.05, it has to be kept in the model because the corresponding interaction item is significant. Therefore, we pick the second model as our final RSM. By plugging $\beta_0 = -.2518396194$, $\beta_1 = 0.4033792418$, $\beta_2 = 0.1856925928$ and $\beta_3 = 0.0973528574$ into the model, and then taking an exponential transformation of $log(N^*)$ back to absolute values, we can obtain the final fitted VoIP capacity model for 802.11b as

$$\begin{aligned} N^* &= e^{-0.2518+0.4034\ln(T)+0.1857\ln(D)+0.0923\ln(T)\ln(D)} \\ &= 0.7774 \cdot T^{0.4034} \cdot D^{0.1857} \cdot e^{0.0923\ln(T)\ln(D)} \end{aligned} \tag{4.9}$$

where T denotes $VoicePktIntvl$ and D denotes $DataTxRate$.

Table 4.8: ANOVA Table for Model Eq. (4.8) of Case II

Source	**Degrees of Freedom**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr>F**
Model	3	15.89100105	5.29700035	230.31	<.0001
Error	36	0.82799663	0.02299991		
Corrected Total	39	16.71899768			
R-Square	**Coeff Var**	**Root MSE**	**Mean**		
0.950476	7.761563	0.151657	1.953952		
Parameter	**Estimate**	**Standard Error**	**t Value**	**Pr > $\|t\|$**	
Intercept	-.2518396194	0.21952086	-1.15	0.2589	
lVoicePktIntvl	0.4033792418	0.05663709	7.12	<.0001	
lDataTxRate	0.1856925928	0.14524974	1.28	0.2093	
lVoicePkt* lDataTxRate	0.0973528574	0.03747490	2.60	0.0135	

4.4.5 Evaluate the Metamodel: You can NOT add this VoIP call

In order to validate the fitted model, we compare the throughput results from the metamodel to the simulation model and the analytical model in [52, 45]. From Fig. 4.9, we can see the differences among three surfaces, especially over the area of big voice packet intervals and big data transmission rates. It is obvious that our metamodel agrees with the simulation model better than the analytical model. Why? Is this because our simulation model is not valid, or because the analytical model is too optimistic?

In the following sections, we first validate our simulation model, then explain why the capacity bound calculated by the analytical model is too loose and not realistic. Finally we propose a new call admission control scheme based on our metamodel which gives a much tighter bound on the VoIP capacity.

Validate Our Simulation Model

After thorough review, we have come to the conclusion that the analysis methodology adopted by many VoIP capacity works is the same:

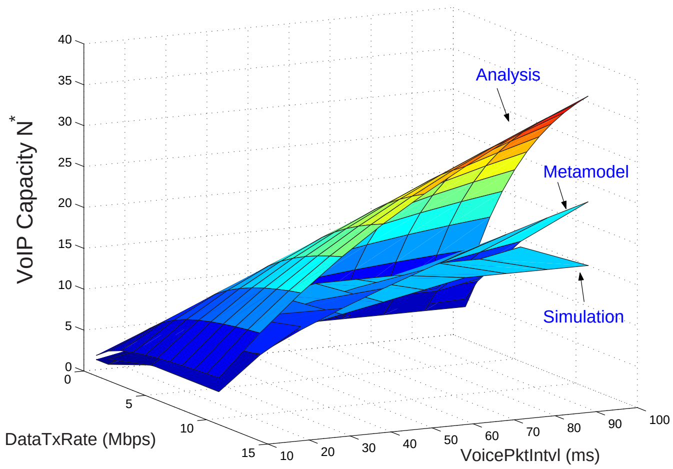


Figure 4.9: Comparison of N^* from Metamodel, Simulation and Analysis

$$capacity\ of\ VoIP\ calls = \frac{capacity\ of\ a\ Wi-Fi\ network}{resource\ requested\ by\ one\ VoIP\ call} \tag{4.10}$$

Here the denominator is easy to calculate. It depends on the voice codec scheme. For G711, the voice sampling rate of 8 K results in a data rate of 64 Kbps. Plus RTP/UDP/IP header of 40 bytes of each packet for every $VoicePktIntvl$ ms, the final bandwidth requested by one VoIP call is $64 + 40 * 8/VoicePktIntvl$ Kbps. For example, if $VoicePktIntvl = 10$ ms, then the bandwidth per G711 VoIP call is 96 Kbps.

The key part is the estimation of Wi-Fi network capacity in the numerator. Garg [51] proposes to use the saturated network throughput 'when there are exactly two active senders', i.e., $S_{total}(N = 2)$. Medepalli [45] assumes a saturated throughput when 'WLAN consists of only the station of interest', i.e., $S_{total}(N = 1)$. The rationale behind this assumption is Hole's statement in [52] that collision probability in infrastructure based WLANs is seen to be small, especially for VoIP traffic. The same argument can be seen in Wang's JSAC paper [48], namely that the possibility of collisions and the increase of backoff time in retransmissions are ignored.

Before we argue this network capacity estimation is wrong, we first prove our simulation model is valid by comparing the throughput results from our simulation to the results calculated from the analytical model of $S_{total}(N = 1)$. From Fig. 4.10, we can see that the network capacity or the maximum achievable throughput from analysis agrees very well with the simulation results under the same assumption of saturated traffic, only one user and no consideration of QoS constraints. An side insight we obtain from this graph is that logarithm of this capacity is in linear relationship with the logarithm of voice packet interval, which determines the packet size.

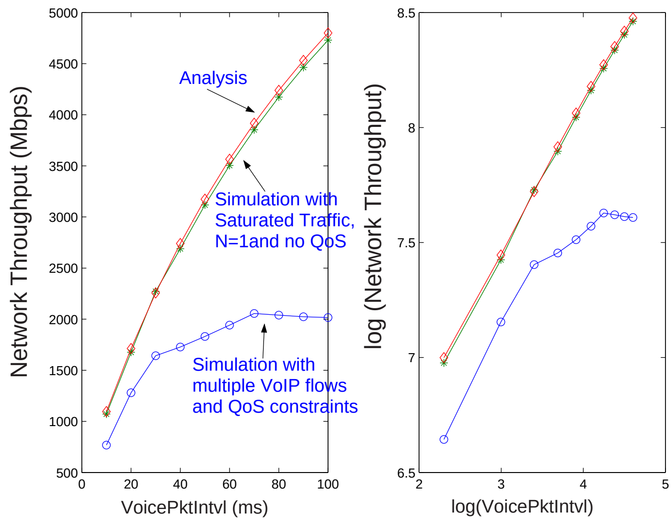


Figure 4.10: Network Capacity from Analysis, Simulated Saturation Throughput and Simulation (DataTxRate=11Mbps)

But one most important message we can derive is not our simulation model is validated, but is that the real VoIP network capacity from the simulation with the consideration of QoS constraints (packet loss rate less than 2% and delay less than $100ms$) is way below the estimation from the analysis.

