Quality of service based distributed control of wireless networks

DOI:
10.6100/IR774635

Document status and date:
Published: 26/06/2014

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Quality of Service Based Distributed Control of Wireless Networks

Samenstelling promotiecommissie:

voorzitter:	prof.dr. E.H.L. Aarts
promoter:	prof.dr. J.J. Lukkien
copromoter:	dr. T. Özçelebi
leden:	prof.dr.ir. I.G.M.M. Niemegeers (Delft University of Technology)
	dr. P. Corcoran (National University of Ireland)
	prof.dr. A. Liotta (Technische Universiteit Eindhoven)
	prof.dr.ir. C.H. van Berkel (Ericsson, Technische Universiteit Eindhoven)

The work leading to this thesis has been financially supported by the Higher Education Commission of Pakistan (primarily), OMECA (Optimization of Modular Embedded Computer-vision Architectures) and COMET (Converged Messaging Technology - FP6) projects.

A catalogue record is available from the Technische Universiteit Eindhoven Library.

Proefschrift - ISBN 978-90-386-3643-6
NUR 980

Typeset: Waqar Aslam using LaTeX
Cover design: Waqar Aslam
Reproduction: Ipskamp Drukkers, Enschede, The Netherlands

Quality of Service Based Distributed Control of Wireless Networks

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op dinsdag 26 juni 2014 om 16.00 uur

door

Waqar Aslam

geboren te Bahawalpur, Pakistan

Dit proefschrift is goedgekeurd door de promotor:

prof.dr. J.J. Lukkien

copromotor:

dr. T. Özçelebi

Summary

Quality of Service Based Distributed Control of Wireless Networks

Wireless networks have seen major research initiatives on service pervasiveness and inspired a multitude of applications. The application areas include consumer electronics, industrial automation, medical appliances and vehicular communications. The applications have different quality of service (QoS) requirements such as minimum throughput and maximum latency. The QoS requirements can be fulfilled by a channel allocation mechanism, without which the available bandwidth is shared evenly. To this end, we set the QoS challenge composed of three parts that are addressed in this thesis. The first challenge concerns the specification of QoS requirements. The second challenge concerns achieving a predictable distributed control of IEEE 802.11e wireless network for QoS-aware Medium Access Control (MAC) based reservations. The third challenge concerns mapping the QoS requirements to network configurations that can offer QoS fulfillment. These challenges are addressed by offering a performance framework that is the main contribution of this thesis work. We evaluate this framework on realism, generality and predictability.

For QoS specification, we propose a tuple of QoS concerns (frequency, share, latency and reliability) to accommodate all major application types. In order to compare QoS properties of different network configurations, we define an order relation on QoS properties. In addition, priority semantics orders applications by their criticality.

For achieving a distributed control of a wireless network, we identify two fundamental system concerns that influence MAC based reservations, namely *gaining control* of the channel that reflects on the opportunity to transmit, and *retaining control* of the channel that reflects on the duration of keeping the opportunity. Both concerns are quantified. The quantification is based on developing a thorough understanding of the MAC behavior through studying MAC traces. For this purpose, a trace framework is developed that describes MAC evolution over time. Next, we develop analysis leading to functions, mapping network configurations to MAC steady state behavior, representing time reduced trace properties. This behavior is underpinned by relevant probability distributions, which leverage development of functions for MAC process QoS properties. The derivation of probability distributions is based on fixed point theory. We thoroughly investigate the convergence behavior of our algorithm and report on outlier network configurations.

For mapping the QoS requirements to network configurations, the order relation on QoS properties is used. A side effect of this relation is multiplicity of solutions. This is dealt with by adopting a Pareto based approach, in which system properties are used to define multiobjective optimization spaces. While single solutions in these spaces are determined by global optimizations, each optimal solution represents a tradeoff point, thus offering choices. The proposed approach keeps separation of the decision process to resolve multiplicity, from the original mapping problem. Thus, such independence allows defining other relevant strategic objectives.

Addressing the QoS challenge has a system aspect, namely, the channel utilization, which may not be optimal for the determined network configuration. We study this system aspect by two approaches. In the first proactive approach, a predictive model of channel utilization is derived that uses a global policy of enriching the MAC decisions by the properties of applications. The gain in channel utilization is related to the application property. In the second reactive approach, MAC processes adapt their behavior with the aim to prevent channel wastage in pessimistic MAC reservation. The extent of wastage avoidance depends on the extent of network suboptimality as ascertained by the network configuration.

All the mathematical mappings derived in this thesis are validated by corresponding software tools that are developed locally, including our QoS based EDCA simulator.

Contents

1

Introduction

Living a connected life enables meaningful and compelling possibilities. It offers rich experiences by accessing services ‘anywhere, anytime and any-device’, which have societal, economic, cultural, entertaining and educational aspects. Steered by wireless pervasiveness, these services predominantly evolve over networks of different scales, which empower consumers with ubiquity, mobility, broad reachability, connectivity, flexibility, personalization and cost reduction with ease of setup. The convenience of ‘air links’ has dominated limitation on their bandwidth, resulting in an explosive growth of wireless devices over recent years (cf. Figure1.1).

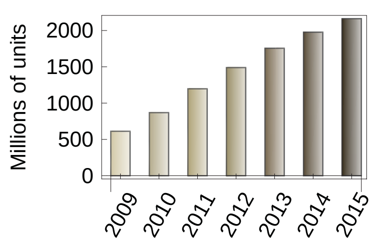


Figure 1.1: Global unit shipments and penetration of wireless local area network (WLAN) capability in suitable electronic products [66].

A wireless network that considers the quality of service (QoS) requirements of applications, but without a channel allocation mechanism would typically seek to share the channel capacity evenly among the users. Without distinction between specific wireless technologies, this simplistic division does not align with differing QoS requirements of applications such as throughput and latency. Consequently this approach emerges as a shortcoming due to which QoS requirements of applications are possibly unfulfilled — hence, they suffer, which is undesired. This competitive but uncontrolled situation over the channel can be avoided by managing its sharing.

The vision of Mark Weiser on ubiquitous computing [225] has been heralding major research initiatives on service pervasiveness, for which distributed wireless networks are key enablers that support scalability, fault tolerance and high availability. Therefore, QoS-aware channel allocation over a distributed wireless network emerges as a highly relevant research area and, forms the general topic accepted in this thesis. Before presenting the problem statement, the specific research questions and the corresponding contributions, the next section will elaborate on the topic.

1.1 Motivation

QoS as a term, is formally defined as, 'Totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service' [59]. It aims at guaranteeing the network capability to deliver predictable results. QoS relevance for networks emerges due to, competition over the channel and varying requirements of different traffic types, which ought to be fulfilled for user satisfaction. Thus, a need of QoS enforcement mechanisms is arguably supported by ever increasing demands of new applications, devices and users. However, capacities of networks and transmission performance are being enhanced continuously, so that over-provisioning allows implicit fulfillment of QoS requirements of applications — thus a tendency to oppose the need of QoS enforcement mechanisms [102]. Amidst this continual battle, internet is growing remarkably. Especially, real-time applications such as an interactive voice/video, streaming and other consumer services are sensitive to QoS, which ensures high-quality performance for them according to their demands. Such QoS-sensitive traffic is increasingly dominating the internet (cf. Figure 1.2).

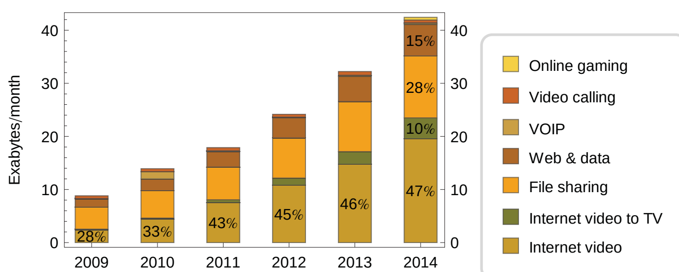


Figure 1.2: Global internet traffic forecast by application with 36% compound annual growth rate in 2009 – 2014 [99].

For network providers, QoS-aware channel allocations leverage efficient utilization of resources and drive productivity by enhancing provided service levels. Other QoS benefits include network manageability from a business perspective, improved user experience and controllability of QoS dimensions [198, 54]. It allows separation of concerns regarding Service Level Agreements between network providers and service providers.

With this background, we set the QoS challenge as reservation of channel bandwidth so as to deliver the desired performance of applications. Within our context, a channel is a source-sink logical connection. In general, the QoS challenge has been addressed at different levels of the Open Systems Interconnection (OSI) stack [98], though dominantly at the Network layer and the Data Link layer (DLL). Addressing the QoS challenge consists of provisioning of service prioritization in IP networks, which typically comprise backbone networks and access networks. Thus, assuring guarantees in the IP networks requires all of their constituent segments to comply with the demands. Such provisioning has an enormous commercial appeal, hence it has been under focus for more than a decade now.

For the backbone networks, various QoS enabled Internet architectures, such as *integrated services* (IntServ) [27] and *differentiated services* (DiffServ) [22], have been proposed to address the QoS challenge at the Network layer. IntServ gives end-to-end guarantees through admission control of flows. Prior to admitting a flow, each router that falls in the end-to-end path is negotiated with, regarding availability of sufficient resources and corresponding reservations. Due to processing overheads, IntServ tends to be complex and less scalable. An improvement to IntServ is proposed in DiffServ, which is simpler and driven by the scalability concern but lacks in both aspects for flows — the end-to-end scope and the guarantee assurance. In DiffServ, the main governing principle is to push complexity to the edges of the backbone networks, where flows are classified according to their priorities and treated accordingly. Due to the tradeoff between predictability of end-to-end service guarantees and scalability, various QoS solutions for the backbone networks have been proposed by combining both architectures [26, 123, 243, 8, 93, 75]. The combined approach resulted in two variants of DiffServ: absolute DiffServ [187, 191] and relative DiffServ [152, 64]. The prior is similar to IntServ without a need to store channel-state-information. The latter allocates resources proportionally such that the QoS ratios amongst the service classes are maintained during resource fluctuations. The work in [64] has proposed a Hybrid Proportional Delay (HPD) scheduling that is extended into delay differentiation queuing (DDQ) [69]. DDQ is based on QoS demands and the actual delay, the flows experience. More related work is reported in [226, 132, 71, 9, 234, 213].

For the last-hop access networks, the QoS challenge is addressed at DLL, in particular at its Medium Access Control (MAC) sublayer. Enforcing QoS at the MAC sublayer brings in distinctive benefits:

- Applications that trespass beyond their declared requirements, are said to misbehave. Such misbehaving applications can be controlled by a QoS-aware MAC within its functional scope, which is independent of specific applications. This offers the necessary QoS protection for suffering applications;
- MAC is closest to the functionally well established Physical layer (PHY), hence quickest in sensing events there precisely, and possibly adapting its behavior accordingly;
- Without a MAC QoS controller, end-to-end QoS cannot be ensured.

Of such last-hop segments, wireless access networks have recently started to predominate other technologies such as wired access networks that have remained the dominant form of network access. Amongst the wireless technologies, the cheap availability of WLANs renders them as a befitted solution for major pervasive environments such as intelligent transport systems, wide mesh networks, smart homes, smart cities, etc. They comply with the IEEE 802.11 standard [92], while their members synchronize into a basic service set (BSS) [92], to operate in either of two modes: *infrastructure* BSS and *independent* BSS (IBSS). In the prior mode, networks are controlled centrally, while in the latter mode, networks are not managed centrally, rather each member competes for the channel independently. These unmanaged networks are implicitly controlled through autonomous behaviors of their members. Network arbitration principle in all WLANs is based on *carrier sense multiple access with collision avoidance* (CSMA/CA). Operating in the latter mode results in random access networks, implying arbitration to the network is non-deterministic.

Over the evolution of WLANs, PHY channel capacity is being enhanced, featuring approvals of major technologies. In parallel, requirements on them have distinctively shifted from fairness to QoS. In QoS-devoid legacy WLANs, based on the IEEE 802.11 standard [91], the two operating modes are provided by Point Coordination Function (PCF) and Distributed Coordination Function (DCF), respectively. After several MAC layer amendments for supporting QoS, the legacy standard evolved into the current IEEE 802.11 standard [92], which has introduced IEEE 802.11e [92] that offers Hybrid Coordination Function (HCF) controlled channel access (HCCA) and Enhanced Distributed Channel Access (EDCA), for the two operating modes, respectively. The availability of the QoS support during WLAN evolution is summarized in Table 1.1 [238].

scheme	DCF	EDCA	PCF	HCCA
access method	contention	contention	hybrid	hybrid
access category	distributed	distributed	centralized	centralized
QoS support	none	prioritized	not explicit	guaranteed

Table 1.1: IEEE 802.11 MAC channel access methods comparison.

The hybrid access method refers to a super period that offers both, contention and contention free types of accesses. Similar to the QoS support offered by the DiffServ and the IntServ architectures for the backbone networks, we rank the extent of the available QoS support on performance assurance, in the current IEEE 802.11 standard [92] (cf. Table 1.2).

control	DCF	EDCA	PCF	HCCA
DiffServ	✗	✓ (partial)	✗	✓
IntServ	✗	✗	✗	✓
✓: available ✓ (grey): partially available ✗: not available				

Table 1.2: QoS support in WLANs.

EDCA provides a QoS differentiation service, though its extent cannot be predicted. Here, we point out the absence of a relationship linking the desired QoS ratios to the MAC parameters. Due to this observation, the QoS support by EDCA is considered partial for the DiffServ case. For assuring QoS guarantees in the IntServ case, a relation from the desired QoS requirements to the MAC parameters, is missing. We assume these two relations would enable relative and absolute QoS provisioning in the last-hop segments, thus realizing a full coverage of the end-to-end path. Insufficient QoS support by EDCA on these two identified aspects, is the core stimulant of research, as addressed in this thesis. Moreover, selection of an IBSS, in which the member devices form an ad-hoc network using QoS-aware EDCA mechanism, is further motivated by unavailability of a central controller. Thus, we set scope of our QoS challenge to such wireless ad-hoc networks.

1.2 Problem statement and research questions

The problem statement that is the subject of this thesis, is formulated as:

- How to specify the communication related QoS requirements of applications?

- In order to fulfill the communication related QoS requirements of applications, network resources have to be reserved through relevant enforcement mechanisms. Such reservations call for asserting a control over the network. Hence, the problem to address is — how to control the network behavior in a distributed manner?

- How to map the communication related QoS requirements of applications to some desired network behavior such that it can fulfill those requirements?

The challenges raised in the problem statement are characterized by a few research questions, which are addressed in relevant chapters and formulated as follows.

Question 1. What is the conceptual model of the desired QoS based MAC reservations on the channel?

This question takes up the first challenge in the problem statement. The QoS requirements of applications can be met by corresponding MAC based reservations on the channel. Enforcement of reservations requires a MAC based distributed control over the channel. This calls for understanding the MAC functionality in detail. Thus, this question concerns sketching a conceptual model of our system. It comprises two subquestions:

a. How to abstract the MAC behavior towards a performance model?

b. How to specify the communication related QoS requirements of applications?

Studying subquestion a. highlights complex execution patterns of random access network protocols, which connote unpredictable behaviors [17, 247, 181]. Under such uncertainty, scheduling nodes for transmission is not trivial, especially in a distributed network. Adding to this complexity is the variable size of data packets. To this end, this subquestion aims for dissecting the complex MAC functionality that may lead to appropriate abstraction. Dissection allows understanding the MAC behavior in detail.

Studying subquestion b. allows to introduce the QoS context of our system. To this end, we aim for a proposal that admits service level agreements (SLAs). SLAs offer ground for common understanding between a service provider and a customer. They are based on performance metrics, reflecting the QoS requirements that need to be met. A formal specification of the QoS requirements reduces management complexity and allows service auditing, which is the assessment of adequacy of a service.

The second challenge in the problem statement focuses on controlling the network behavior in a distributed manner. The aim of such a control is to leverage desired QoS based MAC reservations on the channel. The network behavior emerges as a result of the independent executions of several concurrent MAC processes accessing the channel. The execution history, recorded in network traces, contains rich information about the actual execution. To this end, we want to understand the combined behavior of MAC processes by examining the network traces. The related question is:

Question 2. How to track the EDCA network evolution over time?

This question allows to study the network properties over time. Within its scope, four subquestions are identified:

a. How to discover MAC states resulting from arbitrary subtraces of processes?
   Given subtraces of processes collected independently, this subquestion studies their combined effect at the network level.

b. How to derive low-level performance metrics such as occurrence probabilities of process states, from an arbitrary subtrace of a process?
   This problem is motivated by the observation that mostly the high-level performance metrics, such as the QoS properties, tend to abstract internal behaviors of a process. The metrics sought can also be used to derive high-level performance metrics.

c. How to identify and specify process-specific and MAC event-specific cyclic behaviors in traces?
   Studying the properties of recurrence patterns provides insights into QoS fluctuations.

d. How to examine QoS properties for both, short-term and long-term durations, from an arbitrary subtrace of a process?

Given a trace, the evolution of network QoS properties over time reflects on details that must be considered for a mathematical analysis. Specifically, it implies the recurring cyclic behavior over time that needs to be transformed into a predictive model of stochastic network cycles. For this purpose, we aim for an analysis leading to a function, mapping network configurations to occurrence probabilities of process-specific and MAC event-specific cycles. To keep generality, it is mandatory to consider all influential MAC arbitration mechanisms over a given QoS concern. Such generalization enables to study the extent of their joint influence. The stochastic network cycles exist in a space that represents an average network behavior. A point in this space is termed as a network steady state. Thus, a relevant question is:

Question 3. How to develop a quantitative model for network steady state?

The network steady state is based on steady states of individual MAC processes. To address this question, we aim for an analysis that is led into a single predictive model.

The stochastic network cycles reflect on average MAC effects. Thus, quantitative analysis based on these cycles, can reflect effectively on average MAC performance regarding QoS concerns. Leveraging on these cycles, the relevant question is:

Question 4. How to develop quantitative models for QoS properties of MAC processes?

This question aims to study the QoS properties of stochastic network cycles. By achieving a distributed control of QoS properties over the EDCA network, we are left with the third challenge in the problem statement. It concerns finding some network configuration that can fulfill given QoS requirements of applications. The relevant question is:

Question 5. How to map the communication related QoS requirements of applications to some desired network behavior that fulfills those requirements?

Network behavior, reflected by network steady state, is determined by a corresponding network configuration. Each network behavior offers a set of MAC reservations, one per process. This question seeks a function, mapping the QoS requirements of applications to network behaviors in accordance with the QoS fulfilment relation as defined in the QoS semantics.

The three challenges in the problem statement seek demand-based MAC reservations. Each set of MAC reservations has a channel utilization aspect, which can be regarded as an extra-functional quality of our system. Given that wireless channel is a scarce resource, its maximal utilization is desired. Following two research questions study this system aspect.

Question 6. How to improve the channel utilization for the wireless network under control of a distributed MAC, by exploiting characteristics of applications?

This question studies the impact of enriching MAC decisions with knowledge of applications.

Question 7. How to adapt distributed control of MAC autonomously, to avoid channel wastage due to pessimistic reservations?

This question highlights suboptimal utilization of the channel capacity over time that we seek to improve upon.

1.3 Objectives and contributions

The objective of the research undertaken in this thesis is to address the QoS challenge for wireless ad-hoc networks. Primarily, we steer our research into development of quantitative models that address the challenges in the problem statement, using theory of stochastics and optimization. We aim at analysis developed with sound mathematical reasoning.

The core contribution of this research is the formulation and validation of a performance framework that is able to resolve challenges in the problem statement. The framework is judged on realism, correctness and generality. The first two aspects signal conformance to the IEEE 802.11 standard [92]. Generality implies consideration of influence of all MAC arbitration mechanisms for a performance analysis that stays valid under all settings. Within the framework, quantitative analysis and mappings of requirements to network configurations are developed. Contributions per research question are stated as follows.

Question 1. Addressing this question provides the necessary foundation of our performance framework. We build a conceptual model for our system. It covers the two issues as raised in the question: MAC reservations and the QoS context of our system. Contributions per subquestion are as follows:

a. MAC reservations that fulfill given QoS requirements leverage on a distributed control. A foremost step in achieving this control is dissection of complex MAC functionality that may lead to appropriate abstraction. For our purpose, we select this abstraction to be a network cyclic behavior that occurs disregarding particular network configurations. We formally define network cycles that transcend specific functionalities of MAC arbitration mechanisms coherently. They leverage adoption of a methodical approach to develop quantitative performance analysis, as aimed in this thesis. This abstraction can be identified in a simplified algorithm and a State Transition Diagram of EDCA protocol that we propose.

b. The QoS requirements of applications vary with their types. For instance, delay is the main concern for time sensitive applications, bandwidth/reliability for time insensitive file transfer and delay/reliability for control messages. We approach this question with an aim of accommodating all main application types. To this end, QoS semantics are introduced that can formally specify the QoS requirements of applications as a tuple of channel access frequency, channel share, MAC latency and MAC reliability. The QoS requirements are met by an appropriate allocation of the channel, whose allocation is controlled by MAC. The satisfiability of the QoS requirements is asserted by introducing a QoS fulfilment relation. Relevant to the QoS semantics is a need to handle insufficiency in channel capacity against some set of QoS requirements. To this end, a notion of priority semantics is offered formally that can order applications by their criticality. Thus, ordering applications allows to select a subset of applications, whose QoS requirements can be met by the available channel capacity.

Question 2. To address this question, we introduce a formal theory for a trace framework of a CSMA/CA based network. Within the framework, contributions per subquestion are made as follows:

a. The combined effect of states of processes determines the network state. We express this relation formally.

b. The MAC process changes states due to the corresponding events. We introduce events with attributes, forming the basis of low-level performance metrics.

c. In general, randomness based MAC protocols undergo performance fluctuations, which are pronounced in EDCA, specially due to network configurations that create extreme disparities between processes. In order to understand these fluctuations, we develop a formal reasoning mechanism. It is based on process-specific and MAC event-specific cyclic behaviors that we identify.

d. Given a trace, functions per QoS concern are introduced, mapping to QoS properties of a process, derived on both short-term and long-term durations.

Question 3. A Markov chain based model, similar to [201] is developed, to map network configurations to an average network behavior that is represented by stationary probability distributions, one per QoS class. This model stays valid under the extreme network configurations to allow an exhaustive investigation of their impact on network behavior. The model is represented by a system of linear equations that are solved numerically with a fixed point approach. To this end, performance of our implementation in terms of memory required, iterations needed and time taken, is thoroughly reported for various network configurations.

Question 4. This question is addressed by deriving functions, one per QoS concern, mapping stationary probability distributions to QoS properties of the MAC processes. Effectively, these mappings create, one per QoS concern, multi-level partition views of the channel.

Randomness based MAC protocols, such as EDCA, exhibit QoS performance fluctuations inherently. Their usage becomes questionable in transmitting MPDUs with hard real time requirements on each of them. For instance, such traffic represents control signals in industrial and medical environments. To cater for these stringent requirements, we also investigate achieving a bounded delay on transmission of each MPDU. To this end, we develop a worst case delay model.

Question 5. A mapping is developed from QoS requirements to network configurations in accordance with the QoS fulfilment relation. A side effect of this relation is the fulfilment of given QoS requirements by multiple network configurations — thus calling to resolve the multiplicity issue. In the solution process, we define multiple objectives, possibly conflicting with each other, on which optimality is sought. Each optimal solution offers a tradeoff point. Multiplicity in optimal solutions is resolved by seeking a global optimal solution.

Question 6. To address this question, we offer a policy-based proactive approach, wherein the MAC states are influenced by the application properties to minimize channel wastage. Enhancement of the MAC performance by enriching its knowledge has the potential to improve channel utilization.

Question 7. This question is addressed by a reactive approach, wherein system aims to gain on channel utilization by reacting to interception of implicit suboptimality signals. In this approach, processes monitor continuously and autonomously, the channel usage by their competitors. After each monitoring interval, the network configuration may be updated with an aim to minimize channel wastage. Based on the application properties and the pessimistic network states, which are those resulting from pessimistic reservations on the network, such autonomous adaptations can result in channel utilization improvement.

Finally, all the developed theoretical tools are validated and verified by software tools that we built, which include our Markov chain solver and custom made discrete event simulator for EDCA. Publications related to the work in this thesis are given as a separate list.

1.4 Outline of thesis

The thesis has five main chapters, which describe the performance framework. All of them are self-contained in the sense of research questions they address.

Chapter 2 - this chapter presents a conceptual model of the performance framework. It includes an abstraction of EDCA and the QoS semantics to address research **Question 1**. The abstraction features events and state transitions of EDCA, which are augmented by description of its execution behavior using a representative pseudocode that conforms to the IEEE 802.11 standard [92]. We consider these details to be imperative to understand the distributed nature of CSMA/CA. EDCA detailed states are abstracted into two main states, contention and transmission, whose interleaving creates cycles - hence, lends definition for the network cyclic behavior. This cyclic behavior, also identified in trace framework presented in Chapter 3, underpins the derivation of quantitative stochastic models for QoS properties of applications in Chapter 4. The QoS semantics introduce a QoS specification scheme and a QoS satisfiability relation. A related concept of priority semantics is also introduced for call admission control to deal with insufficient channel capacity.

Chapter 3 - this chapter investigates research **Question 2**. We analyse performance fluctuations in the network behavior by formally defining a trace framework for network execution event logs. The developed theory is based on restrictions, which can project behaviors of interest. The trace framework enables identification of network states from event logs of MAC processes, derive MAC process and MAC event specific cycles, determine low-level performance metrics to gain deeper insights into an impact of a network configuration and detect presence of QoS properties in an arbitrary event log.

Chapter 4 - this chapter considers research **Question 3**, research **Question 4** and research **Question 5**, while it is based on [4, 6]. For research **Question 3**, it presents an analysis that leads to formulating the network steady state as a system balance equation (SBE). Effectively, SBE maps a network configuration to average behaviors of all MAC processes in the form of their stationary probability distributions. The fixed point of SBE is sought iteratively. The performance of the process is reported in terms of computational resources consumed, which are computation time and memory space. For research **Question 4**, stochastic models are developed to predict QoS properties. These models are derived as QoS concerns specific mappings from stationary probability distributions. For research **Question 5**, QoS requirements of applications are mapped to some network configuration, objectively. This approach allows consideration of multiple criteria in decision making and possible tradeoffs on conflicting objectives.

Chapter 5 - this chapter contributes to research **Question 6**, while it is based on [5]. For given QoS requirements of applications and network configuration, it investigates a quantitative method to improve channel utilization. The given method is policy-driven, thus allows to compare impacts of different policies.

Chapter 6 - this chapter, based on [7], addresses research **Question 7**. It considers over time, pessimistic reservations on channel against QoS requirements of applications. Detection of such reservations is signalled as a channel wastage that needs to be avoided by adapting network configuration.

Finally, conclusions are drawn in Chapter 7. Here, recommendations and a glimpse of future work, are also given.

2

A conceptual model

The purpose of this chapter is to build foundational concepts of our system. These include abstract views on the EDCA process/network and the QoS context. The EDCA network is also referred to as a system in this thesis. The EDCA views present the interaction patterns of the EDCA mechanisms as an overall cyclic system behavior, which leverages the quantification of the performance framework. The framework is the major contribution of this thesis — developed over these views, it admits performance analysis for accurately predicting system parameters to achieve a certain performance. The predictability can assist network architects to design WLANs that satisfy the QoS demands of applications. This chapter focuses on the following research question as posed in Chapter 1:

Question 1. What is the conceptual model of the desired QoS based MAC reservations on the channel?

Within its scope, following two subquestions are identified:

a. How to abstract the MAC behavior towards a performance model?

b. How to specify the communication related QoS requirements of applications?

This question is addressed by defining a performance framework. We set off by presenting the requirements and challenges in the form of objectives that this framework shall meet.

For subquestion a., a technical summary of EDCA is given, including a simplified algorithm and a State Transition Diagram — in the sense of captured executional generalizations. The EDCA mechanisms are activated at different phases during the operation of a system, and exhibit a system-wide joint influence. In reality, the system is found in either one of the following two major states: contention and transmission, however, the activation patterns of the EDCA mechanisms are complex, thus unpredictable [17, 247]. Therefore, it is assumed that the system oscillates between these two major states. This state abstraction holds for both Media Access Control (MAC) and Physical (PHY) layers.

This abstraction supports identification of key notion of system cyclic behavior that we aim for. It is represented by emergent system cycles, defined formally in this chapter. The system cycle view leads to a virtual partitioning of the network into two blocks: an arbitrary reference process and a competing network. This partitioning leverages an assessment of the impact on the reference process as offered by the competing network. The benefit of this approach is its ability to directly transform into an equivalent stochastic process, which we model to quantify the system performance in Chapter 4. This model is precise and accurate.

In context of subquestion b., we notice that the IEEE 802.11 standard [92] has introduced QoS classes with priority levels, as enforced by the recommended setting, also termed as the default setting. The priority levels imply mutual preferential treatment to traffic of different classes, though it does not specify how to rank the criticality of traffic qualitatively and quantitatively. Hence, the priority refers only to the implementation settings. Therefore, the following two problems are identified due to this insufficiency and incompleteness:

- The set of QoS requirements of applications that is satisfied by the default setting, is absent;
- The QoS properties of applications that emerge as a result of this default setting, are not mentioned.

In order to resolve these issues, we consider priority assignment and priority implementation as two separate problems and define both of these problems formally. This separation enables to highlight the major problem of implementation as tackled in this thesis. The implementation is dealt with by defining QoS semantics, which include specification of the QoS requirements of applications and a method to fulfill those requirements.

The performance framework relies on two major system aspects. One, the frequency of accessing the system cycles. Two, the duration of transmission in those system cycles that needs to be considered. This consideration leads to identifying two main concerns: *gaining control* and *retaining control* of the channel. These concerns influence the system performance differently as quantified in Chapter 4. The quantification is generic enough to consider all MAC parameters and detailed enough to allow predicting the properties of the system cycles of arbitrary durations and types.

2.1 Related work

Functionally, our performance framework is required to predict the system parameters that configure the EDCA network to exhibit the desired QoS properties. In general, the QoS properties of applications over a communication network can be controlled at multiple OSI layers — application, transport, network and DLL. Network based solutions are proposed for backbone segments that are out of scope of this thesis. The transport and the DLL based solutions rely on flow control services as provided by them, though with different network scopes. The flow controls over a WLAN as provided by these two layers differ in preciseness of the QoS properties. To this end, the research investigations are directed towards deriving performance curves, consideration of specific applications and seeking optimizations.

Throughput/delay curves of uplink/downlink traffic of TCP over multirate WLAN are presented in [111]. Inherently, TCP over WLAN based connections cannot differentiate between data losses due to congestion in routers and high contention levels in WLANs. Hence, TCP reacts to such data losses transparently, which leads to suboptimal performance. This behavior is rooted in TCP semantics offering an end-to-end reliable and in-order delivery service, regardless of its timeliness. Upon need of transporting multimedia streams, a consideration of timeliness is proposed in TCP-RTM [128], though the TCP semantics are compromised. Based on this seminal work, other related proposals have emerged that treat losses due to congestion and contention separately [158, 147, 195, 136, 163]. To this end, TCP variants such as Snoop TCP [11] and SPLADE [164] are proposed to shield the transport layer from the effects of WLANs. The gain is in the form of performance degradation avoidance by optimization on timeliness of data.

An analytical performance of streaming services over HTTP/TCP is derived using a Markov chain [220]. This work proposes to maintain a TCP throughput twice that of the source in order to mitigate the fluctuations due to additive-increase/multiplicative-decrease (AIMD) behavior of the congestion control algorithm in TCP. An extension of TCP-Reno has employed the AIMD algorithm to minimize the difference between the target/achieved rates [183]. A scheme in [95] has proposed to tune the TCP congestion window based on the wireless channel occupancy to improve on packet losses. Fuzzy controller based stream QoS adaptations are proposed in [114]. The model seeks to minimize the difference between the stream qualities at the server/receiver sides.

Given limited buffers, a streaming system needs to adapt the TCP sender rate to the network conditions — otherwise frames may be dropped uncontrollably that can deteriorate the quality on the receiver side. To this end, TCP sender rate is adjusted to the long/short term network fluctuations [192]. Two levels of adjustments are aimed to keep the system state closer to the actual network conditions. The content compression has employed layered encoding, which generates a base layer and one or more enhancement layers. The long term measure adjusts the base layer rate, while the short term measure employs selective dropping in the application layer interface buffers. In order to lessen the impact of missing frames on the receiver side, encoding aims at minimal intra enhancement layer frame dependencies.

The DLL based solutions are geared towards performance curves mostly derived from analytical approaches or simulations. Under Poisson arrivals, a Markov chain model for throughput and delay has accounted for all major EDCA mechanisms [96]. Packet loss is a significant QoS concern that also affects other QoS concerns. Losses in a WLAN can occur at network layer interface buffers due to overflow in them, and at sender MAC processes due to exhaustion of allowed MAC retry attempts. Retry attempts are triggered due to link errors and/or MAC collisions that result in failed transmissions. Given a traffic arrival rate, link condition and the number of MAC processes, an optimal limit on the MAC retry attempts can be sought heuristically that minimizes losses [126]. Admission control is fundamental to ensure QoS features in a network. A QoS framework, DACME-SV, for streaming services over a mobile ad hoc network (MANET), is duly supported by a distributed admission controller [39]. A recent proposal, iPAS [240], provides a scheme of allocation of resources based on the QoS demands — a highly desirable approach for QoS driven systems. The extent of QoS differentiation is adjusted dynamically, according to the network conditions. The performance is measured by closeness between the demanded/obtained QoS using Jain's fairness index [103]. Such a performance indicator reflects upon fairness in proportional allocations without considering the priority of the QoS demand or the extent to which the demand is met.

We aim for a QoS demand driven, formal specification based framework that is transparent over application types and whose performance can be predicted using analysis leading to mathematical relations.

2.2 Objectives of the proposed framework

In general, a QoS-aware resource model should consider, possibly distinct requirements of resource demanders. Similarly, a QoS-aware WLAN should provide applications with demand-specific transport services. These demands typically imply time-constrained scheduling of variable size data segments over a shared wireless channel. The general objective of the performance framework is to meet the requirements of the applications by scheduling them feasibly at the MAC layer. Sharing the wireless channel to achieve a feasible schedule over it requires ascertaining a control over the channel division problem. In this thesis, the scheduling is solved by controlling the wireless MAC parameters, whose tuning generate the schedules. The mapping of these MAC parameters to the required schedules is what we call our performance framework.

The performance framework shall meet the following qualitative and quantitative objectives:

- The predictability of network QoS shall be application independent so as to accommodate the main application types, which include those generating time sensitive and/or time insensitive data traffic;
- The framework shall grant QoS specifiability and comparability;
- The framework shall allow to assess the following QoS properties: channel access frequency, channel share, MAC latency and MAC reliability;
- The performance analysis and the corresponding developed model shall enable exploring the parameter space exhaustively and timely;
- The performance analysis shall be accurate and based on realistic conjectures, with the purpose of correct predictions. By realistic conjectures, we refer, for instance, to the following system aspects:
  - After a transmission, the probability of winning the channel arbitration varies over slotted time;
  - The probability of collision changes over different retry attempts;
  - Each backoff process can avail a limited number of retries;

  This is in contrast to the simplified models in the literature [17, 182, 212, 216, 81, 221, 117, 241, 19, 18, 67, 52, 63, 112, 89, 246, 30, 233, 245, 49, 13, 106, 215, 202, 12];
- Along with long-term performance prediction that indicate an average behavior, the developed model shall also allow predicting the short-term fluctuating behavior.

2.3 A summary of EDCA

Now that the objectives of the performance framework are set, there is a need to summarize the technical details of EDCA, which are abstracted appropriately towards addressing subquestion a.. This section, therefore, presents the governing principles of EDCA. Out of *infrastructure* BSS and IBSS modes of operation, this thesis concentrates on IBSS mode only, in which the devices form an ad-hoc network for which a QoS-aware CSMA/CA access is provided by EDCA.

2.3.1 MAC parameters

EDCA provides the following five MAC parameters, which can be tuned to address the QoS requirements: the arbitration interframe space ($AIFS$), the minimum contention window size ($CWmin$), the maximum contention window size ($CWmax$), the number of retry attempts allowed ($RetryLimit$) and the transmission opportunity limit ($TXOPLimit$), as illustrated in Figure 2.1.

In order to obtain access to the medium, each device executes a backoff process, which repeatedly senses the channel until it is free, followed by silence period of arbitrary length that is ultimately terminated by the transmission of the next MPDU[1]. Channel time during

[1] We use the acronym MPDU (MAC Protocol Data Unit) for MAC frames in order to discriminate them from video frames later.

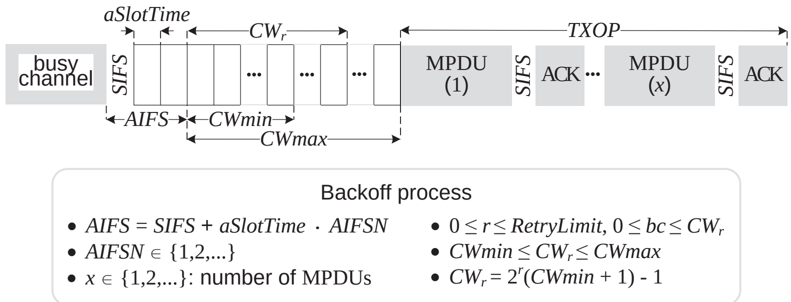


Figure 2.1: The EDCA channel access mechanisms.

silence is slotted and the participating devices synchronize on slot boundaries using waiting times derived from the parameters settings. For being consistent, the nodes keep a uniform view of slots and they are allowed to start transmission only at slot boundaries. The slot size, $aSlotTime$, is defined as the worst case time a device must wait before it can reliably know about the idleness of the slot. Hence, $aSlotTime$ depends on the propagation time of the physical (PHY) layer used.

For a particular process, $AIFS$ represents the mandatory deferral time after the ongoing transmission on the channel ends: $AIFS = SIFS + aSlotTime \cdot AIFSN$, where $AIFSN$, the arbitration interframe space number, represents the number of $aSlotTime$s after a short interframe space ($SIFS$). The $AIFS$ deferral guarantees a fixed delay for the process before it can possibly transmit or update its backoff counter (bc), $bc \in \mathbb{N}_0$[2]. Devices with larger $AIFSN$ can access the channel less frequently and consequently, they get a lower portion of the bandwidth.

The state of a backoff process is defined by the current values of its bc and retry counter (rc), $rc \in \mathbb{N}_0$. The value of bc represents the duration a device should refrain from transmission, in units of an $aSlotTime$. This counter is assigned a value if the channel is found busy on arrival of an MPDU in the send queue. Under all events [92, Section 9.9.1.5], the assigned value is chosen randomly from the uniform distribution: $bc \sim \mathcal{U}(0, CW_r)$, where $CW_r = \mathtt{cw}(r)$ is the contention window at retry attempt, r. $\mathtt{cw}(r)$ grows exponentially with the retransmission attempts, from a minimum of $CWmin$, using stages given by $\mathtt{cw}(r) = 2^r\,(CWmin + 1) - 1$, until a maximum of $CWmax$.

When the channel has no transmission for an $AIFS$ or each subsequent $aSlotTime$, then at the $AIFS$ boundary or the subsequent slot boundary, either one of the following two mutually exclusive events can occur:

- if $(bc > 0)$, $bc \leftarrow bc - 1$;
- if $(bc = 0)$, a transmission starts.

For the first transmission attempt, $\mathtt{cw}(0) = CWmin$. Intuitively, devices with a smaller $CWmin$ draw from a lower range of values for bc, and have to wait less on average; hence, they are rewarded with more frequent transmissions, thereby deriving larger portions of the channel capacity.

A successful reception of an MPDU triggers the corresponding transmission of an acknowledgement frame (ACK). Retransmissions are triggered when a timeout occurs at the

[2]In this thesis, $\mathbb{N}_0 \triangleq \{0, 1, 2, \cdots\}$, $\mathbb{N}_1 \triangleq \{1, 2, \cdots\}$ and $\mathbb{Z} \triangleq \{\cdots, -2, -1, 0, 1, 2, \cdots\}$.

transmitter due to an unacknowledged MPDU — an event, which can occur due to a collision of MPDUs from different processes or when the receiver is out of reach. If the colliding processes reside at the same node, then it is an internal/virtual collision, otherwise an external collision. In this thesis, a collision refers only to an external collision, unless stated otherwise. Additionally, an MPDU or an ACK can be lost or corrupted in an impaired channel, thus also triggering retransmissions.

$TXOPLimit$ bounds the time a transmitter may transmit after having obtained access provided its send queue is not empty whilst the channel is under control. Backoff processes with a larger $TXOPLimit$ setting have longer contention free access to the channel and consequently obtain larger share of the channel capacity.

Each listener device keeps track of the duration of the channel usage using the Network Allocation Vector (NAV), which is a field in an active MPDU. The NAV can be updated in case of an unsuccessful delivery of a transmitted MPDU. After the expiry of their NAVs, the contender devices defer for at least an $AIFS$ duration. This behavior is described in detail in the next section.

2.3.2 Behavioral specifications

Each backoff process is configured separately and executes autonomously, thus all processes together constitute a *network of processes* (NoP). Although autonomicity here refers to local decisions taken independently by the processes themselves, the frequencies of different decisions depend strongly on the extent of cooperation and competition by other processes. These decisions are about the process state transitions, which include decrementing the *bc*, updating the CW and redrawing a new value for the *bc*. For an operational system, a chart of decision frequencies is a key indicator for system performance. Thus, the network is distinguishably a tightly coupled system that must be examined as a whole.

An IEEE 802.11e compliant transmitter process must execute the EDCA functionality sketched in Algorithm 1. We propose this simplified algorithm to model the behavior described in the previous section. The parameters described there, viz., $AIFSN$, $CWmin$, $CWmax$, $RetryLimit$, $TXOPLimit$ are input to it, to govern its behavior, though the degree of their influence depends on their mutual settings. The Logical Link Control (LLC) sublayer passes data to the MAC sublayer via Access Category (AC) queue that is specific to some application (cf. Figure 2.2). Conversely, each executing application in the system, has an exclusive AC assigned at the MAC sublayer. Data enqueued in an AC, contends for the channel access by the AC-specific backoff process.

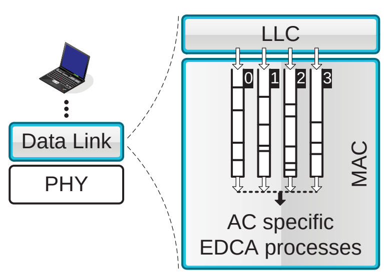


Figure 2.2: MAC layer AC-specific queues.

The presented pseudocode transmits MPDUs perpetually by scheduling them in units of transmission opportunities. In each transmission opportunity, multiple MPDUs may be sent,

Algorithm 1 EDCA()
// behavioral specifications of CSMA/CA priority arbitration

Require: $AIFSN \in \mathbb{N}_1$, $CWmin \in \mathbb{N}_0$, $CWmax \in \mathbb{N}_0$, $RetryLimit \in \mathbb{N}_0$, $TXOPLimit \in \mathbb{N}_1$

Ensure: EDCA compliant behavior

```
while (true) do
    MACset(0); // backoff process initialization
    backoff(AIFSN, bc);
    // mandatory contention before the next transmission opportunity
    wait on empty AC queue;
    HoLmpdu ← new MPDU;
    active ← true; // active ∈ {true, false}
    fresh ← true; // fresh ∈ {true, false}
    while (active) do
        if (¬fresh or CCA() = busy) then
            // CCA : ∅ → {idle, busy}: clear channel assessment function
            backoff(AIFSN, bc); // contention
        end if
        transmitTXOP(RetryLimit, TXOPLimit); // transmission
    end while
end while

return
```

while it is only the first MPDU that can undergo failed attempts. In case of $RetryLimit + 1$ failed attempts, this first MPDU is dropped.

The start of each transmission opportunity is preceded by a mandatory backoff given by backoff($AIFSN, bc$) in Algorithm 2. This backoff is done at the smallest CW, which is $CWmin$, and invoked by setting the MAC to its initial state via a call, MACset(0). MACset($retry$) in Algorithm 4 is used to update rc and CW as well as to assign a new random value to bc: $bc \sim \mathcal{U}(0, CW)$. Whenever a new MPDU is promoted to the head-of-line (HoL) position in the AC queue, two boolean flags are defined. The flag $active$ tracks the lifetime of an MPDU, while the flag $fresh$ checks if it is the first transmission attempt. If the channel is sensed idle during the first attempt, it is deemed as a state of unsaturation on the network, so that it is transmitted opportunistically. Otherwise the process waits in backoff.

The EDCA functionality relies majorly on knowing the channel status in the presence of ongoing transmissions. Hence, a separate clear channel assessment function, CCA : $\varnothing \to \{\text{idle}, \text{busy}\}$ is defined. It senses and reports on the channel being physically and virtually (in)active, wherein the latter type of report is indicated by the NAV.

The backoff($AIFSN, bc$) starts in phase I, during which the process waits on the busy channel, till the end of some transmission. After this event, two types of ordered waiting periods may occur. These waiting periods are termed as phase II and phase III. These phases are possible only if the channel is idle. Whenever either of these phases is interrupted by some transmission from the competing network, the process resumes phase I. During phase II, the $AIFS$ deferral is completed. Phase III is divided into time steps of size $aSlotTime$ and used to decrement bc, till $bc = 0$. Before an MPDU is transmitted, phases II and III may be interrupted repeatedly, hence they may occur recurrently. Longer stays in phases II

Algorithm 2 `backoff`($AIFSN, bc$)
// EDCA compliant contention

Require: $AIFSN \in \mathbb{N}_1$, $bc \in \mathbb{N}_0$
Ensure: waiting according to rules

```
while (true) do
    // phase I: waiting on busy channel
    wait till (CCA() = idle);

    // phase II: AIFS deferral
    AIFS ← AIFSN · aSlotTime + SIFS;
    wait for AIFS while (CCA() = idle);
    if (AIFS deferral is incomplete) then
        continue // resume phase I
    end if

    // phase III: waiting for aSlotTime
    while (true) do
        if (bc = 0) then
            return  // backoff is complete
        else
            bc ← bc − 1;
        end if

        wait for aSlotTime while (CCA() = idle);
        if (aSlotTime waiting is incomplete) then
            break // resume phase I
        end if
    end while
end while

return
```

Algorithm 3 `transmitTXOP`($RetryLimit, TXOPLimit$)
// transmit multiple MPDUs

Require: $RetryLimit \in \mathbb{N}_0$, $TXOPLimit \in \mathbb{N}_1$
Ensure: EDCA compliant transmission opportunity

```
txop ← 0;

do
    transmit(HoLmpdu); // HoL MPDU transmission
    receive(ACK);
    if (¬ACKTimeout()) then
        // successful MPDU delivery
        // ACKTimeout : ∅ → {true, false}
        txop ← txop + 1;
        active ← false;
        MACdrop(HoLmpdu); // HoL MPDU expunging
        if (txop < TXOPLimit) then
            if (AC queue is not empty) then
                HoLmpdu ← new MPDU;
                fresh ← true;
                active ← true;
                MACset(0); // backoff process initialization
                wait for SIFS while (CCA() = idle);
                // interframe gap between MPDUs transmission
                // holding the transmission opportunity
            end if
        else
            break // a fully availed transmission opportunity
        end if
    else
        if (rc < RetryLimit) then
            fresh ← false;
            MACset(rc + 1); // MAC variables update for a retransmission
            break // transmission opportunity end
        else
            active ← false;
            MACdrop(HoLmpdu); // expunge the HoL MPDU
            break // transmission opportunity end
        end if
    end if
while (txop < TXOPLimit)

return
```

Algorithm 4 `MACset`($retry$)
// MAC variables updation

Require: $retry \in \mathbb{N}_0$
Ensure: reassignment of values to rc, CW and bc

$rc \leftarrow retry$; // rc: retry counter
$CW \leftarrow \texttt{cw}(rc)$; // CW: current contention window
$bc \leftarrow \texttt{random}(0, CW)$; // bc: backoff counter

return

Algorithm 5 `MACdrop`($HoLmpdu$)
// expunge the HoL MPDU

Require: $HoLmpdu$ pointer
Ensure: memory release

$HoLmpdu \leftarrow \varnothing$;

return

and III allow the channel to be accessed by other processes.

Note that an MPDU is transmitted only when the process has waited for an *extra slot* after the assignment $bc \leftarrow 0$. Waiting for this *extra slot* is required by the IEEE 802.11 Standard [92, Clause 9.9.1.3], which allows at the end of the slot, either transmission of an MPDU or decrementing bc. This ensures performance compatibility of the current IEEE 802.11 Standard [92] with previous IEEE 802.11 Standard [91]. The compatibility issue has also been reported in [20, Ch. 4].

On acquiring the transmission opportunity, the actual transmission is done via a call to `transmitTXOP`($RetryLimit, TXOPLimit$) in Algorithm 3, which may transmit a series of $TXOPLimit$ MPDUs. It is only the first MPDU of this series that contends for the channel by waiting in phases II and III, subsequent to which $SIFS$ is used as a gap before each next MPDU, to ensure channel control. The successful delivery of an MPDU is signalled explicitly in a stop-and-wait arrangement, wherein its ACK must be received before the expiry of a timer that is signalled by a function, $\texttt{ACKTimeout} : \varnothing \rightarrow \{\text{true}, \text{false}\}$. For each successful delivery, an active MPDU is expunged by a call to `MACdrop`() in Algorithm 5. If the transmission opportunity is not complete yet and an AC queue is still saturated, then `MACset`(0) is called to initialize the process for transmitting the next MPDU. This opportunity is lost in case of a timeout for the active MPDU. If another retry attempt is still allowed for the active MPDU, then EDCA reacts to it by calling `MACset`($rc + 1$), which updates the MAC variables rc, CW and bc appropriately for a retransmission; otherwise the active MPDU is dropped by calling `MACdrop`($HoLmpdu$).

In this thesis $TXOPLimit \in \mathbb{N}_1$ holds the maximum number of MPDUs that can be transmitted in a given opportunity, in contrast to its meaning in the IEEE standard [92] where it represents a time quota. This deviation removes the inherent dependency of the number of MPDUs transmitted in the allotted time on an MPDU size. Hence, it explicates the allowed amount of data transmitted. In the time quota based semantics, if the remaining time is not enough to transmit the next MPDU, then the transmission opportunity is aborted.

Corresponding to the EDCA sender process pseudocode in Algorithm 1, we also illustrate triggering of its different states in Figure 2.3.

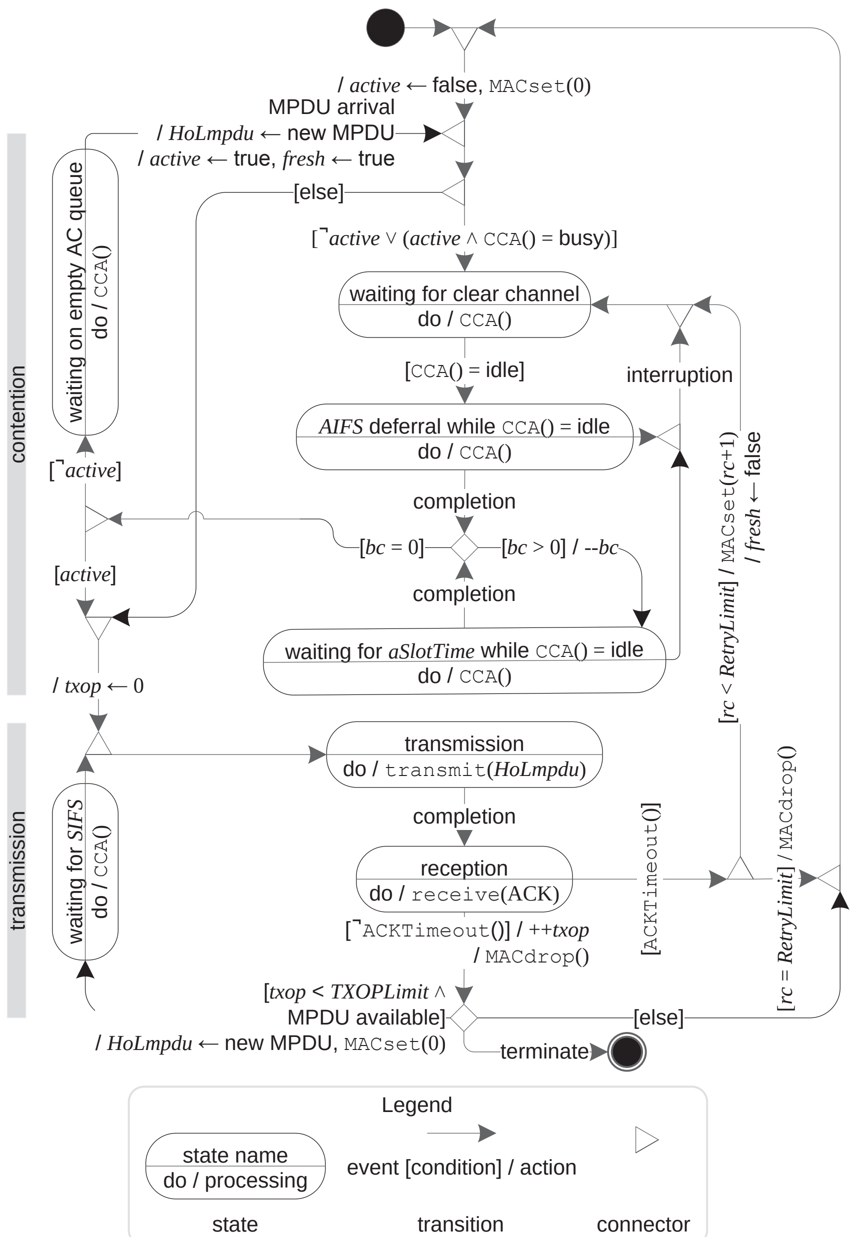


Figure 2.3: The State Transition Diagram of the EDCA sender process.

2.4 A system view

In this section, we present the states and transitions, which the NoP undergoes as a single MAC system. In this MAC system, we aim at identifying aperiodic system cycles, which are considered for further model development. The NoP is viewed as a group of QoS class clusters within a single broadcast range (cf Figure 2.4). Each QoS class cluster has member processes with the same parameter settings, herein referred to as a parameter tuple:

$$P\text{-tuple} \triangleq (AIFSN, \mathbf{CW}, TXOPLimit)\,, \tag{2.1}$$

where vector **CW** has dimension $RetryLimit + 1$. These parameters have limited domains. The number of processes per class is denoted by the vector $\mathbf{n}$. Hence, the number of processes in the NoP is $N = \sum_{q \in \mathcal{Q}} n_q$. Based on P-tuple, different classes are defined and indexed into a class set of $\not{q} + 1$ classes:

$$\mathcal{Q} \triangleq \{0, 1, 2, \cdots, \not{q}\}\,. \tag{2.2}$$

A given class differs from other classes on settings of one or more parameters taken from the P-tuple: $q, q' \in \mathcal{Q}, q \neq q' \implies (P\text{-tuple})_q \neq (P\text{-tuple})_{q'}$. The presence of a class is exhibited by inclusion of its, at least one backoff process in the NoP. We regard priority assignment and implementation as two separate problems — both are treated in Section 2.6. For convenience, the processes are indexed in a fixed but arbitrary manner into a process set:

$$\mathcal{K} \triangleq \{1, 2, 3, \cdots, N\}\,. \tag{2.3}$$

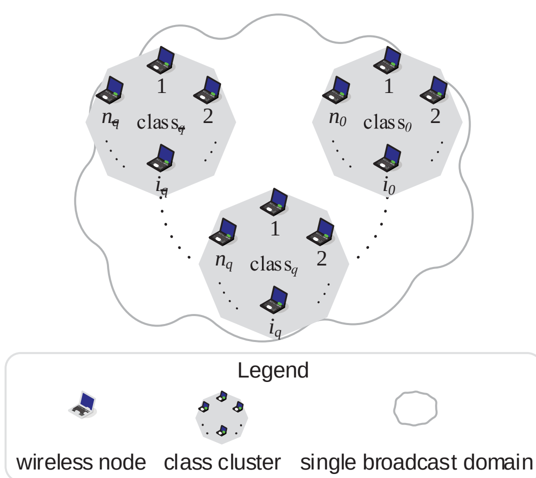


Figure 2.4: NoP: a logical view on QoS classes.

2.4.1 State transition systems

The backoff process states presented in Figure 2.3 are abstracted into two main states: contention and transmission, respectively:

$$\underset{\text{mac}}{\mathcal{S}} \triangleq \{\mathsf{contention}, \mathsf{transmission}\}\,. \tag{2.4}$$

Each backoff process is always found in either of these two states. By combining the individual states of process(es), we generate the states of the NoP. There are 2^N such combinations, while they are classified into two groups using their characteristics: group one has the one and only combination with all processes in contention state, while group two has the remaining combinations. Note that, all combinations in the second group have at least one process in state transmission. We recognize that these two groups, in fact, represent contention and transmission states of the NoP, respectively. Conveniently these two recognized states are members of the existing set $\underset{\text{mac}}{\mathcal{S}}$. This establishes a key conclusion that the states of an individual process and a group of processes are drawn from a single set $\underset{\text{mac}}{\mathcal{S}}$.

A transmission state (including the time for the corresponding ACK) of the network may represent a successful delivery of an MPDU or a collision, hence within transmission state, we discriminate between substates retention and collision. The NoP makes a transition to the state contention when all processes are silent, hence the PHY channel has no ongoing activity. For each recognized state of the NoP, there is a direct impact observed at the PHY channel. Hence, this one-to-one mapping is represented by the corresponding states of the PHY channel:

$$\underset{\text{phy}}{\mathcal{S}} \triangleq \{\mathsf{idle}, \mathsf{busy}\}\,.$$

The conceptual model of both MAC level NoP and PHY channel transition systems (intra-layer) and their mapping (inter-layer) is presented in Figure 2.5. At the PHY channel, its not always possible to distinguish between retention and collision sub-states due to PHY phenomena such as fading, interference and cancellation.

2.4.2 A dynamic system cycle

In this section, we identify and formally present the system cycles, which are instrumental for the performance analysis later. Due to its random nature, the NoP transition system transits between its states nondeterministically. Consequently the time intervals during which the NoP system dwells in these states cannot be determined in advance. Its behavior is cyclic, but not periodic. The lifetime of each state, state is represented by two events: start(state) and end(state). We identify the cycle boundaries in the NoP transition system, using the event end(transmission), which happens at random times.

Definition 2.4.1. The i^{th} MAC system cycle $\underset{\text{mac}}{\mathcal{O}}^{(i)}$ occurs between two events, viz., the termination of the $(i-1)^{\text{th}}$ and the i^{th} transmission states. Formally, it is represented by a tuple:

$$\underset{\text{mac}}{\mathcal{O}}^{(i)} \triangleq \left(\mathtt{end}\left(\mathsf{transmission}^{(i-1)}\right), \mathtt{end}\left(\mathsf{transmission}^{(i)}\right)\right). \tag{2.5}$$

□

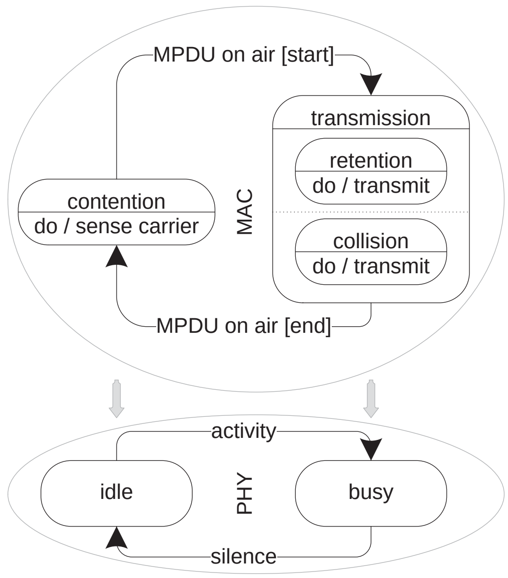


Figure 2.5: MAC (NoP) and PHY channel transition systems.

In this cycle notation, the superscript within braces, always shows the cycle number in a given system trace, irrespective of which transmission substate delimits it. The state transmission in each cycle is always prepended by a period of contention. The two events $\texttt{end}\Big(\mathsf{transmission}^{(i-1)}\Big)$ and $\texttt{start}\Big(\mathsf{contention}^{(i)}\Big)$ occur simultaneously. Accordingly, two types of cycles are distinguished:

- $\Big(\texttt{start}\Big(\mathsf{contention}^{(i)}\Big), \texttt{end}\Big(\mathsf{retention}^{(i)}\Big)\Big)$: corresponding to a successful delivery of at least one MPDU;
- $\Big(\texttt{start}\Big(\mathsf{contention}^{(i)}\Big), \texttt{end}\Big(\mathsf{collision}^{(i)}\Big)\Big)$: corresponding to a collision, in which only the first MPDU of the transmission opportunity collides.

The system cycles emerge as the result of parallel execution of the (non-deterministic) individual processes. The duration of system cycles vary due to:

- a random number of contention slots. The occurrences of these non-deterministic intervals of silence, stem from the random nature of EDCA;
- potentially different time durations for the substates retention and collision.

Consequently, for an outside observer of the channel, the system cycles are stochastic. As stated, retention reflects an exclusive access to the channel, which is the result of winning the contention process.

The PHY system cycle is defined similarly.

Definition 2.4.2. The i^{th} PHY system cycle $\underset{\text{phy}}{\mathcal{O}}^{(i)}$ occurs between two events, viz., the termination of the $(i-1)^{\text{th}}$ and the i^{th} busy states. Formally, it is represented by a tuple:

$$\underset{\text{phy}}{\mathcal{O}}^{(i)} \triangleq \left(\texttt{end}\left(\mathsf{busy}^{(i-1)}\right), \texttt{end}\left(\mathsf{busy}^{(i)}\right)\right). \tag{2.6}$$

□

Although the MAC substates to the PHY channel states mapping is a function, the reverse is only a relation. The MAC substates can not always be implied from the PHY channel states.

2.4.3 A network partition

Being a key enabler for resource sharing, the substate retention needs significant attention. Hence, an elaboration on the system events that lead to a retention is presented here. By its nature, retention involves a single process, i.e., the winner of contention. When the event start(retention) occurs for the winner, all other processes are contending. This observation justifies to conceptually partition the NoP into these two roles: a transmitter and contenders. This separation facilitates to quantify the competition, as offered by the contenders against the transmitter. The said competition is based on several events that occur during the EDCA execution. For brevity, we term these two roles as a *reference process* and a *competing network* respectively. Figure 2.6 illustrates the partitioning into two roles.

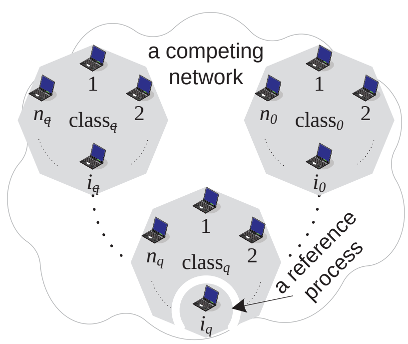


Figure 2.6: Partitioning of the NoP into mutually exclusive and collectively exhaustive blocks.

There are more ways to partition the NoP. For instance, a class based partition would consider an arbitrary class versus the rest of the classes. We selected the first approach, which is fundamental and easier to handle.

As already established in Section 2.4.1, set $\underset{\text{mac}}{\mathcal{S}}$ in (2.4) can represent the states of the reference process and the competing network directly. The combined states of these two blocks is given by the cartesian square of $\underset{\text{mac}}{\mathcal{S}}$:

$$\left(\underset{\text{mac}}{\mathcal{S}}\right)^2 \triangleq \left\{\left(s', s''\right) \mid s', s'' \in \underset{\text{mac}}{\mathcal{S}}\right\}.$$

These combined states are directly related to the states of the NoP. We explicitly state this relation by a mapping,

$$\texttt{Map} : \left(\underset{\text{mac}}{\mathcal{S}}\right)^2 \to \underset{\text{mac}}{\mathcal{S}}. \tag{2.7}$$

`Map(·)` is unfolded in Table 2.1. Based on the EDCA rules, impossible mappings are ruled out. All valid mappings are listed:

0. (contention, contention) → contention: a state of idleness on the PHY channel, generated by all contending processes.

1. (contention, transmission) → transimission: the NoP exhibits a transimission state due to a transimission from the competing network. If a single process from the competing network is transmitting, then it is a retention, otherwise a collision.

2. (transmission, contention) → retention: whenever the reference process is transmitting while the competing network is contending, then NoP has a retention substate.

3. (transmission, transmission) → collision: the NoP is in a collision state when both blocks are transmitting.

$\underset{\text{mac}}{\mathcal{S}}$ / $\left(\underset{\text{mac}}{\mathcal{S}}\right)^2$	contention	transmission: retention	transmission: collision
(contention, contention)	✓	✗	✗
(contention, transmission)	✗	✓ / ✗	✗ / ✓
(transmission, contention)	✗	✓	✗
(transmission, transmission)	✗	✗	✓

✓: valid ✗: invalid

Table 2.1: `Map(·)` in (2.7).

2.5 QoS semantics

From this section onwards, subquestion b. is investigated. Preferential treatment of different application types is fundamental in meeting their QoS requirements due to limited channel capacity. This preferential treatment gives rise to QoS classes. This section expresses the QoS semantics, which are used to solve the QoS problem, stated as follows.

The QoS problem How to specify and fulfill the transportation related QoS requirements of different applications, which compete for the channel?

Elements necessary to address the QoS problem are depicted in Figure 2.7, constituting what is termed as a QoS schema. The applications with corresponding QoS demands, compete for the shared channel, whose allocation is controlled by a network configuration, as represented by a P-tuple. As a result, applications have corresponding QoS achievements.

2.5.1 The QoS specification

For the system that we consider, the QoS requirements of applications are defined at the MAC level as follows:

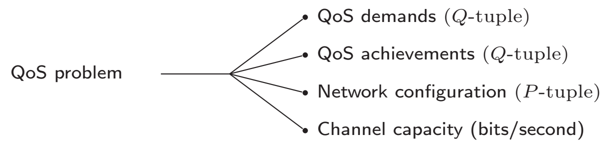


Figure 2.7: A QoS schema for the QoS problem.

channel access frequency is the average number of times a process has access to the channel in one second, denoted by $\nu \in \mathbb{R}$.

channel share is the average amount of data transmitted in one second, denoted by $\theta \in \mathbb{R}$.