Why is the Analytical Bound so Optimistic?

We know from the the former section of Case Study I that the capacity is dependant on the number of users and the traffic conditions. Under saturated traffic condition, we proved through the last metamodel in Eq. (4.6) that S_{total} can be approximated by a constant. Therefore both approximations of $S_{total}(N = 1)$ or $S_{total}(N = 2)$ will be good to estimate the network capacity. But the question is, can we still use this network capacity under saturated traffic conditions in order to estimate the network capacity under VoIP traffic conditions in an infrastructure mode Wi-Fi network with QoS constraints?

The answer is *no*. There are two reasons for this: First, the VoIP traffic is asymmetric. As illustrated in Fig. 4.8, there are N flows out of the AP but only one flow coming out of a station. This asymmetric traffic load correlates with MAC access dynamics, and results in the asymmetric throughput for uplink and downlink shown in Fig. 4.11, if we zoom in and observe the simulations of $DataTxRate = 11Mbps$ and $VoicePktIntvl = 10ms$ while we change the number of VoIP calls. We find out from the plots that throughput per flow in the downlink from the AP degrades a lot when the number of VoIP calls reaches five, then keeps decreasing while the throughput per flow in uplink from each station keeps steady at the load level (96Kbps). The rationale behind this is that the AP has only one MAC queue and one backoff access entity competing with the other N stations. When the system is under-loaded, the throughput AP achieves can be higher than the throughput each station receives. But when the system is overloaded, due to the fair nature of DCF access scheduling, the total throughput carried by AP in the downlink will converge to the throughput of each station.

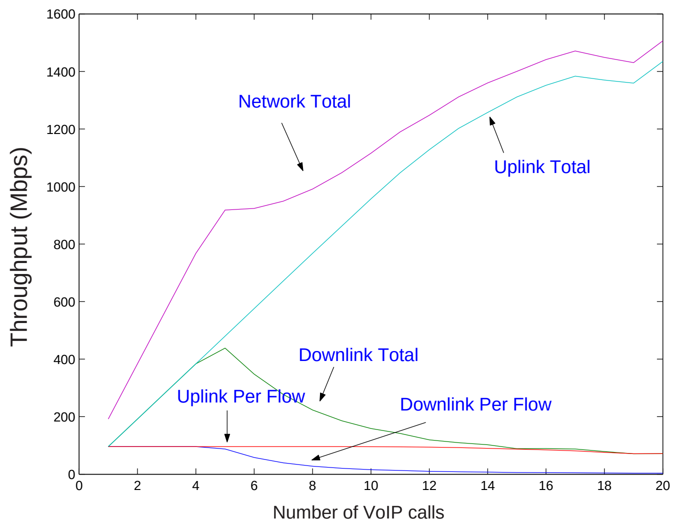


Figure 4.11: Throughputs in Uplink, Downlink and the Network (DataTxRate=11Mbps, VoicePktIntvl=10ms)

Second, no queueing performance metrics such as delay and packet loss constraints are taken into account. However, as we can see from the figure above, when N is larger than 4, the packet loss rate will be too high, resulting in degraded call quality. Therefore, the maximum number of VoIP calls can be supported in such a Wi-Fi network will only be four, smaller than the capacity predicted by the analysis of 5.58. When the voice packet interval is larger, this difference will be more obvious.

New CAC with Tighter Bound

Therefore, we argue that our metamodel supports a tighter bound than the analytical models in the literature. When there is a new VoIP call trying to apply for the service in the network with already N users, if our metamodel tells us that $N+1$ is larger than our predicted capacity, then we deny the application, otherwise, we accept it. But for the same case, the CAC based on the analytical model will accept the new call wrongly, resulting in service degradation of all the current users plus the new one.

4.5 Case Study III: Metamodel for Cross-layer Scheduler in ad-hoc Wi-Fi

4.5.1 Case Background

Motivation

The issue of *cooperation* has received a lot of attention in the context of mobile multihop ad-hoc wireless LAN networks. In an ad-hoc network, participating nodes may be self-interested - they may selfishly turn down the forwarding traffic flow coming from the other nodes and only transmit their own generated traffic for the reason of saving energy. But if every node performs in this manner, no traffic can traverse multiple hops, and network throughput will degrade unacceptably. Therefore, a good utilization of the scarce wireless resources in the whole ad-hoc network depends on the cooperation among participating parties. How to stimulate the cooperation, thus, is a crucial issue in non-cooperative mobile ad-hoc networks.

The existing cooperation stimulation schemes are mostly designed within the *network layer*, either by pricing-based or reputation-based approaches. The core concept of pricing-based schemes is to reward the forwarding or relaying behavior and punish the self transmission either in virtual money [55], [57] or other type of credit [56]. In reputation-based schemes [58, 59, 60], a reputation system is maintained by neighborhood monitoring, thus a misbehaving node with bad reputation can be detected and avoided.

Although these cooperation schemes are implemented in WLANs, they do not consider the inherent properties of wireless channels, i.e., they do not consider medium access control (MAC) effects. Some simulation evaluations have been done but typically in a generic network infrastructure and without a real MAC at the bottom.

However, in packet radio networks, especially in mobile ad-hoc networks, the medium access protocol mainly determines the sharing pattern of the radio channel. Hence, different than in wired networks, the MAC cannot be omitted from studies of cooperation in such wireless networks.

Only few reports [86, 87] have appeared studying the misbehavior and cooperation problem solely at *MAC layer* in WLAN. These articles explore the binary backoff function of the 802.11 Distributed Coordination Function (DCF) MAC and assume a variable contention window (CW) size. However, simply to prevent the misbehavior from the MAC point of view, by changing the backoff parameters, is limited without *simultaneous control* at the network layer.

Therefore, we argue that cooperation in mobile ad-hoc WLAN is a fundamentally *cross-layer* issue. Neither a purely network layer cooperation nor a solely MAC layer cooperation can achieve the best channel utilization, therefore considering both layers is a must. The promise of cross-layer cooperation enforcement is also forecast in [88] and the same author describes a cross-layer framework in [89].

The objective of cross-layer design in mobile ad-hoc networks is to optimize and exploit the cross-layer interdependencies in order to enhance the performance of the network as a whole. Although many attempts at cross-layer designs, such as [90, 91, 92, 93, 94, 95] and [96], can be found in the literature, there are very few quantitative measurements of cross-layer effects. The reason is that such cross-layer effects are very difficult to capture analytically as the combined cross-layer performance function of the system is intractable.

Therefore we are led to investigate a metamodeling technique [61, 62, 63], in order to find an approximate mathematical function of system performance in terms of the cross-layer design parameters and subsequently quantify the cross-layer effects through the evaluation of interaction terms in the model.

Most of the work in the MAC layer only considers the DCF function in which there is only one access entity per node; however, multiple access entities can be supported in the newly QoS-enhanced 802.11e EDCA. Therefore, how to stimulate cooperation in network layer on top of EDCA MAC layer presents a currently unaddressed question.

Motivated by these needs, in this section[3], we study the cross-layer cooperation consisting of a network layer priority scheduler extended from [55] and a MAC layer EDCA priority access scheduler in a mobile ad-hoc network, by applying a metamodeling technique. Our contributions are three-fold: first, we advocate the multidisciplinary use of metamodeling in cross-layer design; second, we provide a metamodel of system throughput in functions of cross-layer cooperation parameters both in network layer and MAC layer; third, we quantify the cross-layer effects between MAC and network layer and bring additional insights into the understanding of cross-layer design.

Related Work

Network Layer Cooperation A virtual currency (or “nuglet”) counter is proposed in [55] to pay for each packet locally generated, and also to be earned by forwarding packets on behalf of other nodes. Only if the nuglet counter is positive, can the node send its own packet. Upon forwarding a packet, the nuglet counter increases by one, while it decreases by the number of hops for transmitting a locally

[3]The work done is this section was also published as the technical report of Center for Advance Computing and Communications in [97].

generated packet. The major limitation of this scheme is the unfair treatment of the edge nodes who cannot pay for their own transmission because of seldom forwarding requests, in addition to the need for a temper-proof hardware module to protect the nuglet counter.

In contract to using a universal utility metric, [57] proposes a layered scheme consisting of a policed best-effort service and a incentive-based priority forwarding: nodes get compensated for forwarding priority packets and nodes are unaffected if they do not forward packets in a priority fashion.

"Sprite", a centralized credit system in [56], determines charge and credit from a game-theoretic perspective and motivates each node to report its actions honestly. There is no need for temper-proof hardware in this scheme, however, some out-of-band mechanism is required for communication of the credits.

The first reputation based scheme is introduced in [58], in which a watchdog identifies misbehaving nodes by performing a neighborhood monitoring and a reputation system keeps track of reputations of each node. Another reputation-based scheme called "CONFIDENT" is introduced in [59]. In CONFIDENT, a node monitor the routing and forwarding behaviors of its neighbors and take reputation record and trust records, then send alarms to isolate bad nodes upon detecting misbehavior. However, it may degrade the network utilization by introducing significant reputation propagation overhead and by overloading the well behaving nodes. A reliability index-based approach [60] takes into account not only the presence of possible selfish/malicious nodes but also situations like congestion and wireless lossy links.

MAC Layer Cooperation: MAC greediness, reflected in a smaller backoff interval, is detected by receivers and corrected by enforcing a bigger value in [86]. But it requires modification of the standard and also assumes nonrealistic traffic always in uplink. A game theoretic scheme for CSMA/CA schemes is presented in [87]. It shows how a Pareto-optimal point is achieved in a dynamic game by adaptively changing the contention window size and misbehavior being penalized by jamming.

4.5.2 Define the Scope of the Problem

Problem Statement

To characterize the cross-layer interaction between application layer load, network layer cooperation scheduler and MAC layer access scheduler in an example ad-hoc Wi-Fi network.

Network Model

For simplicity of explanation and to illustrate our approach, we pick a small example ad-hoc WLAN with three stations as our network model as shown in Fig. 4.12. The insight of using metamodeling to study the cross-layer cooperation can still be applied to any bigger and more complicated networks. In this ad-hoc WLAN, there are three wireless stations (WS). WS2 is located in between WS1 and WS3, and can talk to both of them. But WS1 and WS3 can not reach each other. Two traffic flows compete for the resources in this network: flow 1 is from WS1 to WS3 which has to be relayed at WS2, flow 2 is from WS2 to WS3 directly.

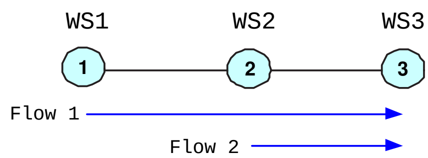


Figure 4.12: Network Model of Case Study III: an small ad-hoc Wi-Fi network.

Our objective of this study is the interaction among three layers, namely, the application layer, network layer and MAC layer. We assume that the transport layer and physical layer parameters are fixed, therefore excluded from our input.

Application Layer: Traffic Profile

We assume an exponentially distributed traffic inter-arrival profile in this ad-hoc network. For flow one, we define $L1$ as the exponentially distributed arrival rate of traffic in WS1. For flow two, we define $L2$ as the exponential rate of traffic generation in WS2. In our network, we assign $L1 = L2 = load$. Note that the load is normalized in terms of the physical layer transmission rate of 802.11.

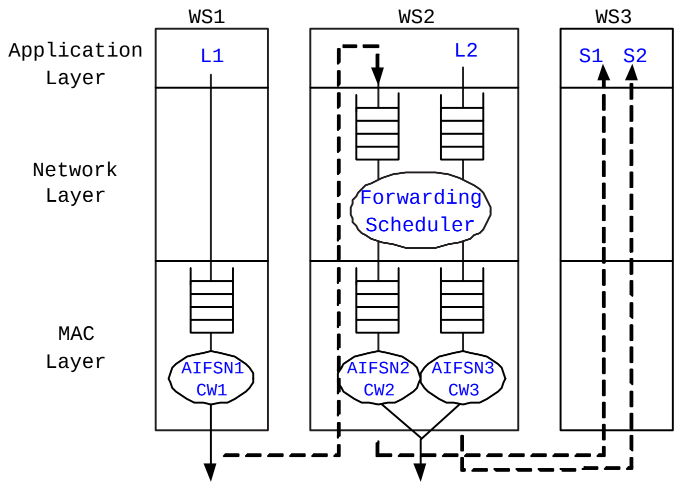


Figure 4.13: Case III: layer configurations.