MAC latency is the average time it takes to transmit an arbitrary MPDU, denoted by $\lambda \in \mathbb{R}$. It is measured from the moment an MPDU assumes HoL position till its transmission is complete.

MAC reliability is the MPDU delivery ratio, denoted by $\omega \in \mathbb{R}$.

When a QoS dimension is not considered, it is specified by an extremely low ranked value. The QoS requested and delivered for an application $k \in \mathcal{K}$ are denoted by e_k and d_k respectively, while they are specified as a Q-tuple:

$$Q\text{-tuple} \triangleq (\nu, \theta, \lambda, \omega) \, .$$

The delivered QoS is said to satisfy the requested QoS iff

$$\begin{aligned} e_k.\nu &\leqslant d_k.\nu, \\ e_k.\theta &\leqslant d_k.\theta, \\ e_k.\lambda &\geqslant d_k.\lambda, \\ e_k.\omega &\leqslant d_k.\omega \, . \end{aligned} \tag{2.8}$$

We denote this by $d_k \, \underline{\texttt{sat}} \, e_k$. From the definition of sat, it is an order (reflexive, transitive and anti-symmetric) representing increasing strictness in QoS. sat is extended to system scope by defining

$$\mathbf{d} \, \underline{\texttt{sat}} \, \mathbf{e} \triangleq \underset{k \in \mathcal{K}}{\forall} d_k \, \underline{\texttt{sat}} \, e_k \, . \tag{2.9}$$

The considered QoS dimensions are collected in a set,

$$\Xi \triangleq \{\nu, \theta, \lambda, \omega\} \, . \tag{2.10}$$

2.5.2 The QoS fulfillment

The schema elements in Figure 2.7 belong to two real-valued spaces: applications, $\mathcal{A}$ and network, $\mathcal{N}$ (cf. Figure 2.8). In this bispace schema, there are two transformation relations, mapping a given state in one space to the corresponding state in the other space. These relations allow emergence of two approaches to address the QoS problem. Both can be adopted independently, and each one offers a complete solution, with related merits and demerits. We present both approaches.

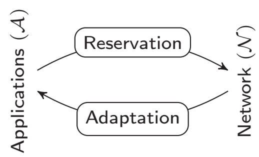


Figure 2.8: A space mapping for the QoS problem.

2.5.2.1 Application-aware networks

In this approach, applications make network reservations. The management is essentially based on a network reservations problem, which is complex. Additionally, the management generally aims to achieve some network related objective — such as channel utilization. Two issues that must be addressed are: what resources to reserve and how to reserve them. The space $\mathcal{A}$ represents the QoS requirements of applications. A mapping,

$$\mathtt{F} : \mathcal{A} \to \mathcal{N}, \tag{2.11}$$

determines a set of states in $\mathcal{N}$ that satisfy a given state in $\mathcal{A}$ (cf. Figure 2.9). A state in $\mathcal{N}$ is termed as a *network configuration point* (ncp). Generally, the QoS requirements of applications are specified independent of the channel capacity, hence there is no guarantee that the available channel capacity is enough to accommodate given QoS requirements. In case of a sufficient channel capacity, $\mathtt{F}(\cdot)$ is not empty. Otherwise the mapping to ncp is not possible and the QoS requirements have to be renegotiated. The renegotiation process may continue iteratively till some solution bails out the process.

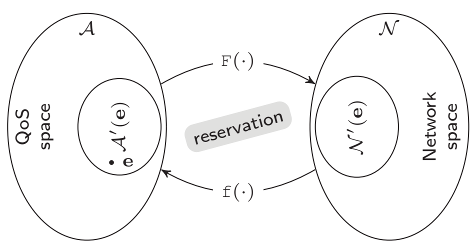


Figure 2.9: Reservation to solve the QoS problem. QoS space represents both, the QoS requirements and the QoS delivery.

Note that, a non empty $\mathtt{F}(\cdot)$ still does not guarantee a solution to our QoS problem. The space, $\mathcal{N}$ is real-valued, so that each ncp in it may be either discrete or non-discrete. An ncp is considered discrete if it is P-tuple conformant, thus contains all discrete values — all remaining ncps are considered non-discrete. It is quite unlikely that $\mathtt{F}(\cdot)$ relates to a discrete ncp, representing some P-tuple. Due to this probable causality, viz., independent QoS requirements may result in a non-discrete ncp, finding an exact solution does not suffice. A remedy for this causality is to shift the exact solution to some discrete ncp. The shift brings in the following problems:

- In order to evaluate the QoS delivered by the discrete ncp, a reverse mapping, $\mathtt{f}(\cdot)$, is required,

$$\mathtt{f} : \mathcal{N} \to \mathcal{A}, \tag{2.12}$$

- The discrete ncp may no longer support the original QoS requirements;
- In case of multiple candidates for discrete ncp, a selection criteria have to be set to determine the best discrete ncp.

These problems are tackled in this thesis as follows.

We set off by determining $\mathtt{f}(\cdot)$, which is used to find the QoS properties repeatedly to build a lookup table. This process effectively discretizes $\mathtt{F}(\cdot)$ by filtering out non-discrete ncps. In the context of NoP, $\mathtt{f}(\cdot)$ represents a function from a vector of P-tuple, $\mathbf{p}$, representing an ncp, to a vector of Q-tuple, $\mathbf{d}$, representing the delivered QoS. Hence,

$$\mathbf{d} = \mathtt{f}(\mathbf{p})$$

means that $[d_1, d_2, \cdots, d_N]$ is the delivered QoS corresponding to the discrete system settings $[p_1, p_2, \cdots, p_N]$. Accordingly, a given vector of Q-tuple, $\mathbf{e} \in \mathcal{A}$, has an image set of vector(s) of P-tuple, $\mathbf{p} \in \mathcal{N}$:

$$\mathbf{p} \in \mathtt{F}(\mathbf{e}) \iff \mathtt{f}(\mathbf{p})\,\underline{\mathtt{sat}}\,\mathbf{e}.$$

The mechanism to establish a certain QoS for all applications in the system, is to set a P-tuple of the corresponding network processes. Configuring the network processes according to the P-tuple leads to a vector of Q-tuple, $\mathbf{d} \in \mathcal{A}$, representing delivered QoS that satisfies all requirements. Due to multiple QoS parameters in a P-tuple, codomain of $\mathtt{F}(\cdot)$ has increasingly high dimensions even for small to medium size networks. Therefore, solving $\mathtt{F}(\cdot)$ is complex and may be done numerically.

The suitability of a discrete ncp to solve some QoS problem is ascertained by its QoS properties. A discrete ncp is a solution if it delivers a QoS that fulfills the `sat`. A set of all solutions is

$$\mathcal{N} \supseteq \mathcal{N}'(\mathbf{e}) \triangleq \{\mathbf{p} \in \mathcal{N} \mid \mathtt{f}(\mathbf{p})\,\underline{\mathtt{sat}}\,\mathbf{e}\}\,. \tag{2.13}$$

A restriction[3] of $\mathtt{f}(\mathbf{p})$ to $\mathcal{N}'(\mathbf{e})$ generates a subspace in $\mathcal{A}$, $\mathcal{A}'(\mathbf{e})$, termed as solution region, with centroid $\mathbf{e}$,

$$\mathtt{f}\mid_{\mathcal{N}'(\mathbf{e})}(\mathbf{p}) \to \mathcal{A}'(\mathbf{e})\,. \tag{2.14}$$

By definition,

$$\mathcal{A} \supseteq \mathcal{A}'(\mathbf{e}) \triangleq \{\mathtt{f}(\mathbf{p}) \in \mathcal{A} \mid \mathtt{f}(\mathbf{p})\,\underline{\mathtt{sat}}\,\mathbf{e}\}\,. \tag{2.15}$$

A characteristic of the space $\mathcal{A}$ is its sparseness. A thinly scattered space is preferred for precision. Sparsity strongly depends on the network size and choices of P-tuple, which determine the sensitivity of QoS properties. Conclusively, the solution region, $\mathcal{N}'(\mathbf{e})$, holds solutions for given QoS requirements of applications, $\mathbf{e}$. Searching for the best solution will be addressed in Chapter 4.

[3] Given a function, $\mathtt{g} : \mathcal{Y} \to \mathcal{Z}$, its restriction, $\mathcal{Y}' \subseteq \mathcal{Y}$, and range restriction, $\mathcal{Z}' \subseteq \mathcal{Z}$, are denoted as $\mathtt{g}\mid_{\mathcal{Y}'}(\cdot)$ and $\mathtt{g}\mid^{\mathcal{Z}'}(\cdot)$, respectively.

2.5.2.2 Network-aware applications

In this approach, the behavior of an application is adapted to network availability. As explained in [24], there are two challenges to address: network awareness, to know the availability of network, and application's adaptation. QoS management is carried out at the application layer, which generally aims to achieve some application related objective — for instance, maximization of user perceived quality.

This concludes the description of two approaches. Some merits and demerits of both approaches are listed below:

- Application-aware networks gear towards service oriented networking. The development of applications can be carried out independently and agnostic to limitations of networks. It also supports a paradigm shift from close to open networks, thus an enabler for intelligent networks. On the downside, applications expose themselves to networks.

- The gap between network behavior over time and requirements of applications, is better filled by network-aware applications. On the downside, a non-incentive cost model promotes aggressiveness in misbehaving applications that may dramatically ruin the QoS of other applications.

2.5.2.3 Our approach

We consider the first approach more viable because of the following reasons:

- This approach keeps a clean separation between the functions of application and network layers, so that technologies on both layers can be developed independently. In context of our QoS problem in Section 2.5, it implies maintaining a semantic separation between two orthogonal problems — how to specify the QoS requirements of applications and how to enforce reservations at the network;

- It promotes network intelligence, hence results in enhanced network architectures. This implies advanced network capabilities such as those supporting extra-functional requirements.

We, therefore, select it to address the QoS problem by building a lookup table. Determining $\mathtt{f}(\cdot)$ is the key problem and is the focus of this thesis. Based on its solution, Pareto curves can be used to represent the solution of $\mathtt{F}(\cdot)$.

In comparison to our approach, the recommended setting in the IEEE standard [92] are incapable to address our QoS problem. There, a single P-tuple is defined, representing four QoS classes. Each class then, contains zero or more applications. The incapability is due to following incompleteness:

- The mapping $\mathtt{F}(\cdot)$ is unknown. In the context of Figure 2.9, the pre-image of the recommended P-tuple in the IEEE standard [92] can not be determined. Hence, the number of applications and their QoS requirements that can be satisfied by the recommended P-tuple, are unknown. Consequently, for some desired QoS requirements of applications, $\mathbf{e}'$, the recommended P-tuple may well lie outside the solution region $\mathcal{N}'(\mathbf{e}')$;

- The mapping $\mathtt{f}(\cdot)$ is unknown. In context of Figure 2.9, the image of the recommended P-tuple in the IEEE standard [92] can not be determined.

In essence, a downside of the given recommendations is that a solution region $\mathcal{N}'(\mathbf{e})$ is not sought for some given QoS requirements of applications, $\mathbf{e}$ — instead, a resulting QoS is observed, which may or may not offer a solution. On the upside, different possibilities can be tabulated by varying only the network size and application profiles.

The key problem is divided into a number of subproblems, defined as range restrictions of $\mathtt{f}(\mathbf{p})$ to $\mathcal{A}'(\mathbf{e})$. For each subproblem, we define a corresponding vector-valued partial function, $\mathtt{f}^{\xi}(\cdot)$, $\xi \in \Xi$ as

$$\mathtt{f}^{\xi}_{k}(\mathbf{p}) = [\mathtt{f}(\mathbf{p})]_k \, .\xi, k \in \mathcal{K} \; . \tag{2.16}$$

For instance, $\mathtt{f}^{\theta}(\mathbf{p})$ represents a vector of shares, which is denoted by $\boldsymbol{\theta}$. Similar vector notations are used for other QoS dimensions. These vectors project partial behaviors, which are determined by their respective partial functions — the latter are derived in the next chapters.

In a given system of some selected MPDU sizes of applications and their MAC data rates, $\mathtt{f}(\cdot)$ maps P-tuple vector to a unique Q-tuple vector, so that all QoS dimensions are determined simultaneously. It is not possible to vary a QoS dimension, without affecting other QoS dimensions. For instance, a vector of some QoS dimension, $\boldsymbol{\varsigma}, \varsigma \in \Xi$, has associated vectors, $\boldsymbol{\xi}_{\boldsymbol{\varsigma}}, \xi \in \Xi \setminus \{\varsigma\}$. When $\boldsymbol{\varsigma}$ is varied by changing, for instance, $\mathbf{p}$, all associated vectors are affected as well: $\boldsymbol{\varsigma}' \neq \boldsymbol{\varsigma}'' \implies \boldsymbol{\xi}_{\boldsymbol{\varsigma}'} \neq \boldsymbol{\xi}_{\boldsymbol{\varsigma}''}$. This means the deduction $\mathbf{p} = \left(\mathtt{f}^{\xi}\right)^{-1}(\boldsymbol{\xi})$ is possible.

In contrast to the above situation, if MPDU sizes of applications and their MAC data rates are not fixed a priori that we consider a relaxation, partial behaviors of the space $\mathcal{A}$ can be generated. In this case, a vector of some QoS dimension, $\boldsymbol{\xi}$, has associated vectors of other QoS dimensions. In particular

$$\begin{aligned} [\boldsymbol{\lambda}_{\boldsymbol{\theta}}]_i &= \mathtt{f}^{\lambda}\left(\left[\left(\mathtt{f}^{\theta}\right)^{-1}(\boldsymbol{\theta})\right]_i\right), \\ [\boldsymbol{\theta}_{\boldsymbol{\lambda}}]_j &= \mathtt{f}^{\theta}\left(\left[\left(\mathtt{f}^{\lambda}\right)^{-1}(\boldsymbol{\lambda})\right]_j\right), \end{aligned} \tag{2.17}$$

represent vectors of latencies and shares. These expressions describe indirect relations between QoS dimensions. Given the relaxation is taken, it allows freedom in generating desired partial behaviors, thus offering choices.

2.6 Priority semantics

It may be required to prioritize applications. We, therefore, establish an ordering of applications based on their importance. The order is based on a priority value assigned to each application.

Classically priority semantics for resource sharing have either an absolute or a relative interpretation. In the absolute priority case, if the resource capacity allows, then generally the QoS requirements of the priority bearer are fulfilled completely — otherwise the resource requester is not admitted to the system. For instance, an admission control system in the telephone network ensures that QoS requirements of all admitted calls are met. In different variants, the fulfillment can be partial. On a resource shrinkage, if all applications cannot coexist, then the lowest priority bearer is the first to leave. The admission control is then based on priorities assigned to the applications, admitting a set of largest priorities that fit. When priorities are equal, a policy such as first come first serve can be adopted to admit a set of applications. In the relative priority case, the priorities imply weights assigned to the resource requesters. These weights may be derived implicitly using the priority values, or given explicitly. In this case, the proportional importance is kept intact, while bearers of all priorities can coexist, although their QoS requirements might be fulfilled partially. Generally, the goal is to maximize the QoS achieved while keeping the proportions.

Priority semantics are relevant in the context that we present now. We set off by introducing a term, mode that is defined as a system state per QoS schema view (cf. Figure 2.7). The system may undergo a mode change on occurrence of at least one of the following events:

- The set of applications changes due to either addition to or removal of elements from this set;
- The QoS requirements of applications change;
- The channel capacity changes.

These events are handled by the QoS Manager (cf. Figure 2.10), which has the following schema:

- Admission controller – it decides if application(s) can be admitted into the network;
- Priority policy – it can be either absolute or relative;
- Mapper – it searches and selects an appropriate mode in the Table by matching the QoS requirements to the best QoS delivery point;
- Table – it contains the mappings as determined by `f(·)`;
- Priority calculator – it assigns a priority value according to a selected Priority policy.

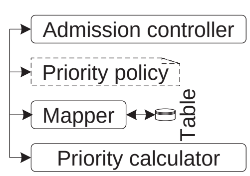


Figure 2.10: A QoS Manager for the schema for the QoS problem.

Upon the occurrence of an event, the QoS Manager estimates a target system state for a potential mode change. State estimation is carried out in accordance with the Priority policy. The QoS Manager sets off by invoking the Mapper to find an appropriate entry in the lookup Table for the target system state. In case of success, a network configuration point for the target system state is returned. When the event is an admittance request(s) for new application(s), the request is evaluated by the Admission controller. In case of failure, prioritization of applications has to be considered to determine a system state that has an entry in the lookup Table. The priority values are computed using the Priority calculator. In an absolute priority case, the prioritization is carried out by considering removal of the lowest priority application(s) from the system and then searching for an entry in the lookup Table. This process continues till success or failure. In a relative priority case, the application requirements may have to be renegotiated in accordance with the weights, which are computed using priority values.

Computing the priority values is done as follows. In general, the priority order of applications can be established by comparing them on a common QoS dimension. When the QoS dimensions vary among different applications, a direct comparison is not possible. Therefore, we dimension this variability onto a weighted average of all QoS dimensions, duly proportioned. Proportionality overcomes the disparity in the QoS dimension scales. The priority value is computed as an inner product of two vectors, $\mathbf{s}_k \cdot \mathbf{g}$, where $\mathbf{s}_k$ is a vector of proportional scores corresponding to the QoS requirements, e_k, and $\mathbf{g}$ is the user assigned weights for the QoS dimensions. Let $\mathtt{ps}(\cdot)$ map e_k to a proportional score, then

$$\mathbf{s}_k = \mathtt{ps}(e_k)\,. \tag{2.18}$$

This proportional score is computed over all applications as

$$\begin{aligned}[]
[\mathtt{ps}(e_k)]_\xi &= \frac{e_k.\xi - \xi_{min}}{\xi_{max} - \xi_{min}}, \xi \in \Xi, \\
\xi_{min} &= \begin{cases} \downarrow_{l \in \mathcal{K}} e_l.\xi & \xi \in \Xi \setminus \{\lambda\} \\ \uparrow_{l \in \mathcal{K}} e_l.\xi & \xi \in \{\lambda\} \end{cases}, \\
\xi_{max} &= \begin{cases} \uparrow_{l \in \mathcal{K}} e_l.\xi & \xi \in \Xi \setminus \{\lambda\} \\ \downarrow_{l \in \mathcal{K}} e_l.\xi & \xi \in \{\lambda\} \end{cases}.
\end{aligned} \tag{2.19}$$

Now we define that application i has higher priority than application j, if

$$\begin{gathered} \mathbf{s}_i \cdot \mathbf{g} > \mathbf{s}_j \cdot \mathbf{g}, \\ [\cdot \text{ is an inner product}]\,. \end{gathered} \tag{2.20}$$

Based on these priority semantics, applications are ordered. In the absolute priority case, the largest number of applications from higher to lower end are admitted, subject to presence of a corresponding mode entry in the Table. In the relative priority case, a relative weight of an application, v_k, can be computed from its priority value, as follows:

$$v_k = \frac{\mathbf{s}_k \cdot \mathbf{g}}{\sum\limits_{l \in \mathcal{K}} \mathbf{s}_l \cdot \mathbf{g}}\,. \tag{2.21}$$

These relative weights are used for relatively apportioning the channel capacity. They also allow emergence of two approaches that can be taken, when the network reservations are not possible due to insufficient channel capacity:

- Scale down the QoS requirements of applications, so that the network reservations can be made. Scaling down may be done according to the relative weights;
- Searching for an entry in the Table whose similar relative weights for delivered QoS points, match best with these relative weights.

2.7 Identification and separation of major concerns

The state transition rates in the state transition systems (cf. Figure 2.5) determine system performance for all considered QoS properties, system contention level and channel wastage. The system can be analyzed by assuming equal durations of two major states, retention and collision, which is quite unlikely. This is due to the fact that a retention duration is an aggregation of transmission times of multiple MPDUs, while a collision duration is defined by transmission time of the longest MPDU involved in it. Hence, the fact that performance is affected by durations of these two states, cannot be ignored in the analysis process. Moreover, retention duration cannot be fixed for each class of applications. This is due to the nature of some applications such as streaming services, which transmit variable size data units — hence, equal sized retention durations are infeasible and unrealistic. Therefore, retention duration needs to be treated as application specific.

The assumption that all processes transmit equal size MPDUs in a selected PHY carrier, implies that the collision duration is MPDU size dependent and fixed. We also evaluate the impact of the shortest duration state, contention. The saturation level of the network, primarily a function of the P-tuple and the number of applications in the network, also bounds the average number of idle slots. Hence, the contention duration can also be assumed fixed. This leaves retention as the only independent state whose duration can drastically affect the system performance — we consider it a dominating state. For instance, if the service times during a successful transmission opportunity vary between different applications, then they derive different channel shares. These deductions call to judge the system performance on two functionally independent aspects:

- *Gaining control*: It addresses the probability for a particular process to acquire exclusive control of the channel, which subsequently leads to a successful transmission. Hence, it is quantified in terms of the transmission priority of the process, while it corresponds to the first type of cycles (cf. Section 2.4.2);
- *Retaining control*: The duration of a successful transmission.

The deductions enable establishing the following:

- The transition probabilities between $\underset{\text{mac}}{\mathcal{S}}$ states in (2.4) are not affected by their durations;
- The stochastic behavior of random access protocols such as DCF [91] and EDCA [92] is only regulated by their transmission frequencies. In the stochastic sense, DCF/EDCA provide equal/unequal transmission frequencies to the processes. For a system of processes tuned to certain transmission frequencies, the channel share changes as a function of *retaining control*. In EDCA, $TXOPLimit$ is used to characterize and quantify this aspect.

While the independence between these two functional aspects is exploited in the model development, we also emphasize that the system performance can not be quantified if one aspect is ignored. For instance, quantifying *gaining control* does not translate directly into channel shares as evident from the next trivial example. Consider a transmission system of two processes, which are uniformly scheduled stochastically. Assume that the second process transmits x times more data than the first one, then their shares are given by $\left(\frac{1}{1+x}, \frac{x}{1+x}\right)$ respectively. The process shares are proportional to the service times when equal transmission opportunities are available. Hence, it implies that in order to determine the shares, there is also a need to incorporate service times on top of transmission opportunities. We express *retaining control* in terms of $TXOPLimit$ transmission transactions.

Definition 2.7.1. A transmission transaction starts with the transmission of first bit of an MPDU and ends with the reception of its ACK. □

During multiple transmission transaction, a gap of $SIFS$ enables to keep control over the channel. This thesis will present the quantified system performance separately for both aspects.

2.8 Conclusions

This chapter raises the basic question of building a system conceptual model. Addressing this question has resulted in our proposed performance framework. We postulate fundamental building blocks for defining and analyzing this performance framework. There are two subquestions that are addressed as follows.

Subquestion a. highlights the need for an appropriate abstraction of the EDCA behavior that supports performance derivation. EDCA has intricate complexities of the invocation sequence. We aim to bind its underlying functionality into a unified view that is used for development of quantitative analysis in the following chapters. For this purpose, its behavior is generalized to a simplified algorithm and a State Transition Diagram that we propose. This unified view is identified as the system cyclic behavior that is expressed as cycles. These cycles transcend specific MAC arbitration mechanisms and their settings, which have different impacts on various QoS concerns (channel access frequency, channel share, MAC latency and MAC reliability). Investigations based on such an approach lead to performance derivation for achieving a distributed control of the network, in the trace framework (research **Question 2**) and stochastic models (research **Question 3** and research **Question 4**).

A distributed networked system can be modeled by identifying and representing the states and their transitions either globally at the network level, or locally at the process level. We take the latter approach by creating a conceptual partition of the network into two blocks, to model the cooperation and competition in the network, per cycle.

Cycle based performance analysis of systems allows extension intuitively. In our case, the definition of cycles can be extended to consider consecutive transmissions that produce MAC capture effect due to extreme asymmetric settings [201].

The term QoS refers to the desired behavior of a service. Generally, it has a loose meaning, unless defined explicitly. Subquestion b. indicates the need for formal specifiability of the QoS requirements of applications to exactly determine their fulfillment, support manageability and service auditing. To this end, we propose QoS semantics, which are independent of specific application types. These types may represent applications with distinguished QoS concerns. The fulfillment of QoS requirements is coupled with management of channel, as controlled by network configurations. To judge this fulfillment aspect quantitatively, we propose a satisfiability relation. This relation is based on all QoS concerns considered in this thesis. For each QoS concern, a model needs to be developed. Each model represents a partial behavior that is mapped to by corresponding partial function. In case fulfillment is not possible due to requirements exceeding the network capacity, there is a need to deny admission access to some of the applications. For this purpose, we establish an order of their criticality using our proposed priority semantics.

An important finding of study of this subquestion is regarding abuse of the term, priority, in context of existing related literature. Therein, priority is taken for preferential treatment, without explicit relation to network configurations. In contrast, we consider priority assignment and priority implementation as two separate issues. They are treated in context of criticality of applications and settings of network configurations, as provided by the technology, respectively. A network configuration determines some network state, hence preferential order of applications is not even required to achieve a distributed control of the network.

Originally, the QoS problem needs a transformation from the QoS space to the network configuration space. Due to technology limitations, not all points in the transformation range are reachable. A method proposing a shift to a reachable point may open more issues related to side effects of the shift. These include no solution and solution multiplicity. In our approach, we avoid this situation by adopting a reverse mapping that considers only reachable points.

All QoS concerns considered together restrict the solution space. By taking departure from some of them, termed as relaxation in our context, some interesting network behaviors are expected. They may support one-to-many relationships between QoS properties. For instance, two solutions can have the same share but different latency values or vice versa. Allowing these behaviors may bring in more scheduling choices.

The proposed conceptual model clearly reveals that quantifying only the transition rate between contention and transmission is not enough to represent all system aspects. Also transmission duration has a major influence on the system performance. This consideration leverages freedom that is required for addressing different application types. Hence, two separate system aspects, namely *gaining control* and *retaining control* are introduced, which offer the required independence and flexibility. Our model considers the influence of both system aspects in a unified manner, and exploits their mutual independence for performance quantification in later chapters. In the next chapter, system behavior is traced formally by considering these two aspects.

3

Tracking system evolution: trace analysis and performance

In simulation or an actual system execution, autonomous decisions taken by the communicating processes and the events triggered thereafter, represent the dynamic behavior of the EDCA network. Although a history of system behaviors can be kept in various trace types, this chapter focuses on traces of the EDCA events. Traces are timed sequences of events, where the events are changes in the process states. A trace represents a possible history. We distinguish between a trace that keeps the entire history of an EDCA network from a subtrace that is some restriction of a trace. Subtraces represent projected history recorded in isolation. Subtraces defined in the same time interval can be combined to form global effects.

The usage of traces in computer networks in general and WLANs in particular, is limited to obtaining performance numbers either to verify mathematical models or for obtaining results only. To this end, no structured approach is adopted. Different tracing tools produce their own specific trace formats. We note that no standard is defined. The existing literature on performance modelling and service/class differentiation of CSMA/CA based protocols lacks a formal framework of event traces. This framework, herein referred to as the trace framework, would allow specification of process and MAC event cycles. Cyclic behavior in the trace may represent fluctuations in the QoS properties. The study of such fluctuations is significant for isolating peculiar behaviors as well as for optimal resource allocation. In the existing work, a formal aggregation method to obtain the MAC system states from states of individual processes, is absent — as will be discussed in Section 3.1, the related work. These two gaps lead to the following research question, as posed in Chapter 1:

Question 2. How to track the EDCA network evolution over time?

We identify the following four subquestions concerning the event traces within the scope of the main question:

a. How to discover MAC states resulting from arbitrary subtraces of processes?

b. How to derive low-level performance metrics such as occurrence probabilities of process states, from an arbitrary subtrace of a process?
   This problem is motivated by the observation that low-level details reflect on system properties more accurately. They have the capability to detect anomalies that may

go unseen by high-level details. Low-level metrics sought can also be used to derive high-level performance metrics.

c. How to identify and specify process-specific and MAC event-specific cyclic behaviors in traces?
This subquestion is motivated by the limitation that the dynamic system cycles defined in Section 2.4.2 are transparent over processes and MAC events.

d. How to examine QoS properties for both, short-term and long-term durations, from an arbitrary subtrace of a process?

In order to address the main question, this chapter introduces theory to formally define a trace framework for the EDCA network. We capture the temporal evolution of internal states of different processes. To do so, we realize that a state change recording by events is enough to solve the subquestions. For each subquestion, following contributions are made, respectively:

a. A formal aggregation method to obtain the MAC system states from individual processes states.

b. A low-level performance metric, backoff process state probability, is derived, which offers deeper insights into process behaviors such as those resulting from the MAC parameter settings.

c. Identification and formal specification of process-specific and MAC event-specific cycles in traces.

d. The presence of QoS properties in an arbitrary subtrace is detected formally for both, short-term fluctuations and long-term expected values. This derivation also relates the actual execution of the system to an analytical model. This contribution is motivated by a famous quotation from Prof. Edsger W. Dijkstra:

> ‘Program testing can be used to show the presence of bugs, but never to show their absence!’ [29];

The trace analysis in this chapter fills the gap between execution traces and analytical model. More specifically, we highlight events which trigger internal state transitions of processes and study their impact at the process/network levels. These state transitions influence performance of individual processes and reflect interesting patterns about their behavior over time. This is achieved by considering time variant properties of events. Formation of our trace framework leverages the study needed for designing our custom made discrete event simulator for EDCA and processing of traces from experiments — taken up in the next chapters. In the next section, the prior approaches to trace analysis in WLAN systems are reviewed.

3.1 Related work

Trace analysis in computer networks has been predominantly applied in the context of security for keeping the privacy of information in traces [138, 45, 239, 28, 46, 56, 57, 155, 73, 189]. Trace analysis has also been extensively used for model-based conformance testing of protocols against specifications [23, 15, 118] and fingerprinting [185, 104, 119, 70, 237]. Traces also allow realistic evaluation of network protocols [83, 143, 156]. In the context of performance issues, trace analysis gives insights for troubleshooting and determining bottlenecks in networks [227, 43]. There is also an appealing work on trace analysis of discrete-event systems to reduce repetitive behavior for isolating irregularities and to detect cycles for studying their properties [110]. Trace collection is a necessary first step in creating realistic models crucial to designing, simulating, and evaluating network protocols [2-4].

Under our assessment of prior work as stated, we are not aware of any work that focuses on the understanding of the EDCA dynamics through its trace analysis. Formalism on trace analysis for performance of EDCA is absent from existing literature.

To the best of our knowledge, the work presented in this chapter departs from other works as it presents a formal trace framework for the EDCA network. Through the contributions to address respective technical problems as identified within the scope of the trace framework, an exact blueprint of the EDCA dynamics is expressed. This includes unfolding the EDCA behavior over time, through a query system that is based on performance cycles, which reflect fluctuations. The captured details lead directly to an analytical model.

3.2 A tracer

We extract a process behavior by studying its trace properties. A trace long enough may reflect the stable behavior of a process. For long enough traces, *bc* drawings are uniformly distributed, according to the law of large numbers (LLN) [14]. Similarly, fluctuating *bc* count down rates over short-term time intervals are smoothed down. Thus, an average impact of network configuration over time is ensured for studying the opposition as offered by the competing network against the reference process.

The trace properties depend on process internal states over time, termed as istates, which arise due to values of internal variables, rc and bc. We differentiate istates from the abstract process states, contention and transmission, as defined in (2.4). By (2.7), these two abstract states collectively form a set of MAC states of the NoP. These MAC states will be derived from our process specific subtraces. Tracking system evolution requires some basic definitions that are presented next.

Definition 3.2.1. A trace, ϕ, is a sequence of events with attributes,

$$\phi = \psi_0\psi_1\psi_2\cdots,$$

where the i^{th} event in the trace is denoted by an anchor, ψ_i, while $\texttt{next}(\psi_i) = \psi_{i+1}$ and $\texttt{prev}(\psi_i) = \psi_{i-1}$ are partial functions. The attributes of event, ψ, are specified by a tuple, (k, r, b, t), with elements defined as follows:

- $\psi.k$: process id;
- $\psi.r$: rc value;
- $\psi.b$: bc value;
- $\psi.t$: event occurrence time.

The empty trace is denoted by ϵ. □

Definition 3.2.2. The duration of a trace ϕ, $\mathtt{d}(\phi)$, is the sum of all istate intervals within ϕ. Formally

$$\mathtt{d}(\psi\phi) = \begin{cases} 0 & \phi = \epsilon \\ (\mathtt{next}(\psi).t - \psi.t) + \mathtt{d}(\phi) & \phi \neq \epsilon \end{cases}. \tag{3.1}$$

□

Definition 3.2.3. The length of a trace ϕ, $\mathtt{g}(\phi)$, is the number of elements in ϕ. Formally

$$\begin{aligned} \mathtt{g}(\epsilon) &= 0, \\ \mathtt{g}(\psi\phi) &= 1 + \mathtt{g}(\phi). \end{aligned} \tag{3.2}$$

□

A subtrace is a projection of trace that is restricted over one or more attributes.

Definition 3.2.4. Given a predicate on anchors, ϱ, the restriction, $\mathcal{R}(\phi, \varrho)$, is the trace obtained from ϕ leaving out anchors not satisfying ϱ. Formally

$$\begin{aligned} \mathcal{R}(\epsilon, \varrho) &= \epsilon, \\ \mathcal{R}(\psi\phi, \varrho) &= \begin{cases} \mathcal{R}(\phi, \varrho) & \neg\varrho(\psi) \\ \psi\mathcal{R}(\phi, \varrho) & \varrho(\psi) \end{cases}. \end{aligned} \tag{3.3}$$

□

We introduce some definitions to harvest this notation.

- $\mathcal{R}\mathtt{k}_k(\phi) \triangleq \mathcal{R}(\phi, \varrho(\psi) : \psi.k = k)$,
  (the restriction of ϕ to anchors of process k);
- $\mathcal{R}\mathtt{b}_b(\phi) \triangleq \mathcal{R}(\phi, \varrho(\psi) : \psi.b = b)$,
  (the restriction of ϕ to anchors with b value equal to b);
- $\mathcal{R}\mathtt{r}_r(\phi) \triangleq \mathcal{R}(\phi, \varrho(\psi) : \psi.r = r)$,
  (the restriction of ϕ to anchors with r value equal to r);
- $\mathcal{R}\mathtt{rb}_{r,b}(\phi) \triangleq \mathcal{R}(\phi, \varrho(\psi) : \psi.r = r \wedge \psi.b = b)$,
  (the restriction of ϕ to anchors with istate (r, b));
- $\mathcal{R}\mathtt{int}_{[t_1,t_2]}(\phi) \triangleq \mathcal{R}(\phi, \varrho(\psi) : t_1 \leqslant \psi.t \leqslant t_2)$,
  (the restriction of ϕ to anchors within an interval $[t_1, t_2]$);
- $\mathcal{R}\mathtt{t}_t(\phi) \triangleq \mathcal{R}(\phi, \varrho(\psi) : \psi.t \leqslant t < \mathtt{next}(\psi).t \vee \mathtt{prev}(\psi).t \leqslant t < \psi.t)$,
  (the restriction of ϕ to two adjoining neighboring anchors to time t — either of the neighbor may non-exist if t is beyond the trace ends).

Let $\phi_k = \mathcal{R}\mathtt{k}_k(\phi)$ denote subtrace of process k. Obviously more restricted subtraces contain fewer anchors than less restricted ones. For instance, $\phi_{k,r,b}^{[t_1,t_2)} \triangleq \mathcal{R}\mathtt{int}_{[t_1,t_2)}(\mathcal{R}\mathtt{rb}_{r,b}(\phi_k))$ is more restricted than $\phi_{k,b}^{[t_1,t_2)} \triangleq \mathcal{R}\mathtt{int}_{[t_1,t_2)}(\mathcal{R}\mathtt{b}_b(\phi_k))$. The latter is ignorant of rc. These two subtraces span over the same interval, with a surjective function mapping the latter to the former, $\mathtt{f} : \phi_{k,b}^{[t_1,t_2)} \twoheadrightarrow \phi_{k,r,b}^{[t_1,t_2)}$. r is bounded as follows:

- for $\mathtt{cw}(r)$ as an exponentially increasing function of r,

$$r' < r'' \implies \mathtt{cw}_k\left(r'\right) < \mathtt{cw}_k\left(r''\right),$$

$$\lceil \hat{r} \rceil \leqslant r', r'' \leqslant RetryLimit_q,$$

$$\hat{r} = \log_d \left(\frac{b+1}{CWmin+1} \right),$$

where base d is the persistence factor for CW change;

- for $\mathtt{cw}(r)$ as an exponentially decreasing function of r,

$$r' < r'' \implies \mathtt{cw}_k\left(r'\right) > \mathtt{cw}_k\left(r''\right),$$

$$0 \leqslant r \leqslant \lfloor \hat{r} \rfloor .$$

As a general property of sequences, we assume that an indexing notation is available. Let $\mathtt{y} : \mathcal{I}_{\phi'} \to \mathcal{I}_{\phi}$ be a mapping from an index set of a subtrace, $\mathcal{I}_{\phi'}$, to an index set of a trace, $\mathcal{I}_{\phi}$. For instance, $\langle \phi_k \rangle_{i \in \mathcal{I}_{\phi'}} = \langle \phi \rangle_{\mathtt{y}(i) \in \mathcal{I}_{\phi}}$.

The process istate does not change between its two consecutive anchors, i.e. during $\left[\psi_{\mathtt{z}(i)}.t, \psi_{\mathtt{z}(i+1)}.t\right)$. The duration of this interval is determined at the system level. This duration implies an $aSlotTime$ during a contention, while it is variable during a retention and a collision — for the latter, it depends on the size of the largest MPDU involved. The frequency of anchors decreases with r. Generally an i^{th} anchor of subtrace $\phi^{[t_1,t_2)}_{k,r\neq0,b\neq0}$ signifies lower bounds on two aspects:

- at least i MPDUs have assumed HoL position in the process MAC queue,

$$\mathtt{g}\left(\phi^{[t_1,t_2)}_{k,r\neq0,b\neq0}\right) \leqslant \mathtt{g}\left(\phi^{[t_1,t_2)}_{k,0,0}\right).$$

  Specifically an i^{th} anchor of the subtrace, $\phi_{k,0,0}$, indicates that exactly i MPDUs have assumed HoL position. Hence, at any point in time, the number of anchors in $\phi_{k,0,0}$ exceeds the number of anchors in $\phi_{k,r,b}, b \neq 0, r \neq 0$. Intuitively, this property reflects a behaviorial property of a process that for some MPDUs, it may not reach some bc and rc values. This is a key observation that must be considered for quantifying process performance.

- at least i MPDUs have been transmitted by the process.

3.3 MAC states from subtraces

Using the process subtraces, we identify the MAC states using two approaches. The first approach is based on time series analysis of istate attributes, while the second one is based on istate evolution rate over time.

3.3.1 istate attributes

In this approach, we identify the MAC states using the istate attributes for a set of processes. The considered MAC states are: contention, retention and collision. Let $\mathtt{e}_{\mathcal{P}}(t)$ map processes in $\mathcal{P}$ to a MAC state at time, t.

$$\mathtt{e}_{\mathcal{P}}(t) = \begin{cases} \textsf{contention} & \mathtt{z}_{\mathcal{P}}(t) = 0 \\ \textsf{retention} & \mathtt{z}_{\mathcal{P}}(t) = 1 \\ \textsf{collision} & \mathtt{z}_{\mathcal{P}}(t) > 1 \end{cases},$$
$$\mathtt{z}_{\mathcal{P}}(t) = \left|\{k \in \mathcal{P} \mid \mathcal{R}\mathtt{t}_t(\phi_k)\,.b = 0\}\right|.$$

Counting the number of processes in transmission istate defines the MAC state at any point in time. For an istate, this mapping has to be extended for the duration of the istate. The extended mapping determines MAC effects with durations. Currently the MAC effect collision incorporates MPDU dropout case, which is now considered separately. The set of considered MAC states, hereby referred to as MAC effects is,

$$\mathcal{M} \triangleq \left\{m^{\mathrm{c}}, m^{\mathrm{r}}, m^{\mathrm{l}}, m^{\mathrm{d}}\right\} \cup \{\diamond\},$$

where

$m^{\mathrm{c}} \triangleq$ a contention,

$m^{\mathrm{r}} \triangleq$ a retention,

$m^{\mathrm{l}} \triangleq$ a collision resulting in next CW,

$m^{\mathrm{d}} \triangleq$ a collision resulting in a dropout,

$\diamond \triangleq$ a null marker to cater for an empty trace, ϵ .

Given a trace ϕ, $\mathtt{E}_{\mathcal{P}}(\psi)$ maps ψ into a corresponding sequence of MAC effects in $\mathcal{M}$, in the interval of the istate.

$$\mathtt{E}_{\mathcal{P}}(\epsilon) = \diamond,$$
$$\mathtt{E}_{\mathcal{P}}(\psi) = \begin{cases} m^{\mathrm{c}} & \underset{\psi.t\leqslant t<\mathtt{next}(\psi).t}{\forall} (\mathtt{e}_{\mathcal{P}}(t) = \textsf{contention}) \\ m^{\mathrm{r}} & \underset{\psi.t\leqslant t<\mathtt{next}(\psi).t}{\forall} (\mathtt{e}_{\mathcal{P}}(t) = \textsf{retention}) \\ m^{\mathrm{l}} & \underset{\psi.t\leqslant t<\mathtt{next}(\psi).t}{\forall} (\mathtt{e}_{\mathcal{P}}(t) = \textsf{collision}) \wedge (\psi.r < \mathit{RetryLimit}) \\ m^{\mathrm{d}} & \underset{\psi.t\leqslant t<\mathtt{next}(\psi).t}{\forall} (\mathtt{e}_{\mathcal{P}}(t) = \textsf{collision}) \wedge (\psi.r = \mathit{RetryLimit}) \end{cases}. \tag{3.4}$$

Asserting on anchors properties during istate interval leverages identifying the MAC state as follows.

The states in the domain of $\mathtt{Map}(\cdot)$ in (2.7) define the MAC states in terms of the reference process, $k \in \mathcal{K}$, and the competing network, $\eta = \mathcal{K} \setminus \{k\}$. The MAC effects due to these two blocks at ψ can be evaluated by $\mathtt{E}_{\mathcal{P}}(\psi)$. The combined MAC effects determine the system MAC states, as follow.

0. (contention, contention): $\left(\mathtt{E}_{\{k\}}(\psi) = m^{\mathrm{c}}\right) \wedge \left(\mathtt{E}_{\eta}(\psi) = m^{\mathrm{c}}\right)$. By the $\mathtt{Map}(\cdot)$, the image is contention, which also directly corresponds to the predicate $\mathtt{E}_{\mathcal{K}}(\psi) = m^{\mathrm{c}}$.

1. (contention, transmission): $\left(\mathtt{E}_{\{k\}}(\psi) = m^{\mathrm{c}}\right) \wedge (\mathtt{E}_{\eta}(\psi) \in \mathcal{M} \setminus \{m^{\mathrm{c}}\})$. By the $\mathtt{Map}(\cdot)$, the image is transmission, which also directly corresponds to the predicate $\mathtt{E}_{\mathcal{K}}(\psi) \in \mathcal{M} \setminus \{m^{\mathrm{c}}\}$.

2. (transmission, contention): $\left(\mathtt{E}_{\{k\}}(\psi) = m^{\mathrm{r}}\right) \wedge (\mathtt{E}_{\eta}(\psi) = m^{\mathrm{c}})$. By the $\mathtt{Map}(\cdot)$, the image is retention, which also directly corresponds to the predicate $\mathtt{E}_{\mathcal{K}}(\psi) = m^{\mathrm{r}}$.

3. (transmission, transmission): $\left(\mathtt{E}_{\{k\}}(\psi) = m^{\mathrm{r}}\right) \wedge (\mathtt{E}_{\eta}(\psi) \in \mathcal{M} \setminus \{m^{\mathrm{c}}\})$. By the $\mathtt{Map}(\cdot)$, the image is transmission, which also directly corresponds to the predicate $\mathtt{E}_{\mathcal{K}}(\psi) \in \mathcal{M} \setminus \{m^{\mathrm{c}}\}$.

3.3.2 istate transition rate

The transition rate of a reference process, $\mathtt{tr}(\phi_k)$, is defined as the number of istate changes per time unit. It is directly given as a ratio of subtrace length to subtrace duration,

$$\mathtt{tr}(\phi_k) = \frac{\mathtt{g}(\phi_k)}{\mathtt{d}(\phi_k)}, \text{ refers (3.2) and (3.1).}$$

This transition rate is influenced by transition rates of other processes in the referred network. In this approach, we exploit this property to identify the MAC states. To this end, we use an observation that during system contention, all processes have a uniform transition rate due to fixed size of PHY $aSlotTime$. Let us call it a globally balanced rate (gbr). During gbr, there is one istate change per $aSlotTime$:

$$\mathtt{tr}(\phi_k) = \frac{1}{aSlotTime}.$$

We are interested in checking the MAC state during a trace interval $[t_1, t_2)$. Let $\phi^{[t_1,t_2)}$ denote $\mathcal{R}\mathtt{int}_{[t_1,t_2)}(\phi_k)$, then a sufficient condition for contention during $[t_1, t_2)$ is that gbr holds during this interval:

$$\mathtt{tr}\left(\phi^{[t_1,t_2)}\right) = \frac{1}{aSlotTime}.$$

In other words, all processes transit in $aSlotTime$ duration steps during contention. This predicate detects only system contention state when all process have already completed their respective $AIFS$ deferrals. During other MAC states, transition rates are not uniform across the board and are less than gbr.

Now that we have generated MAC states from the process subtraces, this concludes addressing subquestion a., as posed in the introduction of this chapter.

3.4 Low-level performance metrics

In this section, subquestion b. is addressed by presenting process istate distributions, which emerge in two spaces. When the processes are traced long enough in these spaces, they attain average behavior.

Bidimensional space

The istate probability, $\pi_{k,r,b}$, over (r,b) space is defined as the occurrence count ratio of istate (r,b) to all istates. Both terms are offered by length of a (sub)trace. Let $\phi_{k,r,b}$ denote $\mathcal{R}\mathtt{rb}_{r,b}(\phi_k)$, then

$$\pi_{k,r,b} = \frac{\mathtt{g}(\phi_{k,r,b})}{\mathtt{g}(\phi_k)}. \tag{3.5}$$

Unidimensional space

We marginalize out r from (3.5) to derive the unidimensional istate distribution. Let $\psi_{k,b}$ denote $\mathcal{R}\mathtt{b}_b(\phi_k)$, then

$$\pi_{k,b} = \frac{\mathtt{g}(\psi_{k,b})}{\mathtt{g}(\psi_k)}. \tag{3.6}$$

3.5 Process and MAC event specific cycles

Subquestion c. is taken up in this section. A recurrent execution of the MAC state machine in Figure 2.5 generates interleaved dynamic system cycles of two types defined in Section 2.4.2. These cycles have the following limitations:

a. They are system scoped, i.e., the cycles due to individual processes are non-separable. As a consequence of this system scoped abstraction, the cycle numbers are not process specific.

b. They do not offer MAC event specific occurrence counting;

c. They do not discriminate between those collisions that result in an MPDU dropout and those that do not.

Due to these limitations, process and MAC event specific fluctuations can not be projected. For instance, consider the number of MPDU dropouts inside a MAC retention cycle, which is a process subtrace between two consecutive retention opportunities. The variation in this count over time, indicates short-term fluctuating behavior of the process — hence, it has a key relevance to identify abnormal behaviors. We aim to offer a formal tool that can project these types of fluctuations in MAC event over time. For this purpose we define process and MAC event specific cycles.

Definition 3.5.1. A partition of a trace, ϕ, into mutually exclusive and collectively exhaustive subtraces defines cycles that are process, k, and MAC event, $m \in \mathcal{M}$ specific, such that the $i^{\text{th}}, (i > 0)$ cycle contains the i^{th} occurrence of m for k. It is formally represented by a time interval,

$$\mathcal{O}_k^{(i)}(\phi, m) \triangleq (t_1, t_2].$$

This time interval is determined by identifying anchors at the cycle boundaries. For a trace, ϕ, let $\mathtt{A}_k^i(\phi, m)$ denote the anchor corresponding to the i^{th} cycle.

$$\begin{aligned} \mathtt{A}_k^i(\epsilon, m) &= \diamond, \\ \mathtt{A}_k^i(\psi\phi, m) &= \begin{cases} \psi & \mathtt{E}_{\{k\}}(\psi) = m \wedge i = 1 \\ \mathtt{A}_k^{i-1}(\phi, m) & i > 1 \wedge \mathtt{E}_{\{k\}}(\psi) = m \\ \mathtt{A}_k^i(\phi, m) & i \geqslant 1 \wedge \mathtt{E}_{\{k\}}(\psi) \neq m \end{cases}. \end{aligned} \tag{3.7}$$

Now $t_1 = \mathtt{A}_k^{i-1}(\phi, m).t$ and $t_2 = \mathtt{A}_k^i(\phi, m).t$.

□

These process and MAC event specific cycles offer a set of rich features that are applied to judge short-term performance fluctuations — demonstrated as follows. Consider a generic query of the form:

What is a occurrence count of a MAC event, m', within an i^{th} cycle of MAC event, m?

Given that an i^{th} cycle of a MAC event, $m \in \mathcal{M}$, $\mathcal{O}_k^{(i)}(\phi, m)$, lies in an interval, $(t_1, t_2]$, let $\phi_k^{(t_1,t_2]} \triangleq \mathcal{R}\mathtt{int}_{(t_1,t_2]}(\phi_k)$ be the subtrace within this interval. A solution to this query is sought by mapping this subtrace to an occurrence count of the MAC event, m', in it. We introduce a restriction of ϕ to a MAC event, $m \in \mathcal{M}$ anchors of process, k,

$$\mathcal{R}\mathtt{m}_m(\phi) \triangleq \mathcal{R}\left(\phi_k, \varrho(\psi) : \mathtt{E}_{\{k\}}(\psi) = m\right).$$

The query is satisfied as $\mathtt{g}\Big(\mathcal{R}\mathtt{m}_{m'}\Big(\phi_k^{[t_1,t_2)}\Big)\Big)$. Our formal tool is explained by three concrete queries:

1. How many MPDUs are dropped in the i^{th} retention cycle, i.e., between the successful transmission of $(i-1)^{\text{th}}$ and i^{th} MPDUs?

   In this case, $\mathcal{O}_k^{(i)}(\phi, m^{\text{r}})$ is defined over the subtrace, $\phi_k^{(t_1,t_2]}$. Hence, the query response is $\mathtt{g}\Big(\mathcal{R}\mathtt{m}_{m^{\text{d}}}\Big(\phi_k^{(t_1,t_2]}\Big)\Big)$.

2. How many collisions are encountered by the i^{th} HoL MPDU before being successfully transmitted?

   Using the same notations, the query response is $\mathtt{g}\Big(\mathcal{R}\mathtt{m}_{m^{\text{l}}}\Big(\phi_k^{(t_1,t_2]}\Big)\Big)$.

3. How many MPDUs are successfully transmitted between the $(i-1)^{\text{th}}$ and i^{th} dropped MPDUs?

   The query response is $\mathtt{g}\Big(\mathcal{R}\mathtt{m}_{m^{\text{r}}}\Big(\phi_k^{(t_1,t_2]}\Big)\Big)$.

The presented queries can determine short-term fluctuations in QoS properties over time. By aggregating the cyclic behaviors over long traces, averages can also be derived. We conclude this section by stating that a multi-faceted querying system offers a formal interpretation of short-term fluctuations.

3.6 QoS properties

In this section, subquestion d. is addressed. The short-term QoS properties are transient, hence they are defined for duration of process and MAC event cycles. For deriving long-term averages, the interval taken is denoted by $(T_1, T_2]$.

3.6.1 Channel access probability

The channel access probability, ρ, is given as ratio of successful transmissions count for a process, k, to a system transmissions count.

$$\rho_k = \frac{\mathtt{g}(\mathcal{R}\mathtt{m}_{m^{\text{r}}}(\phi_k))}{\mathtt{g}(\mathcal{R}\mathtt{b}_0(\phi))}. \tag{3.8}$$

3.6.2 Channel access frequency

The process accesses the channel irregularly. This phenomenon is already highlighted by process/event specific cycles. The channel access frequency, ν, is defined as the number of accesses made per unit time. We determine it for both short-term and long-term durations.

3.6.2.1 Transient

Given that each success cycle, bears only one successful transmission, we take into account the duration of a success cycle. Given that an i^{th} success cycle, $\mathcal{O}_k^{(i)}(\phi, m^{\text{r}})$, is bounded by an interval $(t_1, t_2]$, a transient channel access frequency is

$$\nu_k^{(i)} = \frac{1}{t_2 - t_1}. \tag{3.9}$$

3.6.2.2 Expected

The expected frequency is given as ratio of total MPDU transmitted to long-term duration.

$$\nu_k = \frac{\mathtt{g}\left(\mathcal{R}\mathtt{m}_{m^{\text{r}}}\left(\phi_k^{(T_1, T_2]}\right)\right)}{T_2 - T_1}. \tag{3.10}$$

3.6.3 Channel share

The channel share, θ, is a channel usage time proportion of a given interval, at some transmission rate.

3.6.3.1 Transient

Given that an i^{th} success cycle spans over $(t_1, t_2]$, the channel usage time during this interval is $\mathtt{d}(\mathcal{R}\mathtt{m}_{m^{\text{r}}}(\phi_k))$, hence

$$\theta_k^{(i)} = \frac{\mathtt{d}(\mathcal{R}\mathtt{m}_{m^{\text{r}}}(\phi_k))}{t_2 - t_1}. \tag{3.11}$$

3.6.3.2 Expected

The expected share is averaged over long-term.

$$\theta_k = \frac{\mathtt{d}\left(\mathcal{R}\mathtt{m}_{m^{\text{r}}}\left(\phi_k^{(T_1, T_2]}\right)\right)}{T_2 - T_1}. \tag{3.12}$$

3.6.4 MAC loss

The MAC loss, ι, is MPDU dropout rate.

3.6.4.1 Transient

The transient losses are determined per success cycle, which is given by query 1 in Section 3.5. Hence,

$$\iota_k^{(i)} = \frac{\mathtt{g}\left(\mathcal{R}\mathtt{m}_{m^{\text{d}}}\left(\phi_k^{(t_1, t_2]}\right)\right)}{t_2 - t_1}. \tag{3.13}$$

3.6.4.2 Expected

Short-term fluctuations in losses are smoothed out in expected MAC losses.

$$\iota_k = \frac{\mathtt{g}\left(\mathcal{R}\mathtt{m}_{m^{\mathrm{d}}}\left(\phi_k^{(T_1,T_2]}\right)\right)}{T_2 - T_2}. \tag{3.14}$$

3.6.5 MAC latency

The MAC latency, λ, is defined as the channel contention delay for an MPDU and its transmission time. The channel contention delay starts as soon as a fresh MPDU assumes a HoL position. This position promotion is done immediately after the previous HoL MPDU is either successfully transmitted or dropped out. A fresh MPDU may meet a dropout fate due to exhaustion of retry attempts. Hence, a trace has both types of lifetimes interleaved — those of dropped and of successful MPDUs. The prior type lifetimes should not be considered for deriving latency of former types lifetimes. Consequently, a success cycle can not be used to model an MPDU latency, as its boundaries may include duration of dropped MPDU(s). Thus, it may add up their lifetime(s) to that of a successful one, which is misleading. This restricts us to derive latency which is based on the lifetime of only successful MPDUs.

3.6.5.1 Transient

The lifetime of an i^{th} successful MPDU starts after the lifetime of $(i-1)^{\text{th}}$ MPDU, an event that is marked by either m^{r} or m^{d}, while it ends at $\mathtt{A}_k^i(\phi, m^{\mathrm{r}}).t$. This event lies within the i^{th} success cycle, $\mathcal{O}_k^{(i)}(\phi, m)$ that spans over $(t_1, t_2]$. By taking $\phi_k^{(t_1,t_2]} = \mathcal{R}\mathtt{int}_{(t_1,t_2]}(\phi_k)$, let $j = \mathtt{g}\left(\mathcal{R}\mathtt{m}_{m^{\mathrm{d}}}\left(\phi_k^{(t_1,t_2]}\right)\right)$ be the occurrence count of m^{d} within $\phi_k^{(t_1,t_2]}$. Now this event occurs at $\mathtt{A}_k^j\left(\phi_k^{(t_1,t_2]}, m^{\mathrm{d}}\right).t$, so that

$$\lambda_k^{(i)} = \mathtt{A}_k^i(\phi, m^{\mathrm{r}}).t - \mathtt{A}_k^j\left(\phi_k^{(t_1,t_2]}, m^{\mathrm{d}}\right).t\ . \tag{3.15}$$

3.6.5.2 Expected

An average of all transient latencies over the entire trace is

$$\lambda_k = \frac{\sum_{i=0}^{s} \lambda_k^{(i)}}{\mathtt{g}\left(\mathcal{R}\mathtt{m}_{m^{\mathrm{r}}}\left(\phi_k^{(T_1,T_2]}\right)\right)}, \tag{3.16}$$

where the denominator term is the occurrence count of success cycles in the trace.

3.6.6 MAC reliability

Due to limited number of retries, each MPDU is prone to a dropout, hence its delivery is not guaranteed. We assess its reliability, ω, as a number of delivered MPDUs normalized by all HoL MPDUs.

3.6.6.1 Transient

The transient reliability is defined over duration of a success cycle.

$$\omega_k^{(i)} = \frac{1}{1 + \mathtt{g}\left(\mathcal{R}\mathtt{m}_{m^{\mathrm{d}}}\left(\phi_k^{(t_1,t_2]}\right)\right)}. \tag{3.17}$$

3.6.6.2 Expected

The average reliability over the entire trace is

$$\omega_k = \frac{\mathtt{g}\left(\mathcal{R}\mathtt{m}_{m^{\mathrm{r}}}\left(\phi_k^{(T_1,T_2]}\right)\right)}{\mathtt{g}\left(\mathcal{R}\mathtt{m}_{m^{\mathrm{r}}}\left(\phi_k^{(T_1,T_2]}\right)\right) + \mathtt{g}\left(\mathcal{R}\mathtt{m}_{m^{\mathrm{d}}}\left(\phi_k^{(T_1,T_2]}\right)\right)}. \tag{3.18}$$

3.7 Conclusions

This chapter addresses the question of tracking the EDCA network evolution over time. To this end, a formal trace framework is proposed. This framework aims at a structured approach to specify event traces for knowledge extraction from them. The analysis carried out represents context semantics of events as they occur in the EDCA execution. Within the scope of the trace framework, four subquestions are studied.

Subquestion a. raises the issue of network behavior emergence from behaviors of individual processes. Given subtraces of processes collected independently, this subquestion aims to combine their effect. We achieve this by a relation from states of processes to states of network using two independent approaches. The first approach is based on istate attributes, which are event details, while the second approach is based on istate transition rate over time. The second approach relies on a distributed view that is available locally with the processes. A local process state is only a partial view of the overall network state. It is not enough to deduce all network states. This lacking is due to the fact that distributed views are projections of overall network state — a phenomenon that is generally valid for distributed systems.

Subquestion b. raises the issue of extraction of low-level performance details, which reflect more accurately on the system properties. They expose internal behaviors to the advantage of anomalies detection. Study of this subquestion leads us to define events with attributes, which we record. The ability to extract low-level details from traces in the form of probabilities is needed to verify our stochastic models at the same level. Indeed, low-level matching offers more confidence in simulation and analytical tools that we realize. The low-level details show behaviors that are not easy to explain. For instance, a sudden drop of occurrence probabilities of backoff slots upon crossing CW boundaries. Such low-level details can be used to make more knowledgeable scheduling decisions for applications.

Subquestion c. raises the issue of identifying and specifying cyclic behaviors projected over different views. Study of recurrence patterns is related to reasoning about fluctuations in QoS properties. We address this subquestion by proposing a formal querying system based on projected cycles. Through these cycles, abnormal behaviorial patterns can be identified.

Subquestion d. raises the issue of determining QoS properties over short-term and long-term durations. To address this subquestion, we develop functions, one per QoS concern, mapping events in traces to QoS properties. Short-term QoS properties reflect more closely on transient behaviors. Their predictability allows cost-effective resource provisioning over time.

Our proposed trace framework can also be used to reason about expectations that different initial conditions give rise to event traces with entirely different QoS properties. The independence of initial conditions can be asserted on a proposition that judges change in average QoS properties upon extending the trace length. Lastly, it fulfills the existing gap between event traces and analytical methods. Network performance highlighted from different viewpoints, can be taken directly to quantify network performance using analytical methods. One such method is stochastic based, which is used to derive performance models for the QoS properties in the next chapter.

4

Stochastic performance models and QoS fulfillment

To derive performance objectively, recasting the system behavior with certain long-standing predictability in stochastic terms is presented in this chapter. The two major but separate concerns, *gaining control* and *retaining control*, as identified and presented in Section 2.7, are key enablers for the development of a system performance model here. The analysis is based on quantification of these two concerns. Therefore, an analysis for the system performance model is split into two parts.

First, we determine the probability that the reference process obtains channel control and subsequently has a successful transmission. This probability determines the expected fraction of total MAC system cycles (cf. Definition 2.4.1) that the reference process transmits successfully — it is identical for processes with the same contention parameters. Its accuracy at all settings of P-tuple in (2.1), is already aimed at realistic conjectures — the fourth objective of the performance framework (cf. Section 2.2). The requirements in this objective lead us to consider the correlation between the slots that occur after transmission. This approach is more realistic and conformant to the actual behavior of EDCA. To this end, we specify a low-level internal state of an EDCA process and aim for determining its steady state probability distribution. A combined effect of these internal states, determine the network steady state. Thus, we are led to the following research question, as posed in Chapter 1:

Question 3. How to develop a quantitative model for network steady state?

Due to a dependency of network steady state on steady states of all processes, there is a need to express the latter steady states. This we achieve by deriving their probability distributions that are mapped from the MAC parameters. Therefore, quantitative analysis for the first concern, *gaining control*, leads to a distribution model for the reference process, which represents probability of its low-level internal state. This model is duly validated analytically under different concrete system setups, each one highlighting performance of a relevant parameter from the P-tuple or their combination.

Second, *retaining control* is quantified. To this end, we examine and consider the expected duration of the MAC system cycles, which is dominated by the expected duration of the state retention.

The advantage of separation of concerns, *gaining control* and *retaining control*, can be readily noted by observing that traffic of applications may have same contention 'levels', but different properties. Hence, a desired feature is to keep process-specific expected duration of retention. This is achieved by exploiting the orthogonality between the two concerns, which allows freedom for application-specific traffic.

Considerations for both concerns together allow to form a representative system equation for the MAC system cycles. This equation is a function of all elements in the P-tuple. The equation and the distribution model allow us to develop models for the QoS properties to address the key problem in Section 2.5.2.3. In context of the research questions presented in Chapter 1, it refers to the following:

Question 4. How to develop quantitative models for QoS properties of MAC processes?

This question is addressed by offering functions, one per QoS concern, mapping steady state probability distributions to QoS properties. The models for the QoS properties are validated analytically and verified against simulations.

Finally, the fulfillment aspect of the QoS problem in Section 2.5 is tackled. Thus, we address the following research question:

Question 5. How to map the communication related QoS requirements of applications to some desired network behavior that fulfills those requirements?

To address this question, we search for a network configuration at which QoS requirements of applications are fulfilled. Specifically, we derive functions, mapping to points in a multiobjective optimization space. The Pareto front in this space offers choices for tradeoffs between multiple objectives. Computing the global optimal point on the Pareto front, is offered as the solution for the QoS problem.

4.1 Related work

We review the available literature per research question addressed in this chapter. Mainly predictive approaches are considered.

Research **Question 3** refers to network average behavior assessed over a long duration. We classify the analytical models in this domain, into three categories: slot-based, zone-based and istate-based. Work in these three categories can be distinguished by the assumptions taken and the details captured.

The slot-based view, first presented in the seminal paper on DCF modelling [17], transcends the system execution pattern, generated by an interleaving of actual idle slots and transmission slots, into what we term as a virtual slot of some expected length. This model is characterized by the probabilities of the virtual slot representing either a contention, a retention or a collision. It relies on the assumption that collisions occur with some constant probability over various retry attempts. Several other works have extended this model by accounting for further details of the backoff process operation [216, 18, 81, 221, 30]. In the seminal model [17], the backoff counter is decremented at the beginning of the virtual slot, irrespective of the event that would occur in it. This assumption being nonconforming to the DCF specifications [91] is rectified later [67, 233, 202]. [52] proposes a distributed channel reservation scheme based on announcing in advance, the slot for transmission, thus also reducing collisions. DCF performance in erroneous channels is considered in [246, 245, 41, 63]. DCF fairness issue due to slow stations is addressed in [106]. Finally, [42, 117] present analysis of the limited *RetryLimit* case. With the increased usage of WLANs, the need of QoS provisioning arose, thus DCF evolved into an enhanced DCF (EDCF), a draft of the EDCA standard [92] — some approaches in [182, 212, 241, 76]. The slot-based view leveraged the derivation of network steady state for performance models of service differentiation

[212, 248, 1]. Performance models for the coexistence of DCF and EDCF/EDCA have been presented in [19, 89, 214, 215, 139, 12]. Service differentiation also allowed to resolve the performance anomaly of IEEE 802.11 [112]. A combined steady state based on the network layer and the MAC layer has been reported in [49]. Lastly, the analytical model in [13], is coupled with its mathematical properties to guarantee avoidance of multiple steady states issue.