Network Layer: cooperation forwarding scheduler

Two flows coexist at WS2. At the network layer, WS2 uses a forwarding scheduler to determine the priorities between forwarding traffic and own traffic. Since the station may be selfish and try to maximize its own throughput, cooperation needs to be stimulated and selfishness needs to be punished by setting the appropriate forwarding rule.

There are some scheduler schemes designed in the literature [55, 57, 56]. In this chapter, we design our own priority forwarding scheduler by extending the scheme in [55] with a generalized *award* and *punishment* and an upper threshold ($UpThrshd$) and a lower threshold ($LoThrshd$) over the counter. Table 4.9 shows the scheduling rule, where K is a constant and $UpThrshd$, $LoThrshd$, *award* and *punish* are four control variables.

The advantages of our scheduler are the flexible increasing slope (award) and the decreasing slope (punish) of the counter and also the flexible resources arbitration between forwarding and self traffic by adjusting the two thresholds of the counter.

MAC Layer: IEEE 802.11e EDCA scheduler

We implement an EDCA priority access scheduler at MAC layer to arbitrate

Table 4.9: Network Layer Cooperation Scheduler Scheme.

Virtual money counter N2
WS2 maintains a counter N2. Initially, N2=K
Scheduling Rule
IF $N2 > UpThrshd$,
WS2 only sends its own traffic and does not forward traffic;
ELSEIF $LoThrshd < N2 < UpThrshd$,
WS2 sends its own traffic and also forwards traffic;
ELSEIF $N2 < LoThrshd$,
WS2 only forwards traffic for WS1, and not send its own.
ENDIF
Update Rule of N2
IF the transmitted packet belongs to forwarding traffic,
$N2 = N2 + award$.
ELSEIF the transmitted packet belongs to its own traffic,
$N2 = N2 - punish$.
ENDIF

the radio channel resource to multiple access entities by manipulating their EDCA parameters, include AIFSN, CWmin, CWmax and TXOP.

In our experimental network, there are three access entities sharing the wireless medium. Access entity one carries traffic $L1$ in WS1; access entity two the forwarding traffic for WS1 in WS2; and access entity three transmits WS2 own generated traffic $L2$ in WS2 itself. Our EDCA scheduler is represented in the bottom part of Fig. 4.13.

In our scheduler, we assign the same TXOP for all access entities and one TXOP only accommodates one packet frame. Here, because WS1 and WS2 can pick different values for their EDCA parameters even for the same AC. Hence, disregarding the AC of the three access entities, we denote the EDCA parameters for them to be $AIFSN_1$ and CW_1, $AIFSN_2$ and CW_2, $AIFSN_3$ and CW_3[4], for access entity one, two and three respectively.

[4]We let CW=CWmin and $CWmax = (CWmin - 1)^5 - 1$

Inputs

The input controllable variables of three wireless stations are $X = \{C_1, C_2, C_3\}$, where C_i is a vector representing the controllable protocol parameter set for wireless station i. WS3 has no input in this example.

$$\begin{aligned}
X &= \{C_1,\ C_2,\ C_3\} \\
C_1 &= [L_1\ \ AIFSN_1\ \ CW_1] \\
C_2 &= [L_2\ \ UpThrshd\ \ LwThrshd\ \ award\ \ punish\ \ AIFSN_2\ \ CW_2, AIFSN_3\ \ CW_3] \\
C_3 &= [\ ]
\end{aligned}$$

And these input controllable variables come from three layers:

- *Application layer*: traffic load L_1 and L_2;
- *Network layer*: forwarding scheduler parameters $UpThrshd$, $LwThrshd$, $award$ and $punish$;
- *MAC layer*: EDCA parameters $AIFSN_1$, $AIFSN_2$,$AIFSN_3$, CW_1, CW_2, and CW_3.

Besides the controllable input factors, the rest of the 802.11b 2Mbps DSSS MAC/PHY parameters are fixed and shown in Table 2.1.

Outputs

The output responses that we are interested in are the throughput performance values S_1 and S_2.

$$Y = \{P_1, P_2, P_3\} = \{[S_1], [S_2], [\,]\} = [S_1\ \ S_2]$$

4.5.3 Design of Experiments

The experiment design is to decide the simulation configurations before the runs in order to obtain the desired information. Our experiment design uses a 2^k factorial design approach, and the configuration of each input factor is shown in Table 4.10.

Table 4.10: Experiment Design Parameters for Case III

Factors	**Levels of Variation**	**Level Values**
CW_1, CW_2, CW_3	2	8,32
$AIFS_1, AIFS_2, AIFS_3$	2	0,2
punish,award	2	0,1
LoThrshd	2	0,20
UpThrshd	2	80,100
$L_1 = L_2$=Load	2	0.2,0.5

Out simulation model is built in Arena [98], and can be divided into the following main parts: traffic generator, network layer forwarding scheduler and EDCA access scheduler. Although we are not able to validate the simulation model with respect to the real system, we can achieve partial verification since the EDCA access scheduler is already verified in [83, 42] with respect to an analytical model.

Our simulation model corresponds to a terminating simulation. We run each simulation replication for 2 hours and run 10 replications for each of the input combination.

4.5.4 Choose and Fit the Metamodel

The type of metamodel can be response surface, neural networks, induction learning and Kringing, etc. Here we choose to use response surface model due to its reasonable number of factors and the well-established theory and techniques of response surface methodology [7].

The most widely used response surface approximating functions are low-order polynomials. We pick a first-order polynomial function with interactions because we not only want to study the main effect of each factor but also their interactions.

$$S_1 = \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \sum_{i=1}^{k} \sum_{j=1, i<j}^{k} \beta_{ij} X_i X_j. \tag{4.11}$$

Here, k is the number of the factors, which is 11 in this experiment and $X_1 = Load$, $X_2 = award$, $X_3 = punish$, $X_4 = LoThrshd$, $X_5 = UpThrshd$, $X_6 = CW_1$, $X_7 = CW_2$, $X_8 = CW_3$, $X_9 = AIFSN_1$, $X_{10} = AIFSN_2$, and $X_{11} = AIFSN_3$.

We use least square regression analysis[5] over the simulation data in order to determine the coefficients of the polynomials. We run SAS GLM [84] program over the data we collect from simulation, and obtain the ANOVA statistics for the model. The high value (0.926761) of R^2, a goodness of fit index, indicates that the model exhibits a very high degree of explanatory power in characterizing the throughput performance.

We call this model the full-model, since it includes all input factors. But not all interactions in this model are significant, in other words, can be omitted from the model. We judge that the factors with t-test values larger than 0.05 are statistically insignificant and then delete them from the model. The new model without nonsignificant interactions is called reduced-model.