The slot-based models for service differentiation lose accuracy in presence of $AIFS$ prioritization. Especially the inaccuracy is pronounced for smaller CW and increasing number of stations. Mainly the assumption that the probability of collision is constant per virtual slot, tends to deviate strongly from the factual MAC operation — in the sense that due to $AIFS$ prioritization, not all stations can access the slots that occur subsequent to the last transmission. This deficiency led to adopting zone-based approaches [168, 115, 211, 235, 44, 223, 84, 105, 116], in which each zone represents a region of consecutive slots that is accessible to a subset of stations. With the exception of a few approaches [87, 247], zone-based models are generally characterized by the assumption that probabilities of transmission and collision per zone are constant. Under this assumption, the network steady state is derived that considers major MAC parameters [197, 160] and MAC reservation [130]. A simplistic view is presented in [87] to reduce the complexity of the zone-based analysis. Network steady state, derived jointly with MAC buffers states, is considered in [96]. The zone-based analysis is also applied in vehicular networks [79]. Finally, the fixed point stability analysis of the zone-based view is studied in [50].

The zone-based analysis generates long expressions that are difficult to understand for an increasing number of zones [168]. Moreover, the key assumption of constant probabilities of transmission and collision per zone is not realistic in some network configurations. These include comparable CW sizes that conserve the effect of $AIFS$ priority. CW sizes beyond the conservation range, mitigate the effect of $AIFS$ priority. Instead of treating slots within zones as independent, a more realistic assumption is to keep the correlation between the slots. This implies probabilistic treatment of each slot that is defined by its istate. An analysis that accounts for these details [201] is more general than any prior approach in the sense of its validity under extreme network configurations. Due to the objectives of our performance framework (cf. Section 2.2), we adopt an istate-based approach to address research **Question 3**.

The challenge in the domain of research **Question 4** is, in general, met by developing quantitative models of QoS properties that rely on the network steady state. For predicting the channel access frequency and the channel share, the approach is to determine a system cycle as a weighted average of contention, retention and collision. Channel access count per cycle duration and the data transmitted therein are the two sought QoS properties, respectively. Based on the same classification as for the network steady state, the channel share is modeled using the slot-based view in [17, 216, 67, 202, 245, 63, 241, 13], the zone-based view in [168, 44, 115, 105] and the istate-based view in [201]. Most of the models do not account for a need of burst transmission by considering $TXOP$ [196]. Of the few that do, $TXOP$ consideration adds a dimension to the Markov models based on other MAC parameters, thus increasing complexity [165, 196, 97, 169, 161, 145, 232]. In our approach, we benefit from the two orthogonal concerns, *gaining control* and *retaining control* (cf. Section 2.7) that allow product terms in the resulting analysis to avoid complexity. The channel access frequency is not explicitly addressed in the literature.

The MAC latency concerns only those MPDUs that reach the receiver. This requires subtraction of losses from a given MPDU flow. Accounting for losses, also enables determining the reliability. The relevant models are based on mainly Markov chains and queueing theory — we do not consider the models based on the latter approach. Analysis for an average loss and latency must consider all details of the network steady state representation. To this end, [42, 248, 197, 87, 115, 148, 47, 154, 87, 231, 18] are based on the slot-based view, [167, 232, 130, 96] on the zone-based view, while no paper is reported that relies on the istate-based view. The accuracy and generalization of these models are similar to assumptions taken to derive the network steady state. Other than models based on our classification, few representative approaches for reliability only, are also examined. A joint strategy is possible that considers application level reliability, based on using application/MAC levels features for robustness [175, 127], and channel efficiency, based on caching in the last hop [229]. Another approach is to selectively repeat the MPDUs and consider the channel errors [131]. A feedback based frame aggregation scheme for video broadcasting in WLANs can improve the reliability [129]. Finally, the combined effect of routing, medium access, and physical layers is considered in [217]. We develop an istate-based analysis for the loss, latency and reliability as a mapping from MAC arbitration parameters. Due to the validity of the assumptions on which the istate-based approach relies, the derived QoS properties closely match with those of the actual EDCA operation.

The need for QoS mapping over a network is highlighted in [159]. Addressing research **Question 5** implies searching a network configuration that can fulfill the QoS requirements of applications. To this end, the mapping explicitly refers to some relation between the QoS requirements and the network configuration. This is explicated to compare our approach with those found in the literature. We classify the proposed work into reactive and proactive approaches, wherein the former are characterized by system adaptability to optimize or for QoS constraint(s) satisfaction, while the latter are characterized by predicability.

Reactive approaches are based on a feedback control that tunes the system and observe the effects with an aim to achieve some optimality. In this context, the EDCA parameters are adjusted observing the effect on the QoS properties, which are implicitly considered as the range of QoS requirements that can be met [174, 94]. For convenience, this consideration is termed as implicit QoS requirements. In contrast to absolute requirements, a parameter control algorithm is proposed to achieve some desired proportional throughput [122]. A case of throughput guarantee for implicit QoS requirements of EDCA stations in the presence of DCF stations is studied [12] — the study omits the prioritization effects of $AIFSN$ per AC. A proportional integral controller based on control theory is proposed to minimize gap between requested and achieved throughputs [80]. Again for implicit QoS requirements, a reactive admission controller is proposed that can translate delays to resource reservations [146]. There have also been research initiatives taken for application specific mappings, of which new video encoding technologies have posed interesting challenges. Most of these works match the importance of the video content to EDCA ACs. Video quality based performance comparison of streaming H.264 [78] video coding layer various frame types to default EDCA ACs is presented in [133]. This work is extended by prioritizing on MPDU dropping [134] and considering the channel conditions [222]. In the same context, transportation of 3D video is studied [82]. For a case of multiple video, a max min optimization problem is formulated to assure minimal video quality guarantee [48]. There are some cross layer proposals that optimize jointly on video content, the DLL queues and the MAC parameters [124, 170] — thus mapping of implicit QoS requirements.

The predictability brought about by the proactive approaches is leveraged by various models for EDCA that propose performance relations as a function of network configurations. Their accuracy depends on the validity of the underlying assumptions used to derive the network steady state. The related work dominantly considers implicit QoS requirements — it can be found in the related work of research **Question 4**. The proposals in [116] and [178] are most relevant to our approach of QoS requirements mapping. In the former work, throughput/delay constrained Pareto curves are presented as tradeoffs between the number of stations in two QoS classes, while the analysis assumes Poisson traffic arrivals [116]. Keeping aside the accuracy of the assumptions taken, these Pareto curves are dependent on multiplicity of number of stations, hence inapplicable in some cases — for instance, when the number of stations per class is one, or even fixed. In the latter work, an algorithm seeks an optimal network configuration for delay requirements of unsaturated traffic and implicit QoS requirements of saturated traffic. The presented approach relies on creating two channels using $AIFS$ differentiation for (un)saturated traffic types. Pareto based optimizations have also been studied for WLAN planning [101] and a joint admission control in integrated WLAN and CDMA cellular networks. These works do not explore the MAC parameter space. There is also an increasing focus on medical applications connected by WLANs. Recent approaches aim for guaranteeing their QoS — for instance, medical graded QoS metric is ensured by absolute priorities [121].

Other than reactive and proactive approaches, the term mapping has also been implied in the sense of matching QoS classes between heterogeneous technologies. Such approaches allow QoS based interoperability between Ethernet, IEEE 1394 and IEEE 802.11e [120] and UMTS and WiMAX [171].

There are four aims we set to address the mapping in research **Question 5**: predictability, formal analysis leading to sound mathematical relations, consideration of all elements in the network parameter tuple (cf. (2.1)) as the range of the mapping, and no assumptions about the application traffic distribution. The first aim leads us to adopt a proactive approach, which we follow to formulate our mapping problem. Achieving these aims together is the contribution out of this research question. Additionally, we allow our solution approach to emerge by considering multiplicity of solutions, thus considering tradeoffs objectively.

4.2 Assumptions and notations

In order to address research **Question 3**, the system model is derived under the following assumptions:

A1 The system operates in a single broadcast domain and there are no hidden nodes.

A2 The channel is ideal. It implies retransmissions are triggered only due to collisions.

A3 A collision is always constructive, i.e., whenever it occurs, all nodes know about it.

A4 Each process runs on a separate node, i.e., a collision due to processes residing on the same node, is not possible.

A5 The MAC buffers on sending processes are work conservative, i.e., there is always an MPDU to transmit.

A6 All nodes are synchronized on PHY slot boundaries.

For the reader's convenience, we compile a list of notations in Table 4.1. It includes some notations from Chapter 2 that are relevant in this chapter also. We typically tag the reference process by $k \in \mathcal{K}$, and use the vector notation to denote its parameters. For example, a_k, R_k, w_k and x_k denote $AIFSN$, $RetryLimit$, CW and $TXOPLimit$ settings, respectively. a_k, R_k and w_k are identical per class.

Table 4.1: List of notations

symbol	meaning
$\mathcal{Q}$	the set of class indices
$\mathcal{K}$	the set of process indices
$\mathbf{n}$	the number of processes per QoS class
$\mathbf{a}$	$AIFSN$ per process
$\mathbf{w}$	CW per process per retry
$\mathbf{R}$	$RetryLimit$ per process
$\mathbf{x}$	$TXOPLimit$ per process
$\boldsymbol{\pi}$	the mean-field probabilities per process per bidimensional state
$\boldsymbol{\rho}$	the probability of accessing the MAC system cycle, followed by a successful transmission, per process
ρ'	the probability of a collision
$\mho$	the expected service time of the MAC system cycle
$\mathcal{T}$	the expected transmission service time in the MAC system cycle
$\mathcal{C}$	the expected contention service time in the MAC system cycle
$\boldsymbol{\mu}$	the expected retention service time in the MAC system cycle, per process
μ'	the expected collision service time in the MAC system cycle
$\mathbb{t}$	the transmission transaction time per process
$\mathbf{L}$	the MPDU paylaod limit per process

4.3 Process istate distribution model

This section derives analytically the istate distribution for the reference process, as a first step towards quantifying the first major concern, *gaining control*. This is identical to the distributions presented in Section 3.4. For this purpose, the parameters that influence *gaining control* are selected from the P-tuple: $AIFSN$ and **CW**. When these two parameters are the same for some processes, then they have identical contention strengths, so that these distributions are symmetric for them. Our analysis is based on the derivation in [201]. The reference process continuously performs the following five operations in the way as described in Chapter 2:

a. mandatory deferral using $AIFS$;

b. counting down the backoff counter;

c. updating the retry counter, the contention window and the backoff counter upon a collision;

d. dropping the HoL MPDU after retransmission attempts exceed the $RetryLimit$;

e. updating the retry counter, contention window and backoff counter upon a successful transmission.

Due to these operations, the reference process undergoes state changes that are detailed in the State Transition Diagram of the EDCA sender process (cf. Figure 2.3). These details are abstracted into two main states: contention and transmission (cf. Section 2.4.1). All transitions from contention to transmission enable identification of MAC system cycles (cf. Section 2.4.2). In order to elaborate on the competition as offered by the competing network against the reference process, a NoP partition of two blocks is created (cf. Section 2.4.3).

The states of these two blocks are hereby termed as a state pair. A state pair considered together has a combined effect, which is the resultant system state. These state pairs are related to their combined effects as given by (originally presented as (2.7))

$$\texttt{Map} : \left(\underset{\text{mac}}{\mathcal{S}}\right)^2 \to \underset{\text{mac}}{\mathcal{S}} .$$

We focus on transitions between the state pairs. They are depicted in Figure 4.1. The system behavior is defined by visiting the peripheral state pairs, which are always interleaved by a visit to the central state pair. The times of two consecutive departures from the central state pair, constitute boundaries of the MAC system cycle. This state pair system forms a Markov process, with each transition effected due to transitions in underlying istates. These istates remain to be defined.

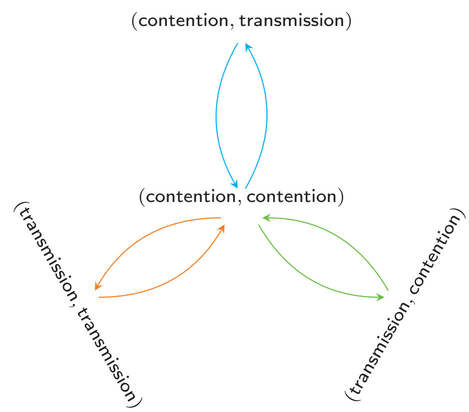


Figure 4.1: A Markov process of state pairs in the domain of $\texttt{Map}(\cdot)$. Transitions are tagged with colors, to establish correspondence with the Markov process of istates in Figure 4.2.

The istate of the reference process, $k \in \mathcal{K}$, at any time t is described by its retry counter and backoff counter pair $(rc_k(t), bc_k(t))$, where probabilities for the transitions between these istates are determined by the stated five operations. Similar to a state pair system, the istate system also forms a bidimensional Markov process of which the steady istate distribution is defined as a time limit probability of process k in istate (r, b) [201]:

$$\pi_{k,r,b} = \lim_{t\to\infty} \mathbb{P}\left(rc_k\left(t\right) = r, bc_k\left(t\right) = b\right), k \in \mathcal{K}, 0 \leqslant r \leqslant R_k, 0 \leqslant b \leqslant w_{k,r}. \tag{4.1}$$

In order to solve the state pair based Markov process, we need to determine the transition probability of each peripheral state pair (cf. Figure 4.1). To this end, we establish mutual correspondence between the state pair based Markov process and istate based Markov process (cf. Figure 4.2). Given a peripheral state pair in the former Markov process, we consider all transitions in the latter Markov process that end up in the same state pair. Towards the effect of state pair transitions in defining a MAC system cycle, we consider all istate transitions that flow into $\pi_{k,r,b}$ to form a system of balanced equations. This system represents istate probability distribution of the reference process. It is solved by a fixed point based iterative approach that is described in Section 4.4. With this background, transition probability of each peripheral state pair is determined as follows.

0. (contention, contention): A new MAC cycle always begins with the start of a contention, during which the mandatory deferral given by minimum $AIFS$ is always completed. Hence, the minimum $AIFS$ is also the guaranteed minimum contention duration. The processes are allowed to transmit only after this minimum $AIFS$, while the actual contention continues till the next transmission. Now consider the system after minimum $AIFS$: the slot immediately next to it is indexed as 0. We denote the $AIFSN$ corresponding to this minimum $AIFS$ by a'. Obviously, the probability of transition from each peripheral state pair to the central state pair is 1.

1. (contention, transmission): A transition to this state pair occurs when only the competing network exhibits a transmission state. The reference process under the contention state may be found undergoing its $AIFS_k$ deferral or decrementing its bc_k — hence, both cases are considered distinctively and accordingly.

   - Within the $AIFS_k$ deferral: The reference process does not change its istate if the competing network starts to transmit while it is still undergoing its $AIFS_k$ deferral. This event is accounted for by the probability of a transmission from the competing network during slot indices within an interval $[0, a_k - a' - 1]$. Considering that the slot next to $AIFS_k$ is numbered 0 for the reference process, $c + a_k - a'$ maps process specific slot number, c, to system specific slot indexes. Let $\gamma(k, c)$ be a probability function of transmission from the competing network at slot number c. transmission at slot number, c, after $AIFS_k$ implies the reference process sees c consecutive idle slots after $AIFS_k$. Contrary to slot indices, which represent absolute numbers, the $\gamma(k, c)$ is defined from the perspective of the reference process - thus c is relative to the reference process. In order to cover cases similar to the current one, in which the competing network may transmit even before the reference process would complete its $AIFS_k$, we extend the definition of the $\gamma(k, c)$ distribution to also allow $c < 0$. Thus, the transmissions from the competing network in the index interval $[0, a_k - a' - 1]$ are covered by $c < 0$. Now the probability of the reference process staying in its current state, is given by the summation over slots within this relative interval, the transmission probability from the competing network:

   $$\pi_{k,r,b} \cdot \sum_{c'=a'-a_k}^{-1} \gamma\big(k, c'\big) \ .$$

   - After the $AIFS_k$ deferral: After the completion of its $AIFS_k$ deferral, the reference process is allowed to decrement its bc_k. For an istate (r, b), we consider all transitions flowing towards it that originate at the same rc_k value, r. Now if the reference process has $bc_k = c'', c'' > b$ at the beginning of the cycle, then the transition $c'' \to b$ requires it to complete its $AIFS_k$ deferral and find exactly $c'' - b - 1$ consecutive idle slots — this is ensured only if the competing network would transmit exactly in the next slot. From the perspective of the reference process, this slot is numbered $c'' - b - 1$ after $AIFS_k$. Hence, the probability of this single flow into istate (r, b) is given by $\pi_{k,r,c''} \cdot \gamma(k, c'' - b - 1)$. In order to consider all similar transitions flowing into istate (r, b), their probabilities need to be summed up over all related slots, $b+1 \leqslant c'' \leqslant w_{k,r}$:

   $$\sum_{c''=b+1}^{w_{k,r}} \pi_{k,r,c''} \cdot \gamma\big(k, c'' - b - 1\big) \ .$$

The two cases considered together form the following transition term:

$$\pi_{k,r,b} \cdot \sum_{c'=a'-a_k}^{-1} \gamma\big(k, c'\big) + \sum_{c''=b+1}^{w_{k,r}} \pi_{k,r,c''} \cdot \gamma\big(k, c'' - b - 1\big), \qquad 0 \leqslant r \leqslant R_k, 0 \leqslant b \leqslant w_{k,r} \ . \tag{4.2}$$

2. (transmission, contention): A transition to this state pair indicates a successful transmission by the reference process leading to a retention. The reference process has $0 \leqslant rc_k \leqslant R_k$ transmission attempts. For an arbitrary cycle, this event can occur only when the reference process transmits earlier than a transmission from the competing network. Now if the reference process has $bc_k = c'''$ at the beginning of this cycle, then it will be a winner of this cycle when it would complete $AIFS_k$, then be allowed to decrement its bc_k in $c''' - 1$ consecutive idle slots and finally complete an *extra slot*. Thus, it would transmit exactly at slot number c'''. In order not to have a collision, all processes in the competing network, $k' \in \mathcal{K} \setminus \{k\}$ must be loosers, i.e., their backoff counters must be greater than $c''' + (a_k - a_{k'})$. The probability of the reference process winning at istate (r, c''') is therefore

$$\pi_{k,r,c'''} \cdot \prod_{k' \in \mathcal{K} \setminus \{k\}} \mathrm{P}\big(bc_{k'} > c''' + (a_k - a_{k'})\big) \ . \tag{4.3}$$

The second factor in the term is denoted by a joint complimentary cumulative distribution function $\beta(k, c''')$, which gives the probability that all processes in the competing network transmit beyond slot number c'''. Similar to $\gamma(k, c)$, $\beta(k, c)$ is also defined from the perspective of the reference process - thus c is relative to the reference process. After the retention, $rc_k \leftarrow 0$ and $CW_k \leftarrow CWmin_k$. Hence, this transition flows towards an istate $(0, b)$, where a value for bc_k is drawn uniformly in an interval $[0, w_{k,0}]$ with probability $\frac{1}{w_{k,0}+1}$. The probability of this event, therefore, is given by a summation over all $rc_k = r$ and $bc_k = c'''$ values, the transitions originating at istates (r, c''') and flowing towards the istate $(0, b)$. The resultant transition term is,

$$\frac{1}{w_{k,0} + 1} \sum_{r=0, c'''=0}^{R_k, w_{k,r}} \pi_{k,r,c'''} \cdot \beta\big(k, c'''\big), 0 \leqslant b \leqslant w_{k,0} \ . \tag{4.4}$$

3. (transmission, transmission): A collision occurs when both blocks in the network partition transmit. For an arbitrary cycle, this event can occur only when the reference process and the competing network, both transmit in the same slot. Now if the reference process has $bc_k = c''''$ at the beginning of this cycle, then it can collide after c'''' consecutive idle slots if at least one process from the competing network would also transmit in the same slot. The probability of this event is given by $\gamma(k, c'''')$. A collision results in $rc_k \leftarrow rc_k + 1$, the next value for CW_k and a redrawal in the interval $[0, w_{k,r}]$ to get a new value for bc_k with probability $\frac{1}{w_{k,r}+1}$. A collision at $rc_k = R_k$ results in a dropout of the HoL MPDU. The generalization for $rc_k \leftarrow R_k + 1$ is kept intact by stating, $R_k + 1 \equiv 0$, so that $rc_k \leftarrow 0$ — also $CW_k \leftarrow CWmin_k$. The probability of one such transition due to a collision, which ends up at istate (r, b) is given by $\pi_{k,r-1,c''''} \cdot \gamma(k, c''''), 0 \leqslant c'''' \leqslant w_{k,r-1}$. Finally, the probability of this event is given by a summation over all c'''', the probability of similar transitions that flow towards the istate (r, b). The transition term due to this event is,

$$\pi_{k,r,b} = \frac{1}{w_{k,r}+1} \sum_{c''''=0}^{w_{k,r-1}} \pi_{k,r-1,c''''} \cdot \gamma\big(k, c''''\big), \qquad 0 \leqslant r \leqslant R_k, 0 \leqslant b \leqslant w_{k,r}. \tag{4.5}$$

Using the transitions in (4.2), (4.4) and (4.5), the joint flow towards istate (r, b) establishes the following system of balanced equations:

$$\begin{aligned}\pi_{k,r,b} \;=\; & \pi_{k,r,b} \cdot \sum_{c'=a'-a_k}^{-1} \gamma(k,c') + \sum_{c''=b+1}^{w_{k,r}} \pi_{k,r,c''} \cdot \gamma(k, c''-b-1) \\ & + \begin{cases} \dfrac{1}{w_{k,0}+1} \displaystyle\sum_{r=0,c'''=0}^{R_k,w_{k,r}} \pi_{k,r,c'''} \cdot \beta\big(k,c'''\big) & r = 0 \\ 0 & r > 0 \end{cases} \\ & + \tfrac{1}{w_{k,r}+1} \sum_{c''''=0}^{w_{k,r-1}} \pi_{k,r-1,c''''} \cdot \gamma(k, c''''), \\ & \qquad 0 \leqslant r \leqslant R_k, 0 \leqslant b \leqslant w_{k,r}\ .\end{aligned} \tag{4.6}$$

All transitions into $\pi_{k,r,b}$ are depicted in Figure 4.2. The $\gamma(\cdot,\cdot)$ and $\beta(\cdot,\cdot)$ distributions are determined now.

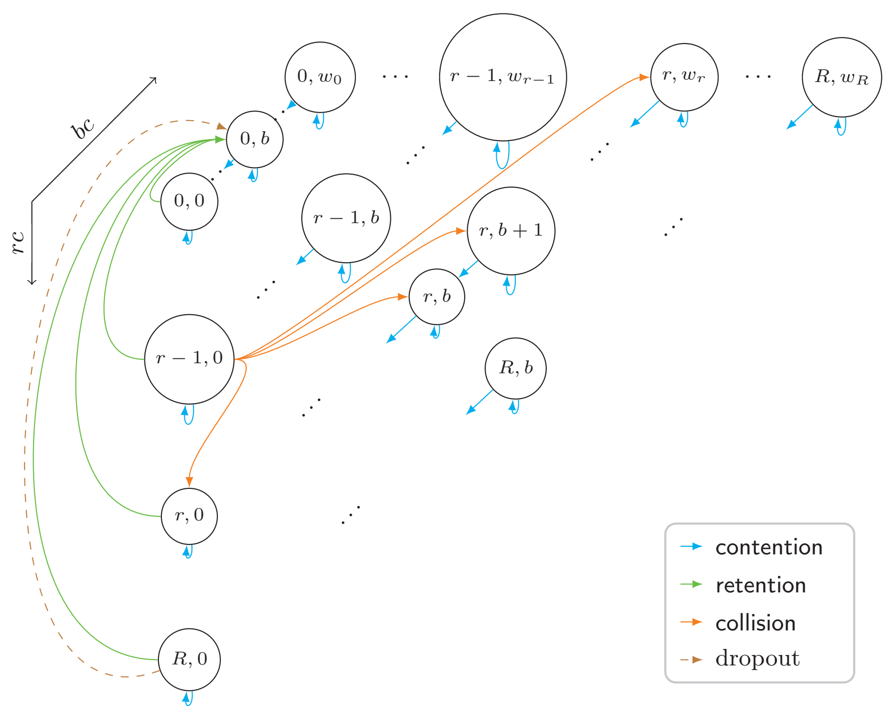


Figure 4.2: A Markov process of istates for the reference process. Colored transitions correspond to transitions between state pairs in the domain of Map(·) in Figure 4.1.

As already stated, $\gamma(k,c)$ gives the probability of transmission by the competing network at slot number c. It is given as a difference between the probabilities of transmission from the competing network beyond slot numbers $c-1$ and c. Hence,

$$\gamma : \mathcal{K} \times \mathbb{Z} \to \mathbb{R},$$
$$\gamma(k,c) = \begin{cases} 1 & c < a' - a_k \\ \beta(k,c-1) - \beta(k,c) & a' - a_k \leqslant c \leqslant CWmax_k \\ 0 & c > CWmax_k \end{cases}. \tag{4.7}$$

As a reminder, $\mathcal{K}$, $\mathbb{Z}$ and $\mathbb{R}$ denote in this thesis, the sets of indices for processes, integers and reals, respectively. As already indicated, $c < 0$ allows to cover cases when the competing network may transmit before the reference process would complete its $AIFS_k$.

The $\beta(k,c)$ distribution models the exceedance of the competing network over $bc_k = c$. It is determined by a product over all processes in the competing network, their tail distributions that exceed c:

$$\beta : \mathcal{K} \times \mathbb{Z} \to \mathbb{R},$$
$$\beta(k,c) = \begin{cases} 1 & c < a' - a_k \\ \prod_{k' \in \mathcal{K} \setminus \{k\}} \left(1 - \mathbb{P}(0 \leqslant bc_{k'} \leqslant c + (a_k - a_{k'}))\right) & a' - a_k \leqslant c \leqslant CWmax_k \\ 0 & c > CWmax_k \end{cases}. \tag{4.8}$$

The probability term represents a cumulative distribution function for process(es) in the competing network. It can be expressed as a summation over slot numbers i, the probability of finding $bc_{k'} = i$. Let this probability, $\mathbb{P}(bc_k = i)$ be given by a probability density function $\alpha(k,i)$, so that

$$\beta(k,c) = \prod_{k' \in \mathcal{K} \setminus \{k\}} \left(1 - \sum_{i=0}^{c + \left(a_k - a_{k'}\right)} \alpha\left(k', i\right)\right), a' - a_k \leqslant c \leqslant CWmax_k. \tag{4.9}$$

In order to determine $\alpha(k,c)$, observe that the reference process may be found with the same bc_k value at different rc_k values — so that irrespective of different rc_k values, all same bc_k values must be considered. Therefore, in order to find a rc_k agnostic distribution, the bidimensional Markov process in (4.1) needs to be flattened to a unidimensional process. Hence, we obtain $\alpha(k,c)$ by summing over all rc_k values, the probability of finding $bc_k = c$:

$$\alpha : \mathcal{K} \times \mathbb{N}_0 \to \mathbb{R},$$
$$\alpha(k,c) = \begin{cases} \sum_{r=0}^{R_k} \pi_{k,r,c} & 0 \leqslant c \leqslant CWmax_k \\ 0 & \text{else} \end{cases}. \tag{4.10}$$

For undefined summands, i.e., when $c > w_{k,r}$, we assume $\pi_{k,r,c} = 0$.

4.4 The fixed point convergence

Due to symmetry, all processes within a class have similar stochastic properties. Without loosing generality, this allows us to rewrite the istate steady distribution, $\pi_{k,r,b}$ in (4.1) as

$\pi_{q,r,b}$. Hence, it suffices to form one system of balanced equations in (4.6) for each class. These systems are solved iteratively to seek fixed points for $\pi_{q,r,b}, q \in \mathcal{Q}, 0 \leqslant r \leqslant R_q, 0 \leqslant b \leqslant w_{q,r}$. For this purpose, this section computes the fixed points of $\gamma(\cdot,\cdot)$, $\beta(\cdot,\cdot)$ and $\alpha(\cdot,\cdot)$ distributions for each class, iteratively. During iterations, the n^{th} step distributions are superscripted with (n). The current distributions are considered to be functions of their previous counterparts. For instance,

$$\pi_{q,r,b}^{(n)} = \mathtt{f}\left(\pi_{q,r,b}^{(n-1)}\right).$$

Successive iterations are carried out till convergence is achieved. The convergence criteria $\mathtt{C}(\cdot,\cdot)$ is defined over system scope. The system reaches the convergence when the Hilbert-Schmidt norm of a difference matrix, $\delta_{q,b} = \alpha^{(n)}(q,b) - \alpha^{(n-1)}(q,b)\,, q \in \mathcal{Q}$ has reached an error precision, ϵ:

$$\begin{aligned} \mathtt{C} : \prod_{q \in \mathcal{Q}} \mathbb{R}^{CWmax_q} \times \mathbb{R} \rightarrow \{\text{true}, \text{false}\}\,, \\ \mathtt{C}\,(\boldsymbol{\delta}, \epsilon) = \begin{cases} \text{false} & \parallel \boldsymbol{\delta} \parallel_2 > \epsilon \\ \text{true} & \parallel \boldsymbol{\delta} \parallel_2 \leqslant \epsilon \end{cases}, \end{aligned} \tag{4.11}$$

where

$$\parallel \boldsymbol{\delta} \parallel_2 = \left(\sum_{q \in \mathcal{Q}} \sum_{b=0}^{CWmax_q} |\delta_{q,b}|^2 \right)^{1/2}.$$

The steps taken to achieve the fixed point are outlined in a pseudocode in Algorithm 6. This algorithm requires an initial istate distribution to enter into iterations. During each iteration, the $\alpha(\cdot,\cdot)$, $\beta(\cdot,\cdot)$ and $\gamma(\cdot,\cdot)$ distributions are computed to form the coefficients of the balanced equations, whose solutions lead to the next step. The initial istate distribution is estimated with an aim to reduce the convergence time. For it, we assume that rc_q is geometrically distributed — for a given rc_q value, bc_q is again geometrically distributed. The computations are done as follows.

A geometric distribution yields a geometric sequence: $z, z \cdot y, z \cdot y^2 \cdots$, where z is the first term and $0 < y < 1$ is the ratio between consecutive terms. The i^{th} term of this sequence is given as $z_i = z \cdot y^{i-1}$, while the sum of first i terms is given as $S_i = z \cdot \frac{1-y^i}{1-y}$. Reducing z, yields $z_i = S_i \cdot \frac{1-y}{1-y^i} \cdot y^{i-1}$. Applying this relation to the case of the istate distribution to find the probability of finding $rc_q = r, 0 \leqslant r \leqslant R_q$ yields

$$\mathtt{P}(rc_q = r) = S_{R_q+1} \cdot \frac{1-y}{1-y^{R_k+1}} \cdot y^r\;.$$

The istate distribution is normalized, so that $S_{R_q+1} = 1$. Given $rc_q = r$, bc_q is geometrically distributed:

$$\begin{aligned} \mathtt{P}(bc_{q,r} = b) = \mathtt{P}(rc_q = r) \cdot \frac{1-y}{1-y^{w_{q,r}+1}} \cdot y^b, \\ \pi_{q,r,b}^{(1)} = \frac{1-y}{1-y^{R_q+1}} \cdot y^r \cdot \frac{1-y}{1-y^{w_{q,r}+1}} \cdot y^b\;. \end{aligned} \tag{4.12}$$

This computation is invoked by a call to a helper function, $\mathtt{h}(q,r,b,y)$,

$$\mathtt{h} : \prod_{q \in \mathcal{Q}} \mathbb{N}_0 \times \prod_{r=0}^{R_q} \mathbb{N}_0 \times \prod_{b=0}^{w_{q,r}} \mathbb{N}_0 \times (0,1) \to \mathbb{R}. \tag{4.13}$$

The selection of a geometric distribution for initial estimation is motivated by the observation that rc_q and $bc_{q,r}$ obtaining larger values, is less probable. Finally please note the following:

- Each iteration in Algorithm 6, updates relevant distributions of all classes. This ensures that class trajectories evolve together to correctly define the system trajectory, as desired for convergence;

- The difference matrix used in the convergence criteria considers distributions of all classes in the system. This is to prevent convergence which is based on a single class alone, albeit it be a most preferred class. This strategy avoids early convergence, while distributions of remaining classes are still approaching their fixed points.

Algorithm 6 $\mathtt{FPC}(\epsilon, ratio)$
// fixed point convergence

Require: $\epsilon \in \mathbb{R}$, $ratio \in \mathbb{R}$
Ensure: istate stationary distribution $\pi_{q,r,b}, q \in \mathcal{Q}$

// initial estimate of istate distribution
$n \leftarrow 1$
for all $q \in \mathcal{Q}, 0 \leqslant r \leqslant R_q, 0 \leqslant b \leqslant w_{q,r}$ **do**
    $\pi^{(n)}_{q,r,b} \leftarrow \mathtt{h}(q, r, b, ratio)$, refers (4.13)
end for
for all $q \in \mathcal{Q}$ **do**
    Compute $\alpha^{(n)}(q, \cdot)$, $\beta^{(n)}(q, \cdot)$, $\gamma^{(n)}(q, \cdot)$ distributions, refers (4.10), (4.8) and (4.7)
end for

// iterations
do
    $n \leftarrow n + 1$
    $\pi^{(n)}_{q,r,b} \leftarrow$ solution of (4.6)
    for all $q \in \mathcal{Q}$ **do**
        Compute $\alpha^{(n)}(q, \cdot)$, $\beta^{(n)}(q, \cdot)$, $\gamma^{(n)}(q, \cdot)$ distributions
    end for
    $\delta_{q,b} = \alpha^{(n)}(q, b) - \alpha^{(n-1)}(q, b)\,, q \in \mathcal{Q}, 0 \leqslant b \leqslant CWmax_q$
while not $\left(\mathbb{C}\,(\delta, \epsilon)\right)$
return

4.5 Quantification of major concerns

This section quantifies the two major concerns identified in Section 2.7. Our analysis is based on the stationary distribution of istate Markov process $\pi_{k,r,b}$, as derived in the last section. Each concern is tackled separately.