We re-fit the regression model for this reduced-model by SAS GLM. The R^2 of the reduced-model is still high enough (0.925798), which means it can still explain the data well. Also, the t-test values for each polynomial terms in the new model is statistically significant.

Table 4.11: ANOVA Table for Reduced Model of S_1 of Case III

Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr>F Pr>F**
Model	33	24.52219053	0.74309668	789.82	<.0001
Error	2014	1.89485694	0.00094084		
Corrected Total	2047	26.41704748			

R-Square	**Coeff Var**	**Root MSE**	**S1 Mean**
0.925798	27.03109	0.030673	0.113474

Therefore, after inserting these fitted coefficients of the reduced model βs into the

[5] supported by GLM in SAS

equation 4.11 [6] and transforming the absolute value back to the unit of $Mbps$, our final metamodel of throughput S_1 is:

$$
\begin{aligned}
S_1 \quad = \quad & -.4670 - .0024 * load + 0.1946 * award \\
& +0.5214 * punish - .0014 * LoThrshd \\
& +0.0062 * UpThrshd - .0017 * CW_1 - .0006 * CW_2 \\
& +0.0010 * CW_3 + -.0043 * AIFS_1 \\
& -.0027 * AIFS_2 + 0.0053 * AIFS_3 \\
& -.1470 * load * award + 0.2421 * load * punish \\
& +0.0045 * load * CW_1 + 0.0022 * load * CW_2 \\
& -.0055 * load * CW_3 + 0.0181 * load * AIFS_1 \\
& +0.0126 * load * AIFS_2 - .0278 * load * AIFS_3 \\
& +0.0103 * award * punish + 0.0017 * award * LoThrshd \\
& -.0025 * award * UpThrshd + 0.0003 * award * CW_2 \\
& -.0002 * award * CW_3 - .0047 * punish * UpThrshd \\
& +0.0007 * punish * CW_1 + 0.0003 * punish * CW_2 \\
& -.0008 * punish * CW_3 - .0047 * punish * AIFS_3 \\
& -.000008 * CW_1 * CW_2 + 0.00002 * CW_1 * CW_3 \\
& +0.0001 * CW_1 * AIFS_3
\end{aligned}
$$

4.5.5 Evaluate the Metamodel: Cross-layer effects

From the existence of interactions consisting of two factors from different layers (Table 4.12), we come to the conclusion that these two factors are cross-layer correlated. For example, $award$ from the network layer forwarding scheduler has different effects on S_1 for different value of $CW2$ from MAC layer. When keeping all other factors unchanged, a unit increase of $award$ will increase S_1 by $(0.1946 + 0.0103 punish +$

[6] The coefficients for the nonsignificant terms are zero.

$0.0017LoThrshd - 0.0025UpThrshd + \mathbf{0.0003*8} - 0.0002CW3$) for $CW2 = 8$ and by $(0.1946 + 0.0103punish + 0.0017LoThrshd - 0.0025UpThrshd + \mathbf{0.0003*32} - 0.0002CW3)$ for $CW2 = 32$.

Table 4.12: Cross-Layer Interactions

Application & Network Layer Interactions
$load * award$, $load * punish$
Application & MAC Layer Interactions
$load * CW_1$, $load * CW_2$, $load * CW_3$, $load * AIFS_1$, $load * AIFS_2$, $load * AIFS_3$
Network & MAC Layer Interactions
$award * CW_2$, $award * CW_3$, $punish * CW_1$ $punish * CW_2$, $punish * CW_3$, $punish * AIFS_3$

Therefore, we are able to quantify the cross-layer effects using a metamodeling technique. Taking into account all interactions, a cross-layer optimization is necessary in order to achieve the optimal throughput of $S1$, and our metamodel can serve well towards this goal. Although we only discuss the metamodeling of S_1 in this chapter, the same procedure can be applied to S_2 and the total throughput, and the same conclusion about cross-layer effects will apply.

4.6 Conclusions

Performance modeling of Wi-Fi networks is becoming increasingly important but also more challenging, with Wi-Fi networks becoming ubiquitous and carrying a large number of emerging applications.

The core of our work consists of advocating the use of metamodeling for addressing these challenges, and making a key methodological contribution: that is, to first build a framework of metamodeling network performance evaluation for Wi-Fi networks. Under this framework, many problems can be formulated and studied systematically; and very useful insights can be achieved for the better understanding and design of Wi-Fi networks through evaluation of the fitted metamodels. For example, we show three relevant cases of Wi-Fi network performance studies in this chapter. In

Case I, we are able to find an interesting log-linear relationship between the DCF capacity and the number of users and get an interesting message that the change of CWmin will not affect the throughput significantly in the range specified in our experiment; in Case II, we explain and illustrate the invalidity of the well-adopted VoIP capacity analysis, and provide a much tighter bound based on our metamodel leading to better CAC; and in Case III, the cross-layer interactions among MAC and network layer schedulers are first characterized and quantified, therefore can serve as the basis for cross-layer optimization and control.

The examples we illustrated here use only polynomial response surface models, however, some other problems may require other types of metamodels, such as neural networks or Kringing models. We also want to note that there are many issues about metamodels not mentioned in the scope of this chapter but require attention, such as how to choose an appropriate functional form, how to assess systematically the adequacy of the fitted metamodel, the range of application scenarios and the robustness of the prediction. Still, our work just exposes the tip of the proverbial iceberg, and hopefully there will be more efforts to make this multidisciplinary framework more mature and, most importantly, to highly boost the understanding and design of Wi-Fi networks.

Chapter 5

Summary

Wi-Fi access is becoming ubiquitous at homes, in public areas and enterprises to free people from the cumbersome wires, so far mainly for data traffic applications. How to provide QoS to emerging multimedia applications remains a challenging problem. The question itself is very broad. Good solutions cross over many areas including call admission control, fast handover and security, and such solutions also need to consider the impact of several layers, ranging from the application layer down to the MAC and PHY layer.

In the scope of this dissertation, we address the QoS provisioning problem mainly from the perspective of capacity estimation, which is a key component for efficient resource allocation schemes. We focus on the MAC layer through the legacy DCF or, better, through the newly approved EDCA, which constitute the core of Wi-Fi technology. Our contributions, in six parts, can be categorized into two broad areas (Fig.5.1)[1]:

- Capacity Modeling
  - DCF capacity metamodeling with saturation traffic.
  - VoWiFi capacity metamodeling with DCF MAC;
  - EDCA capacity (plus delay) analytical modeling with saturation traffic;

[1]Analytical models are in green color and metamodels are in yellow color. The control schemes are connected to the models on which they are based with brown dash lines.