4.5.1 *Gaining control*

This concern is quantified in terms of the transmission priority of the reference process. It is determined by the probability that it gains channel control and subsequently has a successful transmission. This probability determines the expected fraction of the number of MAC system cycles that it transmits, and is identical for processes within the same class.

Each MAC system cycle starts with a contention which continues till the next transmission. Let ρ_k denote the probability that the reference process is selected at the end of the contention and that this process subsequently performs a successful transmission. ρ_k is given by a summation over all possible consecutive idle slots, the probability of winning the MAC system cycles after them.

$$\rho_k = \sum_{b=0}^{CWmax_k} \left(\mathrm{P}(bc_k = b) \cdot \prod_{k' \in \mathcal{K} \setminus \{k\}} \mathrm{P}(bc_{k'} > b + (a_k - a_{k'})) \right). \tag{4.14}$$

The first factor in the summation is the $\alpha(\cdot,\cdot)$ distribution. Using the discussion before (4.4), the second factor is identified to be the $\beta(\cdot,\cdot)$ distribution in (4.8). Thus,

$$\rho_k = \sum_{b=0}^{CWmax_k} \alpha(k,b) \cdot \beta(k,b) \ . \tag{4.15}$$

4.5.2 *Retaining control*

This concern is quantified by considering the duration of a successful transmission. Based on this, we derive the expected service time of the MAC system cycles (see Section 2.4.2). This expected service time consists of two parts: the expected contention service time and the expected transmission service time:

$$\mho = \mathcal{C} + \mathcal{T}. \tag{4.16}$$

For computing the expected contention service time, observe that starting from slot index 0, a contention duration of i slots means that precisely i adjacent slots remain idle in the system. This happens if the next transmission occurs precisely at slot index i, disregarding the number of process(es) initiating it. This transmission implies that initiating process(es) have already completed their respective $AIFS$. They satisfy the condition that the slot index i falls beyond their respective $AIFS$. The relation between the set of processes that can transmit at slot i is readily seen by an assertion. Let $\mathcal{J}$ be a set of these transmitting process(es). Then this observation implies $\underset{j \in \mathcal{J}}{\forall} (a_j - a' \leqslant i)$.

In order for the next transmission to occur precisely at slot index i, all processes must have their bc values at least i: $\underset{l \in \mathcal{K}}{\forall} (bc_l \geqslant i - (a_l - a'))$. The probability of this event is given by

$$\prod_{l \in \mathcal{K}} \mathrm{P}\big(bc_l \geqslant i - \big(a_l - a'\big)\big) = \prod_{l \in \mathcal{K}} \left(1 - \sum_{b=0}^{i-\left(a_l - a'\right)} \alpha(l,b) \right). \tag{4.17}$$

The expected contention service time is obtained by summing over all possible slot indices, the probability of next transmission in the system at each of them, multiplied by the slot time, $aSlotTime$. The maximum slot index to consider for transmission is given by I',

$$I' = \downarrow_{l \in \mathcal{K}} \left(CWmax_l + \left(a_l - a'\right)\right). \tag{4.18}$$

Beyond it, the probability of having idle slots is zero. Thus,

$$\mathcal{C} = aSlotTime \cdot \sum_{i=0}^{I'} \left(\prod_{l \in \mathcal{K}} \left(1 - \sum_{b=0}^{i-\left(a_l - a'\right)} \alpha(l,b) \right) \right) + \left(a' \cdot aSlotTime + SIFS\right), \tag{4.19}$$

where the last term is the time corresponding to the smallest $AIFS$.

$\mathcal{T}$ consists, again, of two parts: retention and collision. The expected retention service time is given by a weighted average of retention service times of all processes, where the weights are their probabilities of accessing the MAC system cycles. Let μ_k denote the expected retention service time of process k. The expected collision service time is given by a product of probability of collision, $\rho' = 1 - \sum_{k \in \mathcal{K}} \rho_k$ and the expected collision service time, μ'. Now

$$\mathcal{T} = \sum_{i \in \mathcal{K}} \rho_i \cdot \mu_i + \rho' \cdot \mu'. \tag{4.20}$$

μ_k depends on $TXOPLimit_k$ transmission transactions (cf. Definition 2.7.1). Let $\mathbf{x}$, $\mathbf{tt}$ and $\mathbf{L}$ denote the vectors of the $TXOPLimit$, the transmission transaction time and the application payload limit per MPDU in bytes, all per process, respectively. Then

$$\mu_k = x_k \left(tt_k + SIFS\right) - SIFS. \tag{4.21}$$

During the transmission transaction, the MPDU and the corresponding ACK are transmitted at the MAC data rate, $\mathscr{R}^{\mathrm{d}}$. Both are preceded by their respective PHY headers (the preamble), which are transmitted at the MAC basic rate, $\mathscr{R}^{\mathrm{b}}$.

$$tt_k = \frac{h^{\mathrm{phy}}}{\mathscr{R}^{\mathrm{b}}} + \frac{8\left(L_k + H\right)}{\mathscr{R}^{\mathrm{d}}} + t^{\mathrm{p}} + SIFS + \frac{h^{\mathrm{phy}}}{\mathscr{R}^{\mathrm{b}}} + \frac{8 \cdot ACK}{\mathscr{R}^{\mathrm{d}}} + t^{\mathrm{p}}, \tag{4.22}$$

where H is the sum of lengths (in number of bytes) of the OSI headers and t^{p} is the propagation delay. The time overhead per MPDU in a transmission opportunity, t^{o}, includes time for transmitting all headers, two propagation delays, ACK transmission time and two $SIFS$. It is expressed using tt_k,

$$t^{\mathrm{o}} = \left(tt_k - \frac{8 \cdot L_k}{\mathscr{R}^{\mathrm{d}}}\right) + SIFS. \tag{4.23}$$

Finally the expected collision service time is determined by noting that the occurrence of a collision is observed at the end of the first transmission transaction in an acquired transmission opportunity. On its detection, the transmission opportunity is terminated. Therefore, the expected collision service time is determined by the expected service time of the first MPDU, which is the leading MPDU in the transmission opportunity. The expected collision service time is computed by omitting k subscript in (4.21), taking $x = 1$ and considering the longest transmission transaction time as dependent on the largest application payload limit per MPDU,

$$\mu' = \uparrow_{k \in \mathcal{K}} tt_k. \tag{4.24}$$

This concludes the description of the stochastic model.

4.6 Designing experiments

In this section, we address the following questions:

- How to validate the presented process istate distribution model?
- How to validate models for the QoS properties, and verify them against simulations? This includes selection of an appropriate simulator. These models will be derived later in this chapter and are needed to address the QoS problem (cf. Section 2.5).

We proceed by designing system configurations, grouped in four sets of experiments. Each experiment exercises a different priority mechanism (i.e., MAC settings) that is applied in a general system context. Hence, during each experiment, multiple settings are tested to confirm the hypothesis on priority mechanisms and assert the soundness of proposed models.

4.6.1 General system setup

A setup common to all experiments, shown in Figure 4.3, defines the general scope of the system.

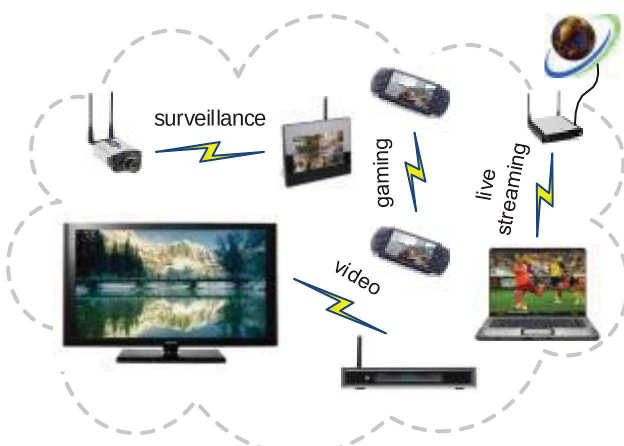


Figure 4.3: QoS based system setup with eight devices on the same ad-hoc network, arranged in four transmitter/receiver pairs.

Priority levels Four different streams share the channel, which we represent by four classes of one stream each: $\mathcal{Q} = \{0, 1, 2, 3\}$ and $\mathbf{n} = [1, 1, 1, 1]$, so that $\mathcal{K} = \{1, 2, 3, 4\}$. Taking four classes is in conformance with the EDCA default number of classes, given in IEEE 802.11e standard [92].

Traffic model The application sources are assumed to generate Constant Bit Rate (CBR) streams, which are transmitted using the Real Time Procotol (RTP) over the User Datagram Protocol (UDP). The CBR frame size is selected such that after adding the RTP, the UDP, the Internet Protocol (IP) and the MAC headers, the MPDU size becomes 1500 bytes. All application sources are work conservative. This is to ensure MAC input queue saturation, as assumed in the stochastic model (cf. Section 4.2), wherein a new MPDU is scheduled immediately after the lifetime of the current MPDU. For ensuring similar conditions in simulations, MPDUs are generated at ten times higher rate than their consumption on the network.

Network mode The network is operated in an ad-hoc mode, assuming that all devices are in the broadcast range of each other.

Channel access mechanism EDCA is used to access the channel at the MAC sublayer, whose parameters draw values from the following sets:

- $AIFSN \in \{1, \cdots, 7\}$;
- $CW \in \{7, \cdots, 1023\}$. CW grows until three steps;
- $TXOP \in \{1, \cdots, 7\}$.

OSI parameters Our EDCA network runs atop IEEE 802.11b[1] [92] PHY parameters, which provide an effective bandwidth of about 7 Mbps. These parameters are listed in Table 4.2, along with the OSI headers that are used to configure the system and to compute the time overhead, t^{o}.

Simulation time Excluding the initial start-up and final wind-off times, all simulations are run for 100 seconds. The averages over this interval are reported for all results.

System convergence Algorithm 6 is executed on a dual core CPU @2.80 GHz laptop and invoked as $\texttt{FPC}\left(10^{-10}, 0.75\right)$.

Table 4.2: OSI headers and IEEE 802.11b PHY/MAC parameters

parameter		value
$aSlotTime$		20 µs
$SIFS$		10 µs
propagation delay (t^{p})		1 µs
Basic rate ($\mathscr{R}^{\mathrm{b}}$)		1 bit/µs
Data rate ($\mathscr{R}^{\mathrm{d}}$)		11 bits/µs
ACK		14 bytes
PHY header (h^{phy})		192 bits
OSI headers (H)	RTP	12 bytes
	UDP	8 bytes
	IP	20 bytes
	MAC	28 bytes
CBR frame size (L_k)		1432 bytes
MPDU size ($L_k + H$)		1500 bytes
time overhead per MPDU (t^{o})		466 µs

[1] IEEE 802.11b is the cheapest and the most common wireless technology due to its large coverage, which allows to avoid interference from other appliances. The analysis for the process istate distribution model (cf. Section 4.3) is independent of specific WLAN technology, so that the quantification of *gaining control* (Section 4.5.1) can be applied for any WLAN technology. For quantification of *retaining control* (Section 4.5.2), only (4.22) is WLAN technology dependent.

4.6.2 Concrete system setups

The MAC settings specific for different experiments, are presented here. They are applied in the general system context. We use a *table/row* notation to refer to the data in the presented tables, where the table, *table*, contains the row number, *row*, marked on the right side of table rows.

4.6.2.1 *Gaining control* – $AIFSN$ driven arbitration

This experiment targets to highlight the MAC arbitration because of $AIFSN$ prioritization. The priority order in $\mathcal{Q}$ translates to MAC settings with the property that for $q, q' \in \mathcal{Q}'$, if $q < q'$ then

$$\begin{aligned} a_q &< a_{q'}, \\ \mathbf{w}_q &= \mathbf{w}_{q'} \ . \end{aligned} \tag{4.25}$$

Let the $AIFSN$ and CW settings be collected in sets $\mathcal{A}^a$ and $\mathcal{W}^a$, respectively. Both sets are presented in Table 4.3. The performance is studied at all setting points in $\mathcal{A}^a \times \mathcal{W}^a$. $RetryLimit = 7$ in all experiments. Similar to the CW, the $TXOP$ settings are also kept uniform for all processes using $\mathbf{x} = [1, 1, 1, 1]$.

Table 4.3: $AIFSN$ driven arbitration.

(a) $AIFSN$ settings.

$\mathbf{a} \in \mathcal{A}^a$	
$[1, 1, 1, 1]$	1
$[1, 2, 3, 4]$	2
$[1, 2, 4, 5]$	3
$[1, 2, 5, 6]$	4
$[1, 2, 6, 7]$	5

(b) Uniform CW settings for all processes, with $CWmin$ in bold.

$\mathbf{w} \in \mathcal{W}^a$	
$\begin{bmatrix} \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \\ \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \\ \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \\ \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \end{bmatrix}$	1
$\begin{bmatrix} \mathbf{15}, 31, 63, 127, 127, 127, 127, 127 \\ \mathbf{15}, 31, 63, 127, 127, 127, 127, 127 \\ \mathbf{15}, 31, 63, 127, 127, 127, 127, 127 \\ \mathbf{15}, 31, 63, 127, 127, 127, 127, 127 \end{bmatrix}$	2
$\begin{bmatrix} \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \\ \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \\ \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \\ \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \end{bmatrix}$	3
$\begin{bmatrix} \mathbf{63}, 127, 255, 511, 511, 511, 511, 511 \\ \mathbf{63}, 127, 255, 511, 511, 511, 511, 511 \\ \mathbf{63}, 127, 255, 511, 511, 511, 511, 511 \\ \mathbf{63}, 127, 255, 511, 511, 511, 511, 511 \end{bmatrix}$	4

4.6.2.2 *Gaining control* – CW driven arbitration

This experiment targets to highlight the MAC arbitration due to CW prioritization. The priority order in $\mathcal{Q}$ translates to MAC settings with the property that for $q, q' \in \mathcal{Q}'$, if $q < q'$

then

$$\begin{aligned} a_q &= a_{q'}, \\ \mathbf{w}_q &< \mathbf{w}_{q'}, \end{aligned} \tag{4.26}$$

where $\mathbf{w}_q < \mathbf{w}_{q'} \equiv w_{q,r} < w_{q',r}, 0 \leqslant r \leqslant RetryLimit$. In this case, CW prioritization is created at various $RetryLimit$ values, which are kept uniform across the board. Let these $RetryLimit$ values be collected in a set, $\mathcal{R}^w = \{r \cdot \mathbf{1} \mid 1 \leqslant r \leqslant 7\}$, where $\mathbf{1} = [1, 1, 1, 1]$ is a unit vector. Further, let the CW values for $RetryLimit = 7$ be collected in a set, $\mathcal{W}^w$, given in Table 4.4. For $RetryLimit < 7$, appropriate parts from elements of $\mathcal{W}^w$ are considered. For brevity, these parts are still considered as members of $\mathcal{W}^w$. $AIFSN$ and $TXOP$ settings are also kept uniform for all classes, with $\mathbf{a} = [1, 1, 1, 1]$ and $\mathbf{x} = [1, 1, 1, 1]$, respectively.

Table 4.4: CW driven arbitration: CW settings.

$\mathbf{w} \in \mathcal{W}^w$	
$\begin{bmatrix} \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \\ \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \\ \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \\ \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \end{bmatrix}$	1
$\begin{bmatrix} \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \\ \mathbf{15}, 31, 63, 127, 127, 127, 127, 127 \\ \mathbf{23}, 47, 95, 191, 191, 191, 191, 191 \\ \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \end{bmatrix}$	2
$\begin{bmatrix} \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \\ \mathbf{15}, 31, 63, 127, 127, 127, 127, 127 \\ \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \\ \mathbf{39}, 79, 159, 319, 319, 319, 319, 319 \end{bmatrix}$	3
$\begin{bmatrix} \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \\ \mathbf{15}, 31, 63, 127, 127, 127, 127, 127 \\ \mathbf{39}, 79, 159, 319, 319, 319, 319, 319 \\ \mathbf{47}, 95, 191, 383, 383, 383, 383, 383 \end{bmatrix}$	4

4.6.2.3 *Retaining control* – $TXOP$ driven performance

This experiment targets to highlight the dependence of system performance on duration of the retention. The priority order in $\mathcal{Q}$ translates to MAC settings with the property that for $q, q' \in \mathcal{Q}'$, if $q < q'$ then

$$\begin{aligned} a_q &= a_{q'}, \\ \mathbf{w}_q &= \mathbf{w}_{q'}, \\ x_q &< x_{q'} \ . \end{aligned} \tag{4.27}$$

Let $\mathcal{X}^x$ denote a set of $TXOP$ settings tested. Other MAC arbitration settings are kept uniform across the board: $\mathbf{a} = [1, 1, 1, 1]$ and let CW settings be collected in a set, $\mathcal{W}^x$, for $RetryLimit = 7$. Both $\mathcal{X}^x$ and $\mathcal{W}^x$ are presented in Table 4.5.

Table 4.5: $TXOP$ driven performance.

(a) $TXOP$ settings.

$\mathbf{x} \in \mathcal{X}^x$	
$[1, 1, 1, 1]$	1
$[7, 6, 5, 4]$	2
$[6, 5, 4, 3]$	3
$[5, 4, 3, 2]$	4
$[4, 3, 2, 1]$	5

(b) Uniform CW settings for all processes, with $CWmin$ in bold.

$\mathbf{w} \in \mathcal{W}^x$	
$\begin{bmatrix} \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \\ \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \\ \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \\ \mathbf{7}, 15, 31, 63, 63, 63, 63, 63 \end{bmatrix}$	1
$\begin{bmatrix} \mathbf{15}, 31, 63, 127, 127, 127, 127, 127 \\ \mathbf{15}, 31, 63, 127, 127, 127, 127, 127 \\ \mathbf{15}, 31, 63, 127, 127, 127, 127, 127 \\ \mathbf{15}, 31, 63, 127, 127, 127, 127, 127 \end{bmatrix}$	2
$\begin{bmatrix} \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \\ \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \\ \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \\ \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \end{bmatrix}$	3

4.6.2.4 Default EDCA parameter set driven arbitration

The last experiment is designed using the default EDCA parameter set [92, Table 7-37], which contains only one network configuration. It is reproduced here in Table 4.6. Let $\mathcal{W}^d$ denote the set of CW settings at $RetryLimit = 7$, for the default EDCA parameter set — given in Table 4.7. Given the recommended setting, EDCA admits only a best-effort delivery service in absence of a prediction model. The dramatic variation of the delivery service with the system load renders QoS properties unpredictable. We, therefore, assert that the recommended setting can not specify a system with deterministic QoS properties. Specifically the two major concerns can not be quantified at such an incompleteness. For *gaining control*, it does not guide on the number of processes and allowed retry attempts for them. For *retaining control*, it does not guide on the MPDU sizes, so that the number of transmission transactions is not known.

These shortcomings offer us the opportunity to fill the gap and study the system performance under various settings of $RetryLimit$ and the number of processes per class. Both parameters are kept uniform across all classes and collected in sets of scaled unit vectors, $\mathcal{R}^d$ and $\mathcal{N}^d$, respectively: $\mathcal{R}^d = \{r \cdot \mathbf{1} \mid 1 \leqslant r \leqslant 7\}$ and $\mathcal{N}^d = \{n \cdot \mathbf{1} \mid 1 \leqslant n \leqslant 5\}$. For $RetryLimit < 7$, appropriate parts from elements of $\mathcal{W}^d$ are considered. Finally, $\mathbf{x} = [3, 2, 1, 1]$.

Table 4.6: Default EDCA parameter set. $aCWmin$ and $aCWmax$ are IEEE 802.11 [92] notations for $CWmin$ and $CWmax$. Note that our class priority order is increasing, in contrast to this decreasing order.

AC	$CWmin$	$CWmax$	$AIFSN$	$TXOP$
0	$aCWmin$	$aCWmax$	7	0
1	$aCWmin$	$aCWmax$	3	0
2	$(aCWmin + 1)/2 - 1$	$aCWmin$	2	6.016 ms
3	$(aCWmin + 1)/4 - 1$	$(aCWmin + 1)/2 - 1$	2	3.264 ms

Table 4.7: CW settings for the default EDCA parameter set.

AC	$\mathbf{w} \in \mathcal{W}^d$	$q \in \mathcal{Q}$
0	$\mathbf{31}, 63, 127, 255, 511, 1023, 1023, 1023$	3
1	$\mathbf{31}, 63, 127, 255, 511, 1023, 1023, 1023$	2
2	$\mathbf{15}, 31, 31, 31, 31, 31, 31, 31$	1
3	$\mathbf{7}, 15, 15, 15, 15, 15, 15, 15$	0

4.7 Process istate distribution model – validation

The presented stochastic model is validated by predicting the steady istate distribution, $\pi_{k,r,b}$ in (4.1) in various concrete system setups. The derived distribution will be used later to determine the QoS properties. For zooming in on significant probability values, we do not show negligible tail values. Additionally, we also present performance details of Algorithm 6, to judge its feasibility over other software/hardware platforms.

4.7.1 *Gaining control* – $AIFSN$ driven arbitration

The results using the system setups in Section 4.6.2.1 are presented here.

4.7.1.1 Steady istate distribution

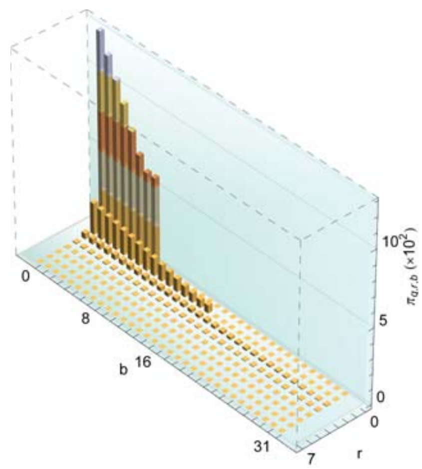
The long-term average behavior is reported in terms of probability distributions. They provide low-level performance quantifications to allow deep inspection of the backoff process. All distributions are positively skewed. For brevity we pick only one setting from Table 4.3 to report the $\pi_{k,r,b}$ distribution: $\mathcal{A}^a \ni \mathbf{a} = 4.3a/2$ and $\mathcal{W}^a \ni \mathbf{w} = 4.3b/1$. The results are presented in two perspectives:

Class view

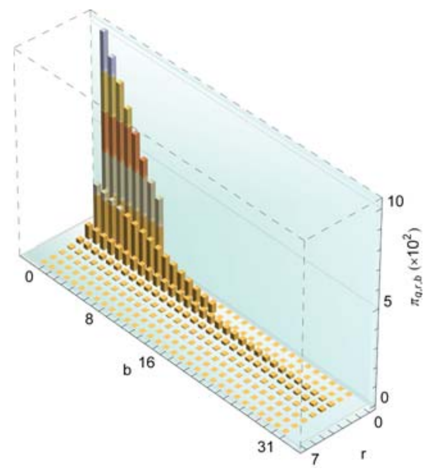
The class view in Figure 4.4, highlights the steady distribution of a single process from each class over various rc and bc values. The hypothesis that a process is found more often in lower bc and lower rc values, is clearly demonstrated across the board.

Retry view

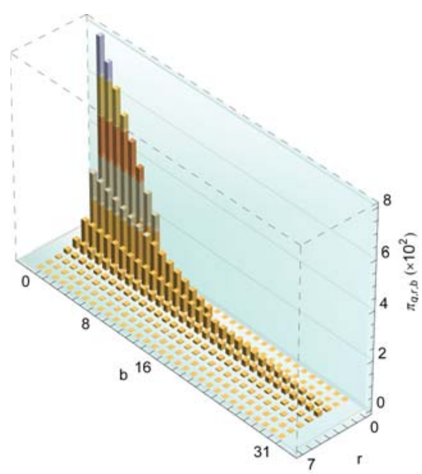
The retry view in Figure 4.5, compares the distributions at different rc values. We notice that for decreasing class priority, the probability mass is increasing with rc values. It is accounted for as follows. Due to different $AIFSN$ settings, the higher priority classes experience lesser competition on $AIFS$ based protected slots, which are lower in order. This aspect is quantified by the first term on the RHS of (4.2). It allows the higher priority classes to transmit mostly earlier than the lower priority classes, i.e., at lower rc values. This trend decreases the competition on the higher order rc, to benefit the lower priority classes there. In other words, the probability mass towards higher rc values is decreasing with class priority and vice versa. Consequently, higher rc values are seen more often by lower priority classes, as they are pushed to these regions of low competition.



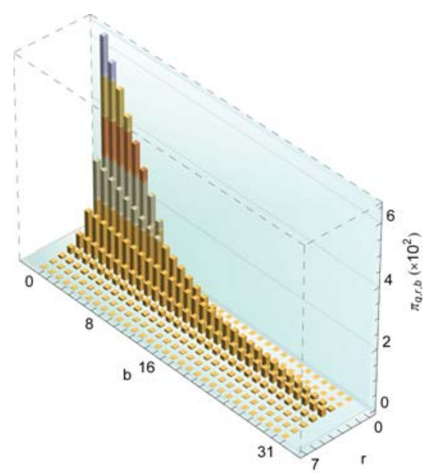
(a) Class, $q = 0$.



(b) Class, $q = 1$.

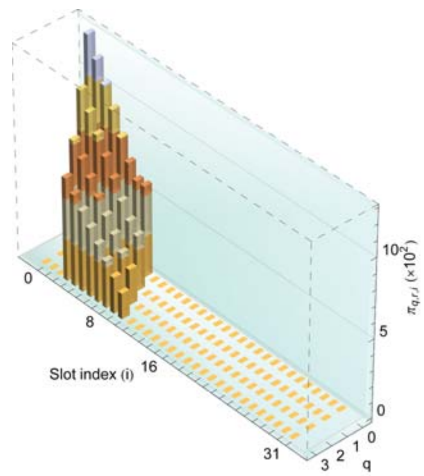


(c) Class, $q = 2$.

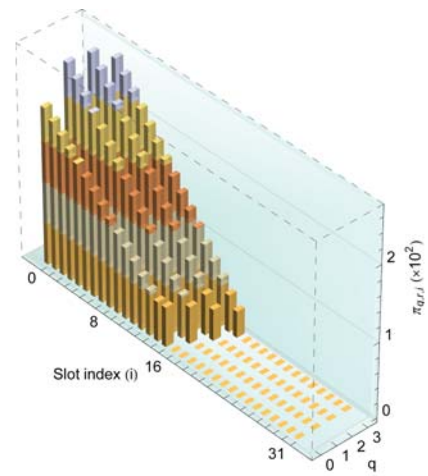


(d) Class, $q = 3$.

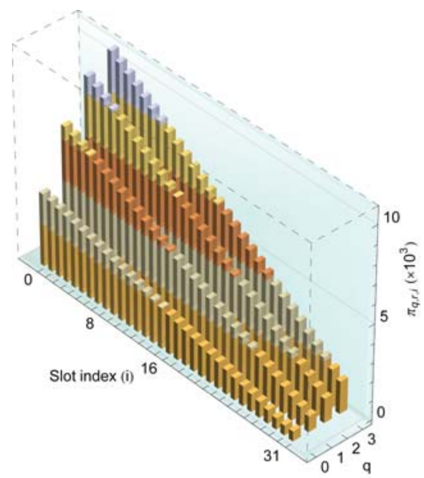
Figure 4.4: *Gaining control* – $AIFSN$ driven arbitration performance using steady istate distribution, $\pi_{k,r,b}$. Configuration: $\mathcal{A}^a \ni \mathbf{a} = 4.3a/2$ and $\mathcal{W}^a \ni \mathbf{w} = 4.3b/1$. Please note the changing range of $\pi_{q,r,b}$ axis, making graphs appear similar.



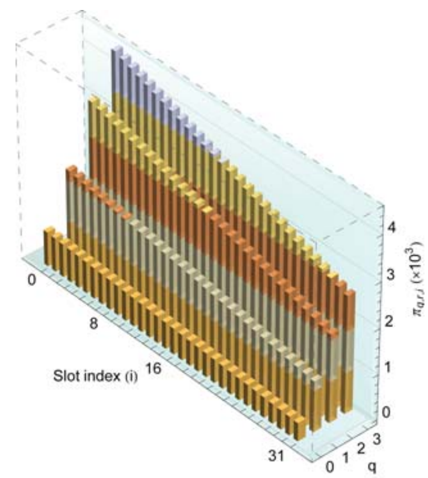
(a) Retry, $r = 0$.



(b) Retry, $r = 1$.



(c) Retry, $r = 2$.



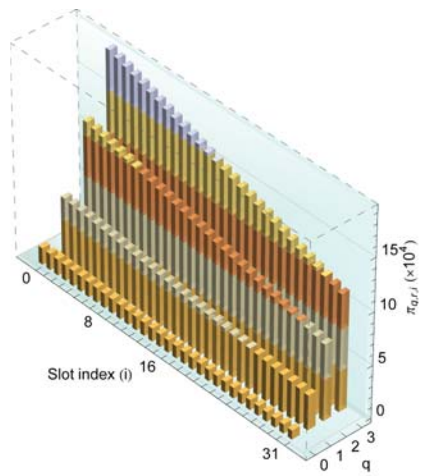
(d) Retry, $r = 3$.

Figure 4.5: *Gaining control* – $AIFSN$ driven arbitration performance by comparing steady istate distributions of one process per class, at various rc values. Configuration: $\mathcal{A}^a \ni \mathbf{a} = 4.3a/2$ and $\mathcal{W}^a \ni \mathbf{w} = 4.3b/1$. Please note the changing range and different scaling of $\pi_{q,r,b}$ axis, along with q axis order swapped.

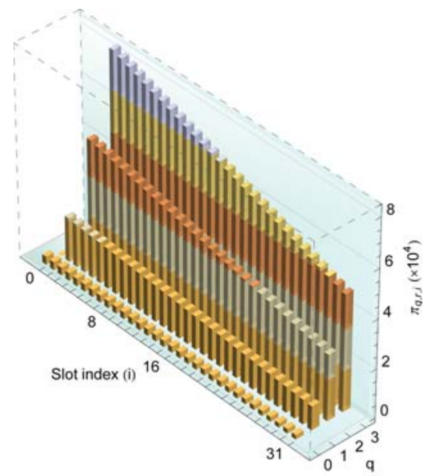
4.7.1.2 Steady state access dependency

The success probability, ρ_k in (4.15) is determined for all network configurations in $\mathcal{A}^a \times \mathcal{W}^a$ and plotted in Figure 4.6. Following observations are pinned down:

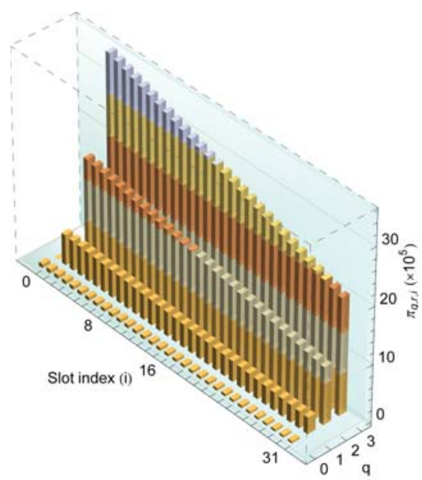
- The probability gap between higher and lower priority classes is increasing with $AIFSN$ separation, as evident from settings along the $AIFSN$ ($\mathbf{a} \in \mathcal{A}^a$) axis. It is intuitively explained as due to different number of deferral slots, during the contention period. This translates into a delayed permission to update the backoff counter for a larger deferral, and vice versa. Consequently higher priority classes win the MAC system cycles more



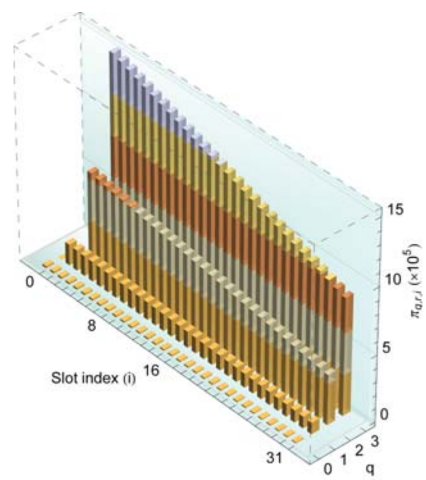
(e) Retry, $r = 4$.



(f) Retry, $r = 5$.



(g) Retry, $r = 6$.



(h) Retry, $r = 7$.

Figure 4.5: ⋯ continued.

often.

- The performance impact due to $AIFSN$ advantage has a strong relationship with the CW settings. This advantage is decreasing along the CW ($\mathbf{w} \in \mathcal{W}^a$) axis. This owes to the fact that higher CW settings lowers the per slot access probability. Resultantly, the $AIFSN$ advantage over the protected slots is decreased.