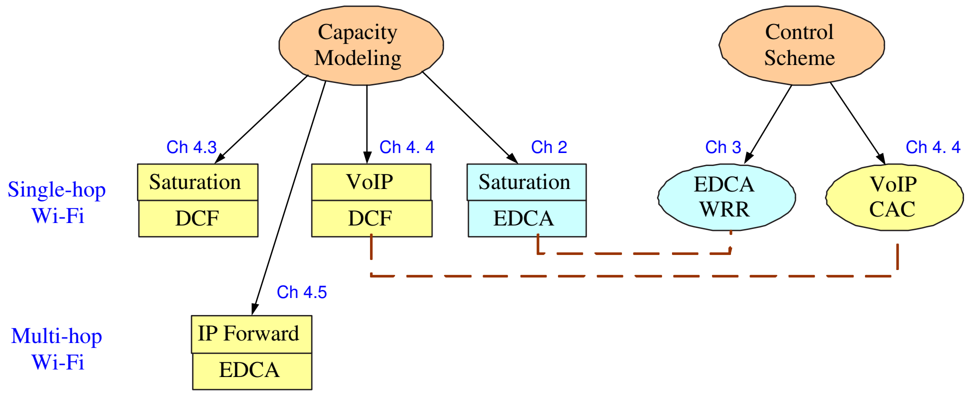


Figure 5.1: Contributions of this dissertation.

  - Throughput metamodeling with network layer forwarding scheduler and EDCA in multi-hop WLAN.

- Control Scheme

  - WRR scheduler at MAC layer through EDCA parameters manipulating based on EDCA analytical model;

  - CAC for VoIP application with QoS considerations based on VoWiFi capacity metamodel.

The analytical model of EDCA capacity estimation under saturation traffic unified three widely used models and outperforms the other models due to its rigorousness and ease of application. Based on this model, we design a MAC layer WRR scheduler that allocates radio channel resources, achieving higher utilization than a network layer WRR. Due to the limitations and difficulties of analytical modeling in most realistic circumstances, we first advocate the use of metamodeling for the study of Wi-Fi network performance and we build a metamodeling framework under which many problems can be formulated and solved systematically.

The application of this framework to three important case studies provides intriguing results. First, a log-linear relationship between DCF capacity and the number of users is exposed for the first time; second, VoWiFi admission capacity is derived with much tighter accuracy by our metamodel than current analysis bounds, and reasons behind this are also considered for the first time; third, the cross-layer factor interactions among network layer schedulers and MAC layer EDCA is characterized for the first time through our metamodel of multi-hop ad hoc networks.

There is clearly more work needed to be done in order to guarantee QoS for end users with multimedia applications in Wi-Fi networks. Mixed analytical modeling plus metamodeling will be a practical way to solve more complicated performance problems. And more powerful CAC with MAC and PHY adaptations based on performance models deserve great effort in the future.

Bibliography

[1] T. Nandagopal, S. Lu, and V. Bharghavan, "A unified architecture for the design and evaluation of wireless fair queueing algorithms," in *Proc. ACM MOBICOM 1999*, Seattle, USA, Aug. 1999, pp. 132–142.

[2] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet network," *IEEE/ACM Trans. Networking.*

[3] T. S. E. Ng, I. Stoica, and H. Zhang, "Packet fair queueing algorithms for wireless networks with location-dependent errors," in *Proc. INFOCOM (3)*, 1998, pp. 1103–1111. [Online]. Available: citeseer.nj.nec.com/article/ng98packet.html

[4] S. Lu, T. Nandagopal, and V. Bharghavan, "Fair scheduling in wireless packet networks," in *Proc. ACM MOBICOM 1998*, Oct. 1998.

[5] "IEEE 802.11 part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications," Aug. 1999, IEEE WG.

[6] IEEE 802.11 WG, "Draft Supplement to IEEE Standard 802.11-1999: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)," Nov. 2003, IEEE 802.11e D6.0.

[7] R. H. Myers and D. C. Montgomery, *Response Surface Methodology*, 1995.

[8] T. W. Simpson, J. D. Peplinski, P. N. Koch, and J. K. Allen, "Metamodels for Computer-based Engineering Design: Survery and recommnendations," *Engineering with Computers*, pp. 129–130, 2001.

[9] G. Bianchi, "IEEE 802.11 Saturation Throughput Analysis," *IEEE Commun. Lett.*, vol. 2, pp. 318–320, Dec 1998.

[10] ——, "Performance analysis of the IEEE 802.11distributed coordination function," *IEEE J. Select. Areas Commun.*, vol. 18, no. 3, pp. 535–547, 2000.

[11] F. Calì, M. Conti, and E. Gregori, "IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement," in *Proc. INFOCOM*, 1998.

[12] ——, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Trans. Networking*, vol. 18, no. 6, pp. 785–799, Dec 2000.

[13] Y. C. Tay and K. C. Chua, "A capacity analysis for the IEEE 802.11 MAC protocol," *Wireless Networks*, pp. 159–171, July 2001.

[14] 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 *Proc. INFOCOM 2002*, Jun 2002.

[15] A. Imad and C. Claude, "Differentiation mechanisms for IEEE 802.11," in *Proc. IEEE INFOCOM 2001*, Anchorage, Alaska, April 2001.

[16] A. Lindgren, A. Almquist, and O. Schelén, "Evaluation of Quality of Service schemes for IEEE 802.11 wireless LANs," in *Proc. of the 26th Annual IEEE Conference on Local Computer Networks*, Nov 2001.

[17] ——, "Quality of service schemes for ieee 802.11 wireless lans: an evaulation," *ACM Mobile Networks and Applications (MONET)*, pp. 223–235, June 2003.

[18] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service (invited paper)," in *Proc. the European Wireless*, vol. 1, Florence, Italy, February 2002, pp. 32–39.

[19] S. Choi, J. D. Prado, Sai Shankar N, and S. Mangold, "IEEE 802.11e Contention-Based Channel Access (EDCF) Performance Evaluation," in *Proc. ICC 2003*, May 2003.

[20] H. L. Truong and G. Vannuccini, "The IEEE 802.11e MAC for Quality of Service in Wireless LANs," in *Proc. SSGRR 2003*, Fan 2003.

[21] D. He and C. Q. Shen, "Simulation Study of IEEE 802.11e EDCF," in *Proc. the 57th IEEE Semiannual VTC*, Apr 2003.