4.7.1.3 Steady state access distribution

We also zoom in on the success probability of each process as distributed over different accessible slots. This amounts to presenting the individual slot probabilities that are summed

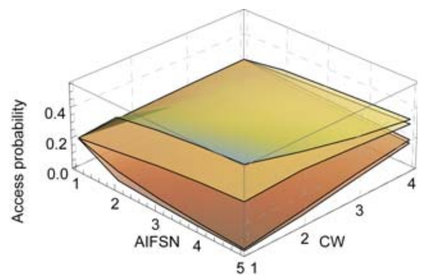


Figure 4.6: *Gaining control* – $AIFSN$ driven arbitration performance by comparing steady state access probability with change of network configurations. Configurations: $\mathcal{A}^a \times \mathcal{W}^a$, with $AIFSN$ and CW axes tick labels showing the row numbers in Table 4.3a and Table 4.3b, respectively.

up in (4.15). For this purpose, the network is configured with the same setting as used for deriving the istate steady distribution, i.e. $\mathcal{A}^a \ni \mathbf{a} = 4.3a/2$ and $\mathcal{W}^a \ni \mathbf{w} = 4.3b/1$. Figure 4.7 presents the results, which are explained as follows:

- The disparity in accessible slots after contention for different processes is enforced by their respective $AIFSN$ settings. A decrease in the $AIFS$ based slot protection with class priority is clearly reflected in the plot.

- The steady state access probability of each process declines sharply with increasing slot indices. We assume this trend can be approximated by a geometric distribution — hence, its selection to estimate the initial distribution $\pi^{(1)}_{q,r,b}$ in (4.12). The sharpness declines with lower priority processes. Again this is due to the fact that higher order slots have decreasing competition, and vice versa. As a result, with increasing slot indices, more processes surpass their respective $AIFS$ regions, and join the competition. Consequently the sharpness decline is smoothing out with higher order slots.

- The $AIFSN$ based handicap for lower priority processes restricts them to transmit mostly in higher order slots, which are available to all priority levels. This is underpinned by stating that higher priority processes win over most of the MAC system cycles in their initial slots, so that higher order slots occur less frequently for them. Resultantly, higher order slots are mostly used by lower priority processes only.

4.7.1.4 System convergence performance

In this section, we present a few indicators to assess the convergence performance of Algorithm 6. The network configurations in $\mathcal{A}^a \times \mathcal{W}^a$ are used to generate the results.

System state space size

In general, the size of the system state space, denoted by s, is determined by the CW settings. It is defined as $s = \sum_{k \in \mathcal{K}} \sum_{r=0}^{R_k} w_{k,r}$. The results reported in Figure 4.8a clearly reflect the exponential increase in space requirements of Algorithm 6 with growing CW.

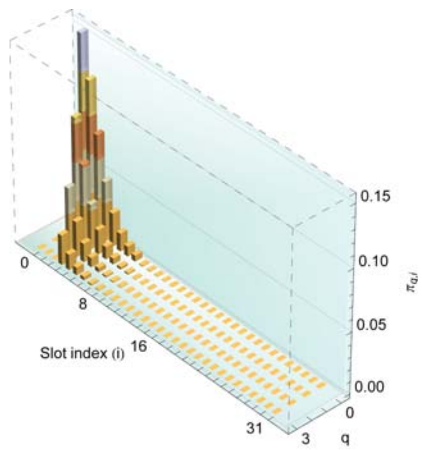


Figure 4.7: *Gaining control* – $AIFSN$ driven arbitration performance using steady state access distribution over various slot indices. Configuration: $\mathcal{A}^a \ni \mathbf{a} = 4.3a/2$ and $\mathcal{W}^a \ni \mathbf{w} = 4.3b/1$.

Number of iterations

The number of iterations taken to converge the system is reported in Figure 4.8b. Two observations are marked here. One, that the number of iterations is strongly decreasing with CW. Higher CW settings increase the standard deviation of $\alpha(k, b)$ distribution in (4.10). This observation can be associated to faster convergence rate. Two, that varying the $AIFSN$ separation does not influence the number of iterations much, especially at relatively high CW settings. Interestingly, maximum iterations are required in a classless case at the lowest CW setting. This is again due to slow convergence rate.

Time for convergence

The convergence time leverages the feasibility study for dynamic system configuration during an operational network. This feature enables QoS requirements renegotiations during admission control and channel capacity changes. Figure 4.8c presents the results. It is clear on the onset that the time of convergence, stays constant per iteration, remains almost independent of an $AIFSN$ separation and strongly increases with CW. For all CW settings, the classless/uniform cases are the slowest to converge. A polynomial of system state space size, s, is given to express the dependence of time in CW

$$\texttt{time}\,(s) = 21.172 - 0.00879702s + 3.1036 \times 10^{-6}s^2 + 8.90389 \times 10^{-12}s^3. \qquad (4.28)$$

This polynomial allows us to determine time for convergence on another hardware platform, $\texttt{time}\,(s) \cdot \frac{p}{p'}$, where p and p' are speeds of processors used to derive our results and the target processor, respectively.

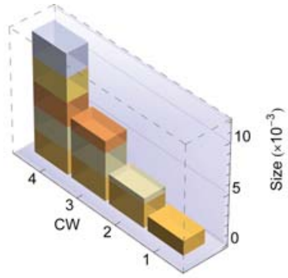
Average time per state

This metric is the first order differential of `time` (s) with respect to s. The results are shown in Figure 4.8d. The observed trend follows the pattern of convergence time (cf. Figure 4.8c) — the average time per state increases with CW, while it is almost independent of an $AIFSN$ separation.

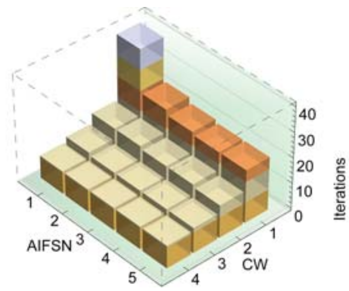
Average time per iteration

This metric is the first order differential of `time` (s) with respect to the number of iterations, `iteration` (s), which is considered to be dependent on the system state size, s, though its polynomial of s is not presented here. We use it to detect if the time for convergence and the number of iterations are linearly related. The results reported in Figure 4.8e denies such a relationship. This average increases with CW, while it remains almost independent of an $AIFSN$ separation. It appears that high state space requirements render the execution of Algorithm 6 slow on the chosen platform.

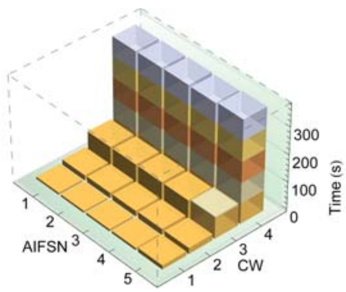
Most of the reported results indicate infeasibility of determining system stable states during an operational network. Nevertheless, in some situations the results can be used to achieve time restricted execution of Algorithm 6, in which case it would be called off prematurely by relaxing the convergence criteria. A tradeoff can be made between allocated time and amount of prematurity with respect to achieving system stable state.



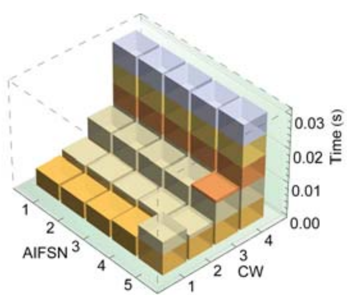
(a) System state space size.



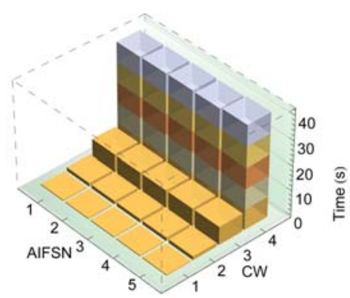
(b) Number of iterations.



(c) Time of convergence.



(d) Average time per state.



(e) Average time per iteration.

Figure 4.8: *Gaining control* – $AIFSN$ driven arbitration performance using convergence indicators. Configurations: $\mathcal{A}^a \times \mathcal{W}^a$, with $AIFSN$ and CW axes tick labels showing the row numbers in Table 4.3a and Table 4.3b, respectively. Please note that the color coding on the bars corresponds to the heights only.

4.7.2 *Gaining control* – CW driven arbitration

Here we illustrate the results derived at system setups in Section 4.6.2.2, which correspond to CW prioritization.

4.7.2.1 Steady istate distribution

A comparison of CW driven scheme against its $AIFSN$ counterpart reveals in general that the probabilities are more evenly distributed over various rc values in the former case. This is due to an absence of the first term on the RHS of (4.2) in the former case, which guarantees $AIFS$ based priority protection. The $\pi_{k,r,b}$ distribution is derived at $\mathcal{W}^w \ni \mathbf{w} = 4.4/2$ and $\mathcal{R}^w \ni \mathbf{R} = [7, 7, 7, 7]$.

Class view

The class view presented in Figure 4.9, shares the general observation as presented in the $AIFSN$ counterpart (cf. Figure 4.4). The highest priority class, i.e., class 0 in both cases are comparable. Their extent of matching depends, indeed on the chosen configurations. The matching is diminishing with decreasing class priority. Further, a comparison of CW and $AIFSN$ class views, leads us to conclude that lower rc values have more probability mass in the $AIFSN$ case than the CW counterpart. For instance, compare $rc = 0$ against $rc > 0$ for all classes in the two cases.

Retry view

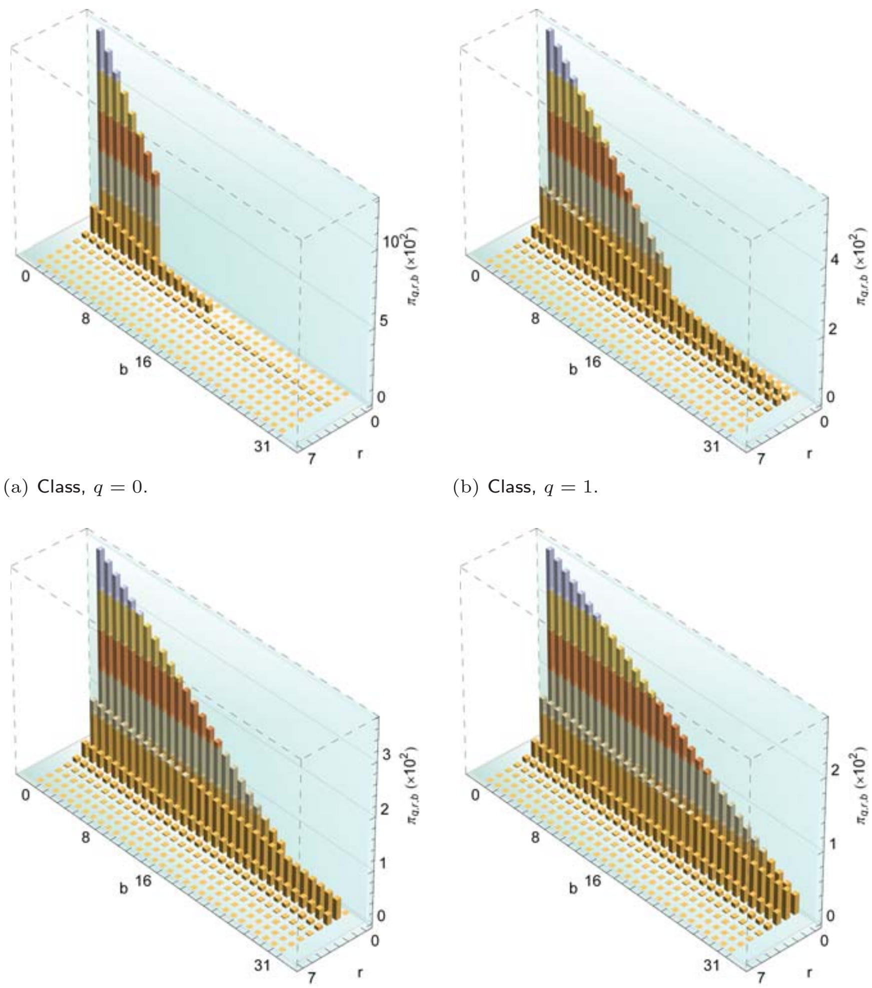
The retry view is given in Figure 4.10. Similar to the $AIFSN$ case, the probability mass is increasing with rc for decreasing class priority. In contrast, the increase is less, since as already noted, the probabilities are more evenly distributed over various rc values. Lastly, the observation that higher priority classes transmit less often than lower priority classes in regions of higher rc values holds in this CW case also, though the disparity between classes is lower than the $AIFSN$ case.

4.7.2.2 Steady state access dependency

The success distribution is determined for all network configurations in $\mathcal{W}^w \times \mathcal{R}^w$. The results are plotted in Figure 4.11. We present the following observations:

- By moving along the CW ($\mathbf{w} \in \mathcal{W}^w$) axis, it is clear that the probability gap in favor of high priority classes is increasing with the CW separation. It is explained by decreasing per slot access probability with CW, so that high priority classes win more MAC cycles.

- The CW arbitration performance is evaluated at various system contention levels by varying $RetryLimit$ values. The probabilities of all classes are increasing with the $RetryLimit$. It is due to the fact that for a given CW series, per slot congestion decreases with $RetryLimit$ — hence, fewer collisions. Along the $RetryLimit$ ($\mathbf{R} \in \mathcal{R}^w$) axis, the system achieves optimality, which favors all processes.



(a) Class, $q = 0$.

(b) Class, $q = 1$.

(c) Class, $q = 2$.

(d) Class, $q = 3$.

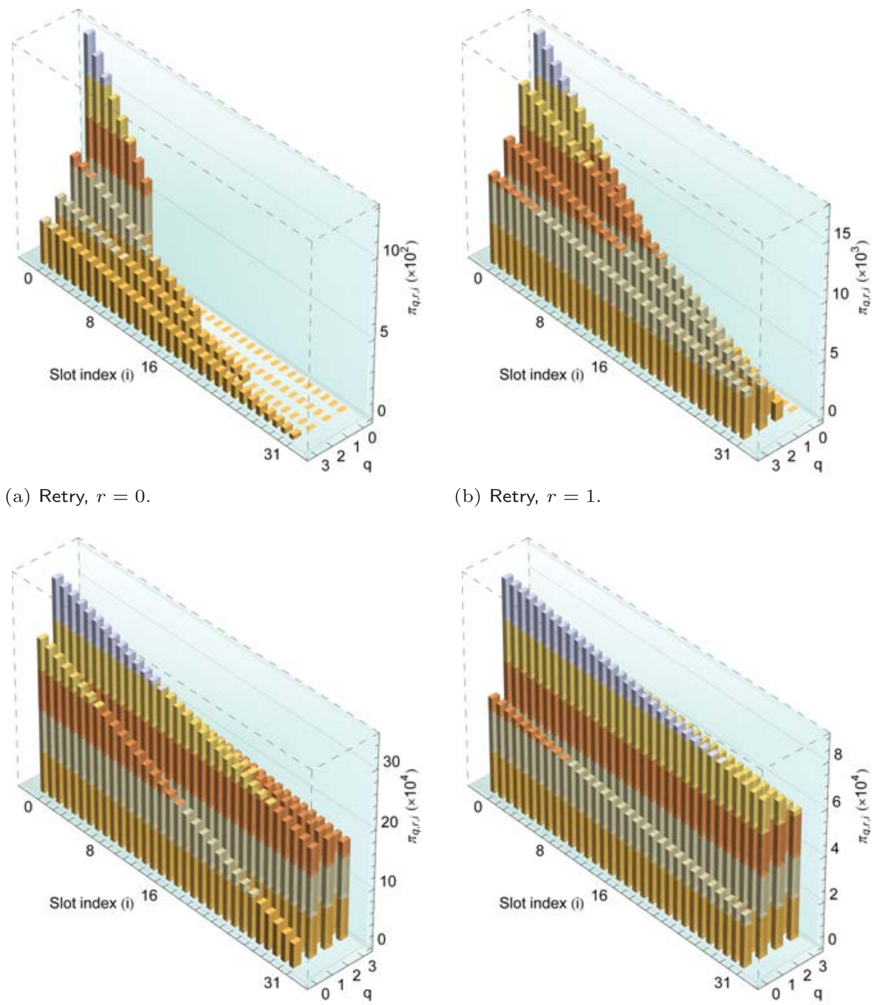
Figure 4.9: *Gaining control – CW* driven arbitration performance using stationary istate distribution, $\pi_{k,r,b}$. Configuration: $\mathcal{W}^w \ni \mathbf{w} = 4.4/2$ and $\mathcal{R}^w \ni \mathbf{R} = [7,7,7,7]$. Please note the changing range of graphs.

4.7.2.3 Steady state access distribution

By zooming in on the same network configuration that is used for deriving the steady istate distribution, i.e. $\mathcal{W}^w \ni \mathbf{w} = 4.4/2$ and $\mathcal{R}^w \ni \mathbf{R} = [7,7,7,7]$, we present in Figure 4.12, the steady state access distribution over different accessible slots. The results are as follows:

- All low indexed slots are accessible to all classes, as there are no prohibited slots. This poses more competition over these slots, resulting in lower probability values as compared to the $AIFSN$ case.

- There is a noticeable jump immediately after the slots available at $rc = 0$, i.e. when slot index $i > 7$. The size of this jump is decreasing with lower priority classes. This jump



(a) Retry, $r = 0$.

(b) Retry, $r = 1$.

(c) Retry, $r = 2$.

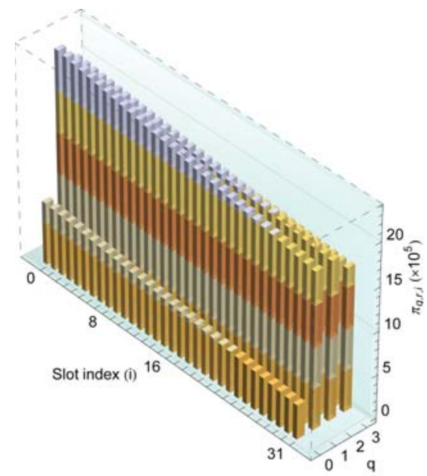
(d) Retry, $r = 3$.

Figure 4.10: *Gaining control* – CW driven arbitration performance by comparing stationary istate distributions one process per class, at various rc values. Configuration: $\mathcal{W}^w \ni \mathbf{w} = 4.4/2$ and $\mathcal{R}^w \ni \mathbf{R} = [7, 7, 7, 7]$. Please note the changing range and different scaling of $\pi_{q,r,b}$ axis, along with q axis order swapped.

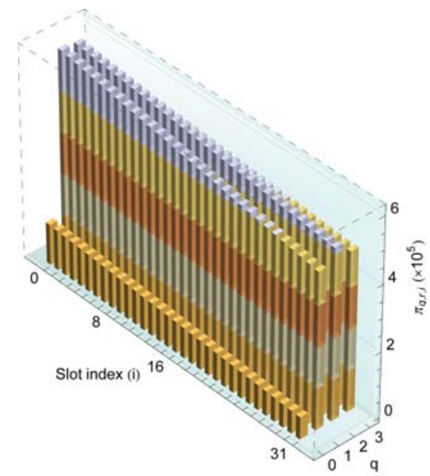
is also observed in the $AIFSN$ case, though at a much lower scale. It is due to the fact that MAC cycles are won mostly at $rc = 0$. In general, probability jumps are expected at slot boundaries of consecutive rc values.

4.7.2.4 System convergence performance

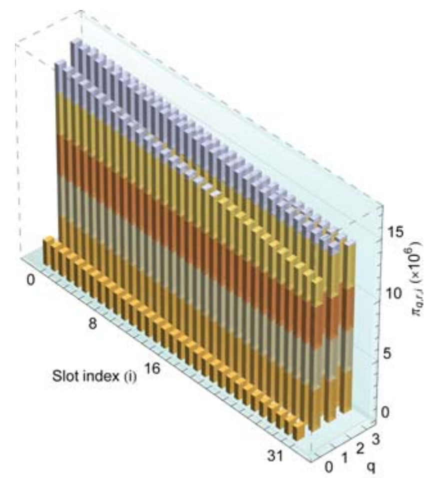
The performance indicators of Algorithm 6 for the CW case are derived using all network configurations in $\mathcal{W}^w \times \mathcal{R}^w$.



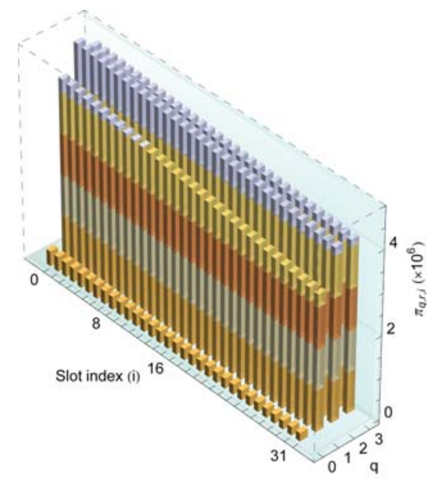
(e) Retry, $r = 4$.



(f) Retry, $r = 5$.



(g) Retry, $r = 6$.



(h) Retry, $r = 7$.

Figure 4.10: $\cdots$ continued.

System state space size

The results for the system state space size are reported in Figure 4.13a. The highest space requirement of the CW case is still lower than the similar measure of the $AIFSN$ case due to all processes having higher CW settings in the latter.

Number of iterations

The results are reported in Figure 4.13b. We make two observations here. One, that the system requires highest number of iterations at the classless settings. It is due to slow convergence rate of Algorithm 6. Two, that the number of iterations is increasing with the

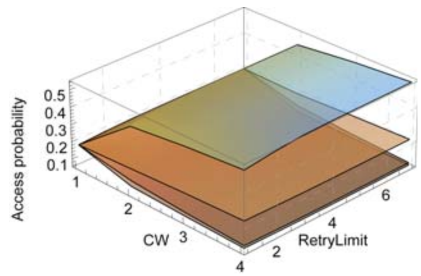


Figure 4.11: *Gaining control* – CW driven arbitration performance by comparing steady state access probability with change of network configurations. Configurations: $\mathcal{W}^w \times \mathcal{R}^w$, with CW and $RetryLimit$ axes tick labels showing the row numbers in Table 4.4 and actual values that are uniform across the board, respectively.

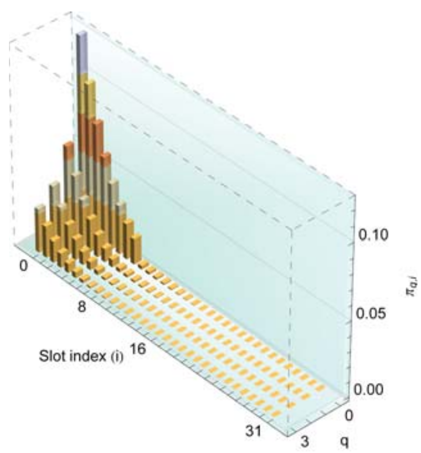


Figure 4.12: *Gaining control* – CW driven arbitration performance using steady state access distribution over various slot indices. Configuration: $\mathcal{W}^w \ni \mathbf{w} = 4.4/2$ and $\mathcal{R}^w \ni \mathbf{R} = [7, 7, 7, 7]$.

$RetryLimit$, with peak value at $RetryLimit = 7$, where per the slot congestion is lowest.

Time for convergence

The results reported in Figure 4.13c indicate a slower convergence rate than the $AIFSN$ case. Time polynomial of system state space size, s, is

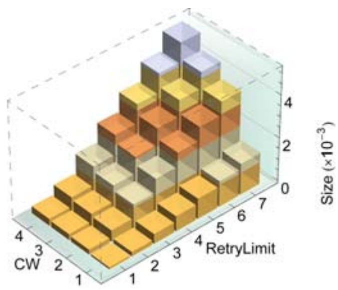
$$\begin{aligned} \texttt{time}(s) = & \, 0.815038 - 0.0154836s + 0.000106252s^2 - 2.18588 \times 10^{-7}s^3 \\ & +2.39208 \times 10^{-10}s^4 - 1.30432 \times 10^{-13}s^5 + 2.77185 \times 10^{-17}s^6. \end{aligned} \tag{4.29}$$

Average time per state

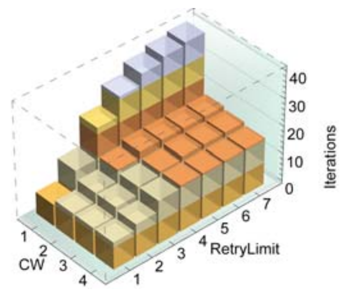
The results are reported in Figure 4.13d.

Average time per iteration

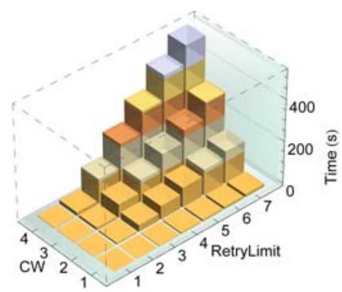
The results are reported in Figure 4.13e.



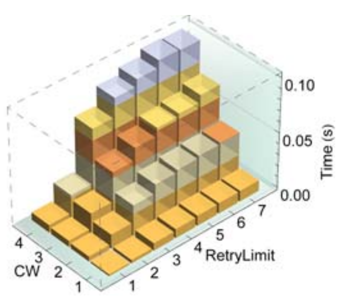
(a) System state space size.



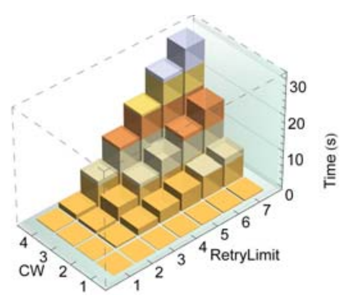
(b) Number of iterations.



(c) Time of convergence.



(d) Average time per state.



(e) Average time per iteration.

Figure 4.13: *Gaining control* – CW driven arbitration performance: convergence indicators. Configuration: $\mathcal{W}^w \times \mathcal{R}^w$, with CW and $RetryLimit$ axes tick labels showing the row numbers in Table 4.4 and actual values that are uniform across the board, respectively.

4.7.3 *Gaining control* – default EDCA parameter set driven arbitration

In this section, we derive the results at system setups given in Section 4.6.2.4 in the context of *gaining control*.

4.7.3.1 Steady istate distribution

The four classes in default EDCA parameter settings prioritize on $AIFSN$ and CW together. In general, the disparity between extreme end classes is high. The $\pi_{k,r,b}$ distribution is derived using the settings in Table 4.6 and Table 4.7, along with $\mathcal{R}^d \ni \mathbf{R} = [7, 7, 7, 7]$ and $\mathcal{N}^d \ni \mathbf{n} = [1, 1, 1, 1]$.

Class view

The class view is shown in Figure 4.14. The disparity between retry values is increasing with the class priority.

Retry view

The retry view is presented in Figure 4.15. The highest priority class wins MAC system cycles mostly during $rc \leqslant 2$ due to a very low competition from the competing network on the low order slots. Resultantly, it has minimal transmissions beyond $rc > 2$.

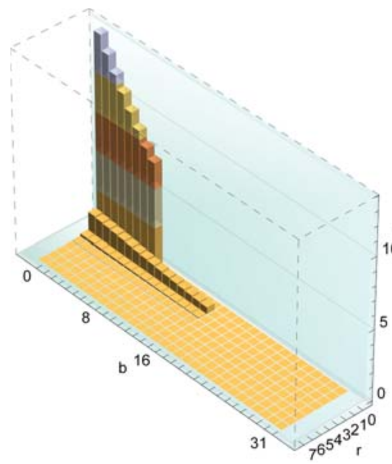
The disparity between extreme end classes appears rather high. We highlight it by comparing dispersion of their retry distributions. The dispersion itself is measured by standard deviation (SD). The probability of finding a process at various retry values is given by $\mathtt{P}(rc_q = r) = \sum_{b=0}^{w_{q,r}} \pi_{q,r,b}$. The SD vector of the retry distributions per class is $[0.288839, 0.23474, 0.149422, 0.0634485]$, which confirms the observation. The results plotted in Figure 4.16, reflect that dispersion of the retry distribution is increasing with the class priority. The behavior resembles a negative exponential distribution (ned), so that the curve sharpness can be represented by the rate parameter in the ned, which is increasing with the class priority. Interestingly, the retry distributions have almost the same means — see the mean vector per class: $[0.124933, 0.125, 0.125, 0.125]$. Lastly, we also notice that the SD values of the current case remain higher than the $AIFSN$ and the CW cases.

4.7.3.2 Steady state access dependency

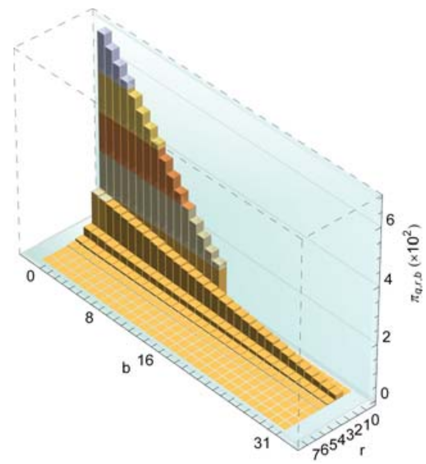
This distribution, derived for settings in $\mathcal{N}^d \times \mathcal{R}^d$, is plotted in Figure 4.17. The impact on probability values is described as follows.

- Per class probability is decreasing along the number of processes per class axis $(\mathbf{n} \in \mathcal{N}^d)$. Intuitively, time sharing of the available system capacity is increasing with the number of processes — thus more competition on each accessible slot, which increases collisions, giving rise to the observed trend.

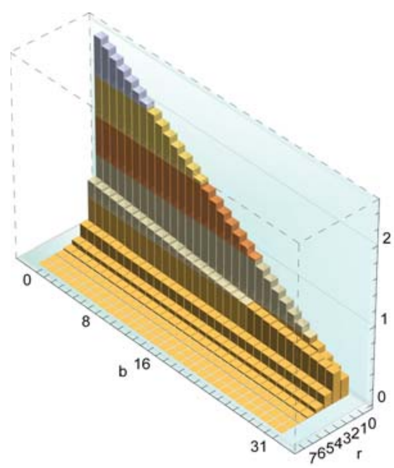
- Along the $RetryLimit$ axis $(\mathbf{R} \in \mathcal{R}^d)$, we observe a positive impact. For a given number of processes per class, increasing the $RetryLimit$ lessens the system competition per slot, thus triggers fewer collisions. Effectively the system seeks optimal success probabilities for all processes.



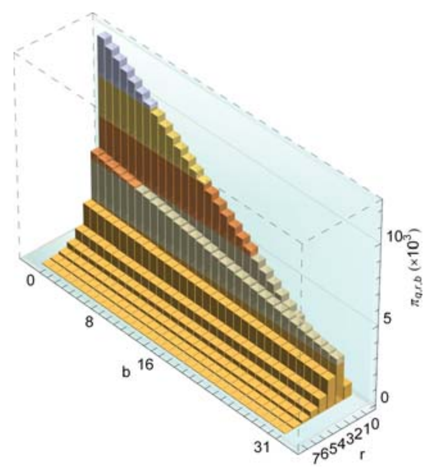
(a) Class, $q = 0$.



(b) Class, $q = 1$.

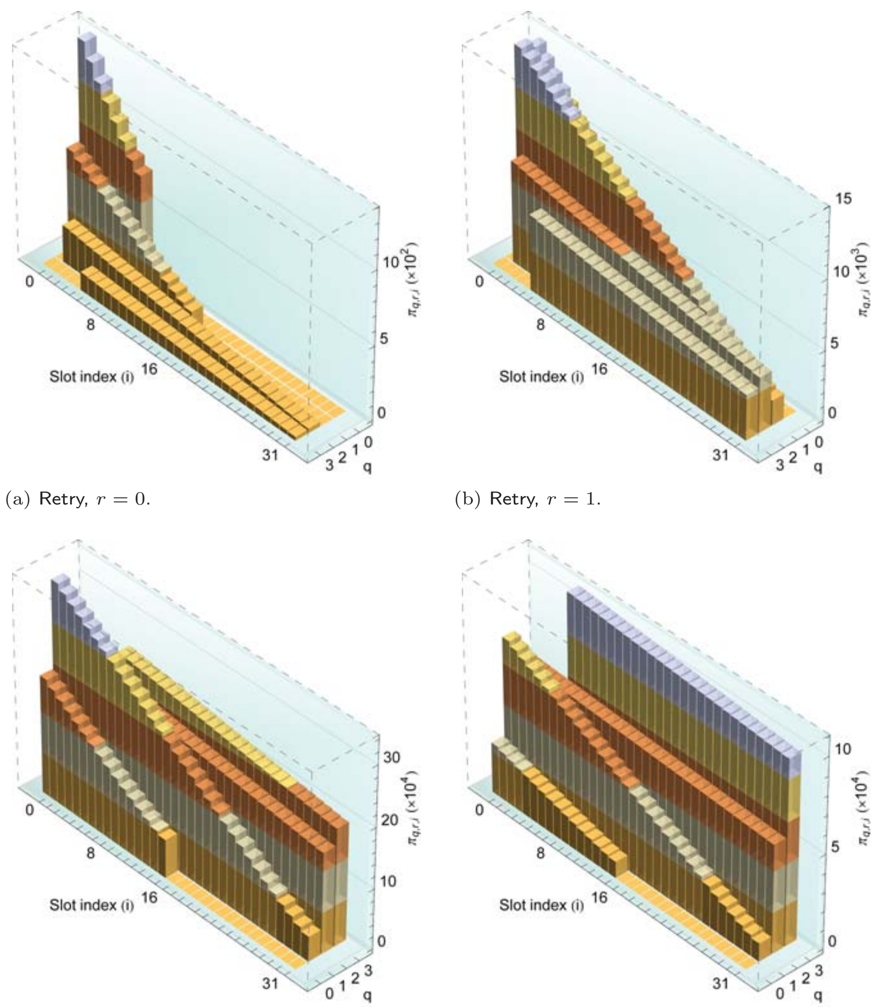


(c) Class, $q = 2$.