[22] A. Grilo and M. Nunes, "Performance Evaluation of IEEE 802.11e," in *Proc. PIMRC'02*, Sep 2002.

[23] S. Mangold, "Analysis of ieee 802.11e and application of game models for support of quality-of-service in coexisting wireless networks," Ph.D. dissertation, June 2003.

[24] D. Qiao and K. G. Shin, "Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802.11 WLAN under the DCF," in *Proc. IWQoS'2002*, May 2002.

[25] J. Zhao, Z. Guo, Q. Zhang, and W. Zhu, "Performance Study of MAC for Service Differentiation in IEEE 802.11," in *Proc. Globecom 2002*, Nov 2002.

[26] G. Bianchi and I. Tinnirello, "Analysis of Priority Mechanisms Based on Differentiated Inter Frame Spacing in CSMA-CA," in *Proc. of the VTC 2003 Fall*, 2003.

[27] Y. Ge and J. Hou, "An Analytical Model for Service Differentiation in IEEE 802.11," in *Proc. ICC 2003*, May 2003.

[28] Y. Xiao, "Backoff-based Priority Schemes for IEEE 802.11," in *Proc. ICC 2003*, May 2003.

[29] ——, "Enhanced DCF of IEEE 802.11e to Support QoS," in *Proc. WCNC 2003*, 2003, pp. 1291–1296.

[30] H. Zhu and I. Chlamtac, "An Analytical Model for IEEE 802.11e EDCF Differential Services," in *Proc. ICCCN 2003*, Oct 2003.

[31] J. He, L. Zheng, Z. Yang, and C. T. Chou, "Performance Analysis and Service Differentiation in IEEE 802.11 WLAN," in *Proc. of the 28th International Conference on Local Computer Networks*, Oct 2003.

[32] J. Hui and M. Devetsikiotis, "Designing Improved MAC Packet Scheduler for 802.11e WLAN," in *Proc. Globecom 2003*, Dec 2003.

[33] K. Xu, Q. Wang, and H. Hassanein, "Performance Analysis of Differentiated QoS Supported by IEEE 802.11e Enhanced Distributed Coordination Function (EDCF) in WLAN," in *Proc. Globecom 2003*, Dec 2003.

[34] R. Battiti and B. Li, "Supporting service differentiation with enhancements of the ieee 802.11 mac protocol: Models and analysis," University of Trento, Italy, Tech. Rep., May 2003.

[35] B. J. Kwak, N. O. Song, and L. E. Miller, "Analysis of the Stability and Performance of Exponential Backoff," in *Proc. WCNC 2003*, 2003.

[36] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed multi-hop scheduling and medium access with delay and throughput constraints," in *Proc. MOBICOM 2001*, Jul 2001.

[37] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless LAN," in *Proc. MOBICOM 2000*, Aug. 2000, pp. 167–178.

[38] H. Luo, S. Lu, and V. Bharghavan, "A new model for packet scheduling in multihop wireless networks," in *Proc. MOBICOM 2000*, Aug. 2000.

[39] A. Banchs and X. Perez, "Providing throughput guarantees in IEEE 802.11 wireless lan," in *Proc. WCNC*, vol. 1, 2002.

[40] L. Wischhof and J. Lockwood, "A resource-based approach to MAC layer independent hierarchical link-sharing in wireless local area networks," in *Proc. European Wireless 2002*, Florence, Italy, Feb 2002.

[41] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single node case," *IEEE/ACM Trans. Networking*, Jun 1993.

[42] J. Hui and M. Devetsikiotis, "A Unified Model for the Performance Analysis of IEEE 802.11e EDCA," *IEEE Transaction on Communications*, Sept. 2005.

[43] F. Anjum, M. Elaoud, D. Famolari, A. Ghosh, R. Vaidyanathan, A. Dutta, and R. Agrawal, "Voice Performance in WLAN Networks- An Experimental Study," in *Proc. IEEE Globecom 2003*, 2003.

[44] K. Medepalli, P. Gopalakrishnan, D. Famolari, and T. Kodama, "Voice Capacity of IEEE 802.11b and 802.11a Wireless LANs in the Presence of Channel Erros and Different User Data Rates," in *Proc. of VTC Fall 2004*, 2004.

[45] ——, "Voice Capacity of IEEE 802.11b, 802.11a and 802.11g Wireless LANs," in *Proc. Globecom'04*, 2004.

[46] P. Clifford, K. Duffy, D. Leith, and D. Matone, "On Improving Voice Capacity in 802.11 Infrastructure Nteworks," in *Proc. of IEEE WirelessCom 2005*, 2005.

[47] D. Matone, K. Duffy, and D. Leith, "Modeling the 802.11 Distributed Coordination Function with Heterogenous Finite Load," in *Resource Allocation in Wireless Networks*, 2005.

[48] W. Wang, S. C. Liew, and V. O. K. Li, "Solutions to Performance Problems in VoIP Over a 802.11 Wireless LAN," *IEEE Transactions on Vehicular Technology*, 2005.

[49] D. Pong and T. Moors, "Call Admission Control for IEEE 802.11 Contention Access Mechanism," in *Proc. Globecom'03*, 2003.

[50] S. Garg and M. Kappes, "Admission Control for VoIP Traffic in IEEE 802.11 Networks," in *Proc. Globecom'03*, 2003.

[51] ——, "Can I add a VoIP Call?" in *Proc. ICC'03*, 2003.

[52] D. P. Hole and F. A. Tobagi, "Capacity of an IEEE 802.11b Wireless LAN supporting VoIP," in *to appear in ICC'04*, 2004.

[53] M. Coupechoux, V. Kumar, and L. Brignol, "Voice over IEEE 802.11b Capacity," in *ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*, 2004.

[54] M. Ergen and P. Varaiya, "Throughput Analysis and Admission Control for IEEE 802.11a," *To appear in ACM-kluwer MONET Special Issue on WLAN optimization at the MAC and Network Levels*, 2004.

[55] L. Buttyan and J. P. Hubaux, "Stimulating cooperation in self-organizing mobile ad noc networks," *ACM/Kluwver Mobile Networks and Applications (MONET)*, October 2003.

[56] S. Zheng, J. Chen, and Y. R. Yang, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-hoc Networks," in *Proc. of Infocom*, March 2003.

[57] B. Raghavan and A. C. Sneoeren, "Priority Forwarding in Ad Hoc Networks with Self-Interested Parties," in *Proc. of Workshop on Economics of Peer-to-Peer Systems*, 1991.

[58] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Proc. of Mobihoc*, June 2000.

[59] S. Buchegger and J. Y. L. Bouder, "Performance Analysis of CONFIDENT Protocol Cooperation of Nodes - Fairness in Dynamic Ad-hoc Netowrks," in *Proc. of Mobihoc*, June 2002.

[60] M. Conti, E. Gregori, and G. Maselli, "Towards Reliable Forwarding in Ad Hoc Networks," in *Proc. of PWC 2003*, September 2003.

[61] W. D. Kelton, "Statistical Analysis of Simulation Output," in *Proc. Winter Simulation Conference*, 1997.

[62] R. R. Barton, "Metamodels for Simulation Input-Output Relations," in *Proc. Winter Simulation Conference*, 1991.

[63] T. G. Bailey, K. Bauer, and A. B. Marsh, "Response Surface Analysis Stochastic Network Performance," in *Proc. Winter Simulation Conference*, 1989.

[64] S. C. Agrawal, *Metamodelig A Study of Approximations in Queueing Models.* The MIT Press, 1985.

[65] W. H. Shaw, "The Application of Metamodeling to Interconnection Network Analysis," *ORSA Journal on Computing*, pp. 369–380, 1994.

[66] C. Barrett, M. Drozda, A. Marathe, and M. V. Marathe, "Characterizing the Interaction Between Routng and MAC Protocols in Ad-hoc Networks," in *Proc. of Mobihoc 2002*, 2002.

[67] ——, "Analyzing Interaction Between Network Protocols, Topology and Traffic in Wireless Radio Networks," in *Proc. of WCNC 2003*, 2003.

[68] K. K. Vadde and V. R. Syrotiuk, "Factor Interaction on Service Delivery in Mobile Ad Hoc Networks," *IEEE Journal on Selected Area in Communications*, pp. 1335–1346, 2004.

[69] J. P. C. Kleijnen, *Statistical Tools for Simulation Practitioners*, 1987.

[70] R. R. Barton, "Metamodeling: A Stated of the Art Review," in *Proc. Winter Simulation Conference*, 1994.

[71] ——, "Simulation Metamodels," in *Proc. Winter Simulation Conference*, 1998.

[72] W. D. Kelton, "A Tutorial on Design and Analysis of Simulation Experiements," in *Proc. Winter Simulation Conference*, 1995.

[73] ——, "Statistical Issues in Simulation," in *Proc. Winter Simulation Conference*, 1996.

[74] R. G. Sargent, "Research issues in metamodeling," in *Proc. Winter Simulation Conference*, 1991.

[75] G. E. P. Box and K. B. Wilson, "On the experimental attainment of optimum conditions," *Journal of Royal Statistical Society*, pp. 1–38, 1951.

[76] G. E. P. Box and N. R. Draper, *Empirical Model Building and Response Surfaces*, 1987.

[77] A. M. Law and W. D. Kelton, *Simualtion Mdodeling and Analysis*, 2000.

[78] J. P. C. Kleijnen, *Experimental design for sensitivity analysis, optimization, and validation of simulation models*, 1998.

[79] R. H. Myers, A. I. Khuri, and W. H. Charter, "Response Surface Methodology: 1966-1988," *Technometrics*, pp. 137–157, 1989.

[80] R. H. Myers, *Journal of Quality Tecnology, Title = "Response Surface Methodology: current status and future directions", pages = "30-74", Year = 1999.*

[81] E. Angün, D. D. Hertog, J. P. C. Kleijnen, and G. Gürkan, "Response Surface Methodology Revisited," in *Proc. Winter Simulation Conference*, 2002.

[82] R. Jain, *The Art of Computer Systems Performance Analysis*, 1991.

[83] J. Hui and M. Devetsikiotis, "Performance Analysis of IEEE 802.11e EDCA by a Unified Model," in *Proc. Globecom 2004*, Dec 2004.

[84] SAS Website, http://www.sas.com.

[85] F. Ohrtman, *Voice Over 802.11.* Aretech House, 2004.

[86] P. Kyasanur and N. Vaidya, "Detection and Handling of MAC Layer Misbehavior in Wireless Networks," in *Proc. of Dependable Computing and Communications*

Sysmposium (DCC) at the International Conferecne on Dependable Systems and Networks (DSN), June 2003.

[87] M. Cagalj, S. Ganeriwal, and I. Aad, "On Cheating in CSMA/CA Ad Hoc Networks," Tech. Rep., 2004.

[88] M. Conti, E. Gregori, and G. Maselli, "Cooperation Issues in Mobile Ad Hoc Networks," in *Proc. of 24th International Conference on Distributed Computing Systems Workshops - W6: WWAN (ICDCSW'04)*, March 2004.

[89] M. Conti, G. Maselli, G. Turi, and S. Giordano, "Cross-layering in mobile ad hoc network design," *IEEE Computer special issue on ad hoc networks*, Feburary 2004.

[90] Y. Kwon and K. Lat, "Improving transport layer performance by using a novel medium access control protocol with fast collision resolution in wireless lans," in *Proc. of International Workshop on Moddling Analysis and Simulation of Wireless and Mobile Systems*, 2002.

[91] S. Toumpis and A. J. Goldsmith, "Capacity Regions for Wireless Ad Hoc Networks," *IEEE Transactions on Wireless Communications*, no. 4, pp. 746–748, April 2003.

[92] R. L. Cruz and A. V. Santhanam, "Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Networks," in *Proc. of Infocom*, March 2003.

[93] S. Pollin, B. Bougard, and G. Lenoir, "Cross-Layer Exploration of Link Adaptation in Wireless LANs with TCP Traffic," in *Proc. of Symposium IEEE Benelux Chapter on Communications and Vehicular Technology*, 2003.

[94] M. Chiang, "To Layer or Not to Layer: Balancing Transport and Physical Layers in Wireless Multihop Networks," in *Proc. of Infocom*, March 2004.

[95] L. Chen, S. H. Low, and J. C. Doyle, "Joint Congestion Control and Media Access Control Design for Ad Hoc Wireless Networks," in *Proc. of Infocom*, March 2005.

[96] X. Lin and N. B. Shroff, "The Impact of Imperfect Scheduling on Cross-Layer Rate Control in Wireless Networks," in *Proc. of Infocom*, March 2005.

[97] Jie Hui and Michael Devetsikiotis, "Metamodeling of Cross-layer Cooperative Scheduling in Wireless Local Area Networks," Center for Advanced Computing and Communications, technical report TR 05/02, 2005.

[98] Arena Website, http://www.arenasimulation.com.