(d) Class, $q = 3$.

Figure 4.14: *Gaining control* – default EDCA parameter set driven arbitration performance using stationary distribution $\pi_{k,r,b}$. Configuration: $\mathcal{R}^d \ni \mathbf{R} = [7, 7, 7, 7]$ and $\mathcal{N}^d \ni \mathbf{n} = [1, 1, 1, 1]$. Please note the changing range and different scaling of $\pi_{q,r,b}$ axis.



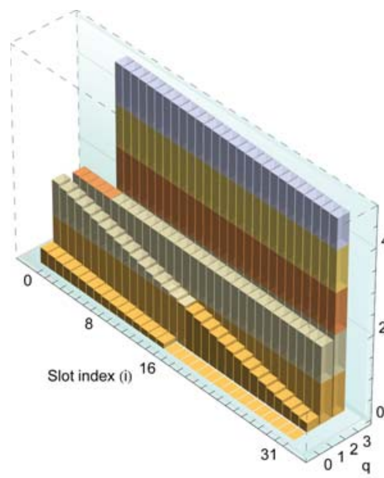
(a) Retry, $r = 0$.

(b) Retry, $r = 1$.

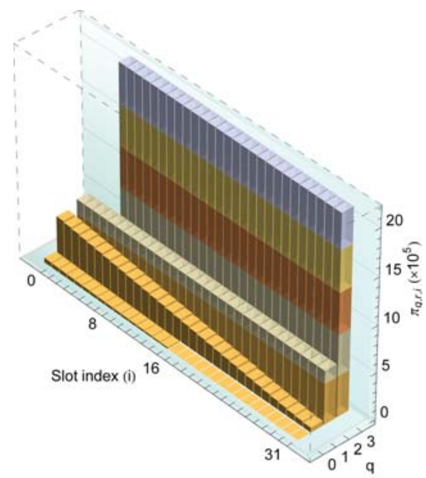
(c) Retry, $r = 2$.

(d) Retry, $r = 3$.

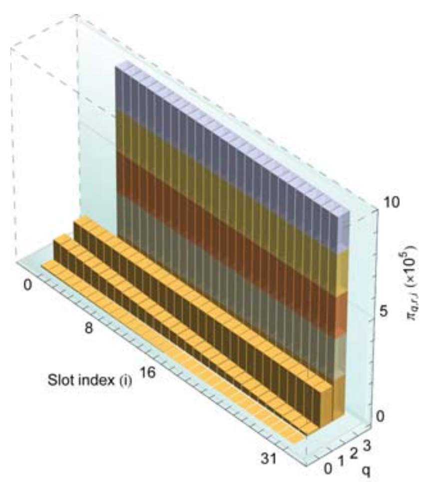
Figure 4.15: *Gaining control* – default EDCA parameter set driven arbitration performance by comparing stationary distributions of processes at various retry counter values. Configuration: $\mathcal{R}^d \ni \mathbf{R} = [7, 7, 7, 7]$ and $\mathcal{N}^d \ni \mathbf{n} = [1, 1, 1, 1]$. Please note the changing range and different scaling of $\pi_{q,r,b}$ axis, along with q axis order swapped.



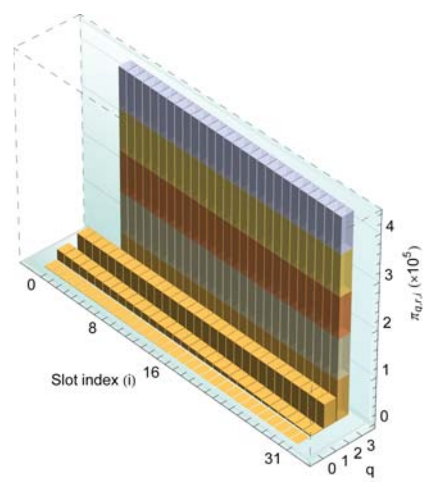
(e) Retry, $r = 4$.



(f) Retry, $r = 5$.



(g) Retry, $r = 6$.



(h) Retry, $r = 7$.

Figure 4.15: $\cdots$ continued.

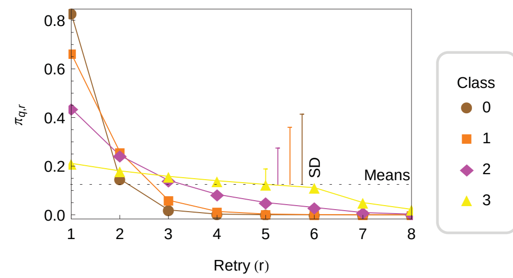


Figure 4.16: *Gaining control* – default EDCA parameter set driven performance — highlighting extreme disparity between class settings by comparing dispersion of their retry distributions. Dotted lines coincide, showing the means of retry distributions are same for classes. The standard deviations of the retry distributions, are shown only above the mean lines. Configuration: $\mathcal{R}^d \ni \mathbf{R} = [7, 7, 7, 7]$ and $\mathcal{N}^d \ni \mathbf{n} = [1, 1, 1, 1]$.

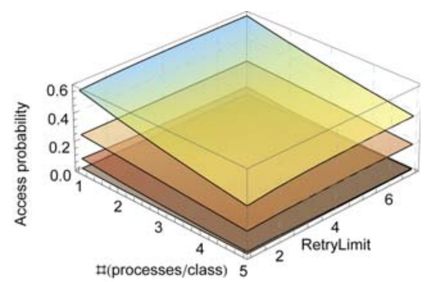


Figure 4.17: *Gaining control* – default EDCA parameter set driven arbitration performance by comparing steady state access probability with change of network configurations. Configurations: $\mathcal{N}^d \times \mathcal{R}^d$, with #(processes/class) and $RetryLimit$ axes tick labels showing the actual values that are uniform across the board, respectively.

4.7.3.3 Steady state access distribution

We also zoom in on the same network configuration as used for deriving the steady istate distribution, $\mathcal{R}^d \ni \mathbf{R} = [7, 7, 7, 7]$ and $\mathcal{N}^d \ni \mathbf{n} = [1, 1, 1, 1]$, to present the steady state access distribution over different accessible slots in Figure 4.18. The drop down of probabilities immediately after passing the slots available at $rc = 0$ is increasing with class priority. It is prominent in the highest priority class. Overall, the lowest priority class is starving. Finally, an effect of high disparity between classes is reflected by steady success probability values for all classes becoming zero at a relatively small slot index, 15.

4.7.3.4 System convergence performance

In this case, we observed an interesting behavior regarding convergence of Algorithm 6. For some network configurations, the convergence criterion, $\epsilon = 10^{-10}$, is found too strict to achieve the fixed point. After few iterations, the system shows contraction and expensiveness alternately, thus bouncing back and forth around the fixed point. Consequently, the convergence criterion has to be relaxed to $\epsilon = 10^{-8}$. Following three network configurations offer this behavior:

- $\mathcal{R}^d \ni \mathbf{R} = [1, 1, 1, 1]$ and $\mathcal{N}^d \ni \mathbf{n} = [4, 4, 4, 4]$;

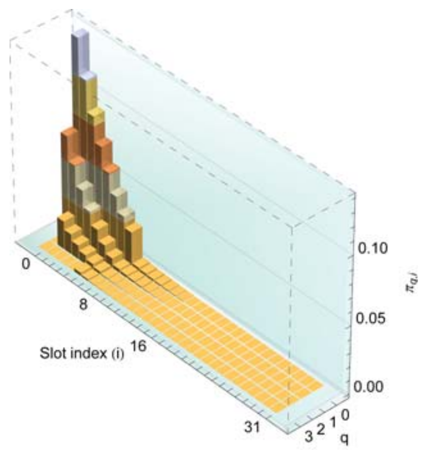


Figure 4.18: *Gaining control* – default EDCA parameter set driven arbitration performance using steady state access distribution over various slot indices. Configuration: $\mathcal{R}^d \ni \mathbf{R} = [7, 7, 7, 7]$ and $\mathcal{N}^d \ni \mathbf{n} = [1, 1, 1, 1]$.

- $\mathcal{R}^d \ni \mathbf{R} = [2, 2, 2, 2]$ and $\mathcal{N}^d \ni \mathbf{n} = [4, 4, 4, 4]$;
- $\mathcal{R}^d \ni \mathbf{R} = [3, 3, 3, 3]$ and $\mathcal{N}^d \ni \mathbf{n} = [5, 5, 5, 5]$.

The performance indicators of Algorithm 6 for settings in $\mathcal{N}^d \times \mathcal{R}^d$ are presented.

System state space size

The results for the system state space size are reported in Figure 4.19a.

Number of iterations

The results presented in Figure 4.19b, indicate an increasing trend with the number of processes per class and the *RetryLimit*, though there are some outliers.

Time for convergence

The results presented in Figure 4.19c, indicate an increasing trend with the number of processes per class and the *RetryLimit*. Time polynomial of system state space size, s, is

$$\begin{aligned}\texttt{time}\,(s) = {} & 1.7555 - 0.0085884s + 0.0000285282s^2 - 9.91715 \times 10^{-9}s^3 \\ & +4.5999 \times 10^{-12}s^4 - 8.05464 \times 10^{-16}s^5 + 4.4398 \times 10^{-20}s^6.\end{aligned} \tag{4.30}$$

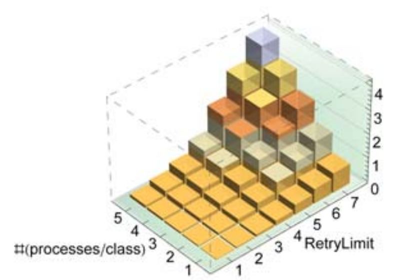
Average time per state

The results presented in Figure 4.19d, indicate a decreasing trend with the number of processes per class, while an increasing trend with the *RetryLimit*. The first observation is similar to the time behavior, while the second is due to the exponential increase of state space size with the number of processes per class. Time of convergence and the state space size are not linearly related.

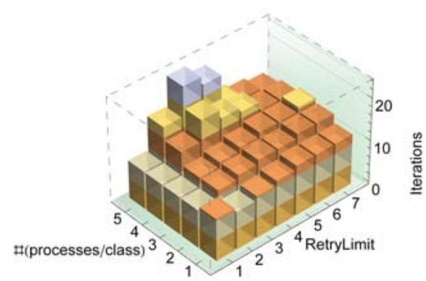
Average time per iteration

The results presented in Figure 4.19e, indicate a uniform trend with the number of processes per class, while an increasing trend with the *RetryLimit*. Time of convergence and the state space size are not linearly related.

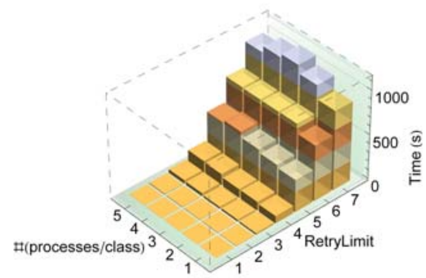
We leave this section by concluding the model for the network steady state and its detailed performance.



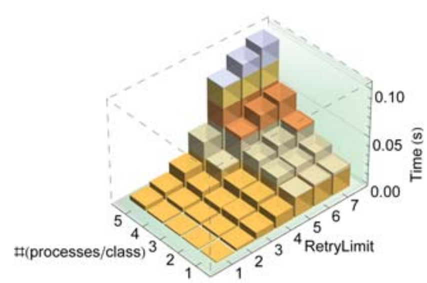
(a) System state space size.



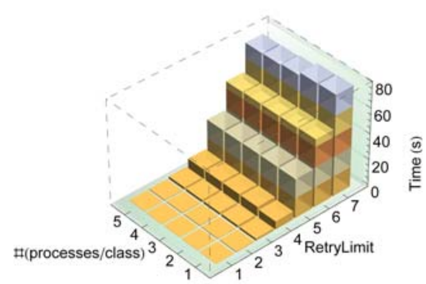
(b) Number of iterations.



(c) Time of convergence.



(d) Average time per state.



(e) Average time per iteration.

Figure 4.19: *Gaining control* – default EDCA parameter set driven arbitration performance: convergence indicators. Configuration: $\mathcal{N}^d \times \mathcal{R}^d$, with #(processes/class) and $RetryLimit$ axes tick labels showing the actual values that are uniform across the board, respectively.

4.8 Models for the QoS properties

From here onwards, we set off to address research **Question 4**. The presented model of process istate distribution reflects closely on process istate performance. The network level properties, corresponding to QoS delivery in our schema for the QoS problem (cf. Figure 2.7), still need to be determined for our system. Therefore, in this section, we derive models for the QoS properties. The analysis is based on already identified two major concerns, which abstract the MAC operation on separate dimensions.

4.8.1 Expected channel access frequency

The expected channel access frequency, ν_k, is defined as the average number of accesses made by a process in one second. By noting that process k accesses the MAC system cycle of expected service time, $\mho$ with probability ρ_k (refers (4.16) and (4.15)), the expected channel access frequency is

$$\nu_k = \frac{\rho_k}{\mho}. \tag{4.31}$$

It corresponds to the partial function, $\mathtt{f}_k^{\nu}(\mathbf{p})$ in (2.16).

4.8.2 Expected channel share

The expected channel share, θ_k, is defined as the expected data transmitted during the MAC system cycle. In this duration, x_k transmission transactions are completed, with each one transmitting L_k bytes (refers (4.21) and (4.20)). Hence, the expected channel share is

$$\theta_k = \nu_k \left(8 \cdot L_k \cdot x_k\right). \tag{4.32}$$

It corresponds to the partial function, $\mathtt{f}_k^{\theta}(\mathbf{p})$ in (2.16).

4.8.3 Expected MAC latency

The stringent QoE requirements from a streaming service are based on how well the spatial/temporal requirements of all frames are met. At the time of its creation, each frame has a deadline. Typically at the UDP layer, it is fragmented before relayed to the MAC layer, where the frame deadline is translated into deadlines of the corresponding MPDUs. To avoid degradation, the MAC QoS control should meet deadlines of all MPDUs of a given frame. For our model that considers multiple MPDUs in each transmission opportunity, this translates into a deadline of each transmission opportunity, where only the leading MPDU contends. A system which allows infinite retransmissions, will eventually see all MPDUs reaching the destination. For a finite case, some leading MPDUs are dropped, whose lifetime should not be considered for the latency calculation of the transmission opportunities. An incorrect calculation will derive increased latency, which may falsely consider some MPDUs stale. This pitfall due to oversimplification in latency calculation needs to be avoided. The desired correct calculation is elucidated in Figure 4.20.

Next, we model a finite *RetryLimit* case for lifetime of only successful MPDUs [41]. The expected MAC latency, λ_k, is defined as the average channel contention delay and transmission opportunity time of successful MPDUs that comprises multiple transmission transactions (cf. Definition 2.7.1). It corresponds to the partial function, $\mathtt{f}_k^{\lambda}(\mathbf{p})$ in (2.16).

Consider a system of class q, HoL MPDUs, which start successful transmission opportunities. The expected combined channel access frequency of all processes within this system

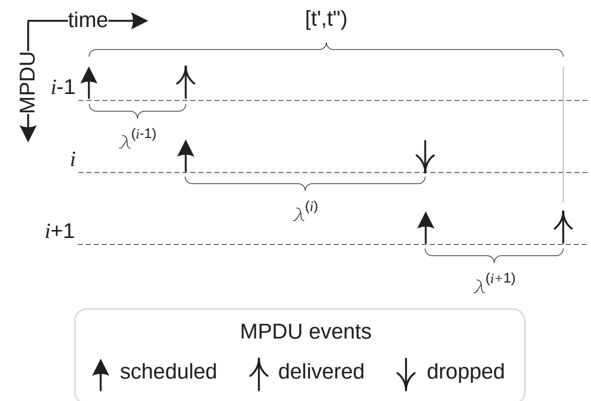


Figure 4.20: Expected MAC latency model for a transmission opportunity of one MPDU. Incorrect latency, λ, calculation over an interval $\left[t', t''\right)$, $\lambda = \frac{t''-t'}{2}$, is avoided by eliminating the lifetime of the dropped MPDU — thus $\lambda = \frac{\lambda^{(i-1)}+\lambda^{(i+1)}}{2}$ considers the latencies of only $(i-1)^{\mathrm{th}}$ and $(i+1)^{\mathrm{th}}$ MPDUs.

is given by $\nu_q = n_q \cdot \nu_k$, where process k belongs to class q. ν_q is also the rate of successful transmission opportunities for class q. Applying Little's theorem, which states that the long-term average number of customers in a stable system is equal to the long-term average effective arrival rate times the average time a customer spends in the system, we express the expected MAC latency, λ_k, as

$$n_q = \lambda_k \cdot \nu_q \ . \tag{4.33}$$

Since all MPDUs may not get through, let ρ_k^{d} be the dropout probability of an arbitrary leading MPDU, then for our system, we rewrite (4.33) as

$$n_q \left(1 - \rho_k^{\mathrm{d}}\right) = \lambda_k \cdot \nu_q \ . \tag{4.34}$$

Using $\nu_q = n_q \cdot \nu_k$ and after simplification,

$$\lambda_k = \frac{1 - \rho_k^{\mathrm{d}}}{\nu_k} \ . \tag{4.35}$$

In order to be dropped out, a fresh HoL MPDU has to undergo $RetryLimit_k + 1$ collisions. At a random time, an MPDU may be found in an arbitrary istate (r, b), thus requires $R_q - r + 1$ collisions to be dropped. While at istate $(r, b)\,, 0 \leqslant b \leqslant w_{k,r}$, the process reacts to a collision by jumping to an istate $(r+1, b')\,, 0 \leqslant b' \leqslant w_{k,r+1}$ with probability $\frac{1}{w_{k,r+1}}$. On subsequent collisions, this action is repeated till the process has $rc_k = RetryLimit_k$, after which the MPDU is dropped. Hence, starting from istate (r, b), these jumps define a path through process istates, which we term as a collision path, p. The length of p is equal to the number of collisions required for a dropout, $R_q - r + 1$. The probability of the MPDU traversing p depends on the istates that compose p. This probability is denoted by $\mathbb{P}(p)$. Given an istate, (r, b), let $\mathcal{P}_{k,r,b}$ denote the set of all possible paths starting from it. The probability of a dropout from this starting istate, $\tilde{\rho}^{\mathrm{d}}_{k,r,b}$, is given by the summation over all paths in $\mathcal{P}_{k,r,b}$, the probability of the MPDU traversing them. Hence,

$$\tilde{\rho}^{\mathrm{d}}_{k,r,b} = \sum_{p \in \mathcal{P}_{k,r,b}} \mathtt{P}(p) \ . \tag{4.36}$$

In order to determine the dropout probability, all arbitrary starting istates need to be considered. Hence, ρ^{d}_k is given by the summation over all arbitrary starting istates, the probability of a dropout from there. Thus

$$\rho^{\mathrm{d}}_k = \sum_{r=0}^{R_k} \sum_{b=0}^{w_{k,r}} \tilde{\rho}^{\mathrm{d}}_{k,r,b} \ . \tag{4.37}$$

$\mathcal{P}_{k,r,b}$ is formed by taking the $(R_k - r)$-ary cartesian product of sets of istates with the same rc_k value,

$$\mathcal{P}_{k,r,b} = \{(r,b)\} \times \{(r+1,0), \cdots, (r+1, w_{k,r+1})\} \times \cdots \times \{(R_k,0), \cdots, (R_k, w_{k,R_k})\} \ . \tag{4.38}$$

Thus $p \in \mathcal{P}_{k,r,b}$ is represented by an $(R_k - r)$-tuple of istates, with $p.i, 1 \leqslant i \leqslant R_k - r$ being the istate at $rc_k = r + i - 1$. The elements of the istate, $p.i$, are accessed by a double indirection: $(p.i.r + i - 1, p.i.b), 0 \leqslant b \leqslant w_{k,r+i-1}$.

Now we traverse p to derive $\mathtt{P}(p)$. Consider $p = ((r,b), (r+1,b'), \cdots, (R_k, b''))$. The probability of first jump in p is given by

$$\mathtt{P}\left(p^{(1)}\right) = \pi_{k,r,b} \cdot \gamma(k,b) \cdot \frac{1}{w_{k,r+1}}, \tag{4.39}$$

where the second factor gives the probability of a transmission in the same slot, from the competing network to ensure a collision (cf. Section 4.3). Generalizing to the i^{th} jump,

$$\mathtt{P}\left(p^{(i)}\right) = \pi_{k,p.i.r+i-1,p.i.b} \cdot \gamma(k,b) \cdot \frac{1}{w_{k,p.i.r+i}} \ . \tag{4.40}$$

$\mathtt{P}(p)$ is, therefore, expressed as a product of all jumps in p to reach $rc_k = R_k$ times the last jump to be dropped:

$$\mathtt{P}(p) = \left[\prod_{i=1}^{R_k - r + 1} \mathtt{P}\left(p^{(i)}\right) \right] \cdot w_{k,0} \ . \tag{4.41}$$

By noting that $R_k + 1 \equiv 0$, the right most factor cancels out the last redundant transition to an istate at $rc_k = 0$, which is not required for a dropout case.

4.8.4 Expected MAC reliability

The expected MAC reliability, ω_k, is defined as the expected number of delivered MPDUs normalized by total MPDUs. This is already expressed by the second factor in numerator of (4.35), hence

$$\omega_k = 1 - \rho^{\mathrm{d}}_k \ . \tag{4.42}$$

It corresponds to the partial function, $\mathtt{f}^{\omega}_k(\mathbf{p})$ in (2.16).

4.9 Models for the QoS properties – validation and verification

In this section, we compute the QoS properties using (4.31), (4.32), (4.35) and (4.42) to fill space $\mathcal{A}$ (cf. Section 2.5.2). The analytical results are verified by corresponding simulations, which run for 500 seconds and repeated 20 times in each case before assuming their average behavior. 95% of the simulations fall within an error margin of 0.05% of this mean.

The selection of a simulation tool has primarily led us to pick ns2 [199], as it is the most popular tool in the research community, besides being open source. We have added an EDCA patch [194] to it and performed some basic validation tests that give inaccurate results. While the arbitration mechanism of EDCA appears to work correctly, few problems are found. These are related to creating a single collision domain using the existing PHY models available, an absence of payload configuration at the MAC level and disturbing effects of routing packets (similar to the work reported in [173]). The strangest and most problematic observed behavior concerns the creation of many more collisions than in the analytical model, thus reducing channel utilization. A possible explanation for this can be bad random generator performance as also reported in [208]. Moreover the channel utilization variability is much larger than expected. Due to the problems encountered, we have developed a new custom made discrete event simulator, GRADES [5], which is directly based on the EDCA specification. A set of validation experiments is conducted for the default EDCA parameter settings in the general system setup with $\mathbf{x} = [1, 1, 1, 1]$, and MPDU payload of 1000 bytes (cf. Table 4.2). These experiments are repeated 20 times, for both ns2 and GRADES, while their performance is compared with the analytical model. The presented results in Figure 4.21 show time variations of channel utilization, $\sum_{k \in \mathcal{K}} \theta_k$ (cf. (4.32)).

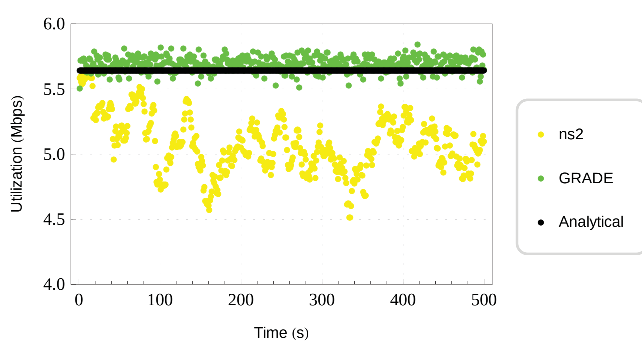


Figure 4.21: Comparing ns2 with GRADES: utilization variability showing average utilization over 1 second intervals. The values show the average (per point) over 20 simulations. The ns2 simulation gives extremely low utilization at times, corresponding to large numbers of collisions. These indicate a high synchrony of the EDCA countdown mechanisms in the different processes, possibly by poor performance of the random number generation.

4.10 QoS fulfillment

This section presents our proposed approach to tackle research **Question 5**. The derived models for QoS properties enable us to address the QoS fulfillment aspect of the QoS problem (cf. Section 2.5.2). As noted, a side effect of the relation, `sat`, can be multiple solutions in the subspace, $\mathcal{N}'(\mathbf{e}) \subseteq \mathcal{N}$ (cf. Figure 2.9). Comparing these solutions by quantifying their 'value', is not trivial in our intractably large and highly complex search space. This section establishes a general mechanism to search for the best solution, termed as a global optimum (GO), from the available solutions.

The solutions in the subspace, $\mathcal{N}'(\mathbf{e})$ has quality properties such as the wastage of channel due to undue idle slots, extra collisions resulting from overly booked slots and the channel utilization. Let $\mathtt{z}_i(\mathbf{p})$ be a function, mapping $\mathbf{p}$ to a quality property i. Define $\mathcal{Z}'(\mathbf{e}) \subseteq \mathcal{Z}$ (cf. Figure 4.22) as the objective subspace obtained by applying functions in $\mathbf{z}(\mathbf{p})$ on elements of $\mathcal{N}'(\mathbf{e})$. For GO, we aim for the solution in $\mathcal{N}'(\mathbf{e})$ that is optimal with respect to values in $\mathcal{Z}'(\mathbf{e})$. Thus our search for GO relies on formulating and solving, what is known as a multiobjective optimization problem (MOP). The solution of the MOP yields GO.

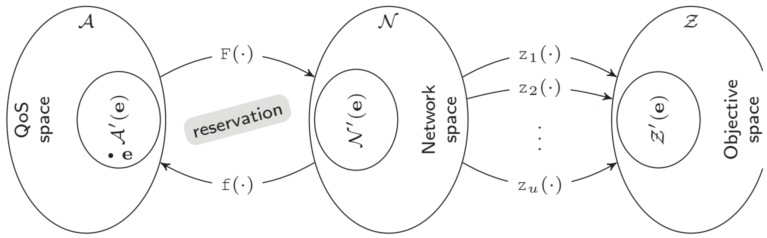


Figure 4.22: A depiction of the objective space for the solution process of the QoS problem.

A MOP is formally defined as,

$$\begin{array}{ll} \text{Minimize} & \mathbf{z}(\mathbf{p})\,, \\ \text{subject to} & \mathbf{c}(\mathbf{p})\,, \end{array} \tag{4.43}$$

where $\mathbf{z}(\mathbf{p}) \triangleq [\mathtt{z}_i(\mathbf{p})]^\intercal_{i\in\mathcal{U}}$ and $\mathbf{c}(\mathbf{p}) \triangleq [\mathtt{c}_j(\mathbf{p})]^\intercal_{j\in\mathcal{V}}$ are vectors of u objective functions and v constraints, respectively, defined on a decision vector of P-tuple, $\mathbf{p} \in \mathcal{N}$, where $\mathcal{U} \triangleq \{1, 2, \cdots, u\}$ and $\mathcal{V} \triangleq \{1, 2, \cdots, v\}$. For the current purpose, a single constraint container suffices:

$$\mathtt{c}_1(\mathbf{p}) \triangleq \mathtt{f}(\mathbf{p}) \in \mathcal{A}'(\mathbf{e})\,,$$

where $\mathcal{A}'(\mathbf{e})$ is defined in (2.14).

Due to the QoS constraints, a restriction of $\mathbf{z}(\mathbf{p})$ to $\mathcal{N}'(\mathbf{e})$, defines the objective subspace, $\mathcal{Z}'(\mathbf{e}) \subseteq \mathcal{Z}$: $\mathbf{z}\,|_{\mathcal{N}'(\mathbf{e})}(\mathbf{p}) \to \mathcal{Z}'(\mathbf{e})$. $\mathcal{Z}'(\mathbf{e})$ is interesting as it contains quality properties of the solutions. Some of these solutions are optimal, in the sense of non-dominated by any other solution in the objective space — a commonly understood notion in Pareto optimality to solve a MOP. Formally, a solution or a decision vector, $\mathbf{p}^* \in \mathcal{N}'(\mathbf{e})$ is Pareto optimal if there does not exist any other decision vector, $\mathbf{p} \in \mathcal{N}'(\mathbf{e})$, which is at least as good as $\mathbf{p}^*$ on some objective functions and better than $\mathbf{p}^*$ on remaining objective functions:

$$\nexists \mathbf{p} \in \mathcal{N}'(\mathbf{e}) \left(\underset{i\in\mathcal{U}}{\forall}\ \mathtt{z}_i(\mathbf{p}) \preceq \mathtt{z}_i(\mathbf{p}^*) \land \exists j \in \mathcal{V}\,(\mathtt{z}_j(\mathbf{p}) \prec \mathtt{z}_j(\mathbf{p}^*)) \right).$$

Optimal solutions are collected in a set to form the Pareto front, $\mathcal{PF}$, which leverages tradeoffs objectively using decision preferences.

The search for GO involves seeking a global minimum in the objective space. Effectively, it implies converting a MOP into a single-objective optimization problem (SOP). To this end, a single-objective optimization function (SOF) is needed. In our case, the SOF is the uniformly weighted euclidian distance, $\mathtt{euc}(y, up)$ between $y \in \mathcal{PF}$ and an ideal global minimum, called a utopia point, $up \in \mathcal{Z}$. In reality, there is no decision vector, $\mathbf{p}$, for which up exists, otherwise it would dominate the points in $\mathcal{PF}$, which cannot be the case. Using the SOF, the global minimum is the one with minimum $\mathtt{euc}(y, up)$:

$$\zeta(\mathcal{PF}, up) = \mathtt{arg} \underset{y \in \mathcal{PF}}{\downarrow} \mathtt{euc}(y, up) \ . \tag{4.44}$$

GO is the decision vector corresponding to $\zeta(\cdot,\cdot)$. It represents a compromise while maintaining the constraints.

Next we demonstrate the QoS fulfillment by presenting three MOPs and their solutions. For all experiments, the decision vector space is built by combining concrete system setups given in Section 4.6 for a case of one application per priority class: $\mathcal{Q} = \{0, 1, 2, 3\}$ and $N = 4$. This includes the following configurations:

- *Gaining control* – $AIFSN$ driven arbitration in Section 4.6.2.1: $\mathcal{A}^a \times \mathcal{W}^a$ for 20 decision vectors.

- *Gaining control* – CW driven arbitration in Section 4.6.2.2: $\mathcal{W}^w \times \mathcal{R}^w$ for 28 decision vectors.

- *Gaining control* – default EDCA parameter set driven arbitration in Section 4.6.2.4: $\mathcal{R}^d \times \{1\}$ for 7 decision vectors.

Hence, our decision vector space has a total of 55 network configurations. The QoS requirements considered are

$$\mathbf{e} = [(30, 2, 5, 1), (30, 1.65, 8, 1), (20, 0.75, 17, 1), (15, 0.33, 35, 1)] \, .$$

For all MOPs, data in the plots is illustrated according to the classification of the decision vectors. Thus, 'Feasible' and 'Solution' belong to $\mathcal{N}$ and $\mathcal{N}'(\mathbf{e})$, respectively. $up = (0, -11)$ always. The decision vector resulting in GO, is specified using the $T/r/i$ notation, which extends the T/r notation (cf. Section 4.6.2) to also include the element at vector index, i.

4.10.1 The channel utilization vs the expected latency of the network

In this MOP, we compute a biobjective space of two conflicting objectives, viz., the channel utilization and the average latency of the network defined over all applications, as follows:

$$\begin{aligned} \mathtt{z}_1(\mathbf{p}) &= \mathbb{E}\Big(\mathtt{f}^{\lambda}(\mathbf{p})\Big) = \frac{1}{N}\sum_{k \in \mathcal{K}} \lambda_k, \quad \text{refers (4.35)}, \\ \mathtt{z}_2(\mathbf{p}) &= -\sum_{k \in \mathcal{K}} \mathtt{f}^{\theta}_k(\mathbf{p}) = -\sum_{k \in \mathcal{K}} \theta_k, \quad \text{refers (4.32)}. \end{aligned} \tag{4.45}$$

Figure 4.23 illustrates the solution. GO is computed to be at a decision vector that configures the network using the $AIFSN$ driven arbitration,

$$\mathbf{p} = [(1, 4.3b/4/1, 1), (2, 4.3b/4/2, 1), (4, 4.3b/4/3, 1), (5, 4.3b/4/4, 1)] \, . \tag{4.46}$$

The Pareto front in this MOP offers choices for running elastic and inelastic applications. These include, applications for file transfers and gaming or video streaming, respectively. Maximizing the channel utilization favors the prior type, while for the time sensitive applications, the latency of each MPDU needs to be considered.

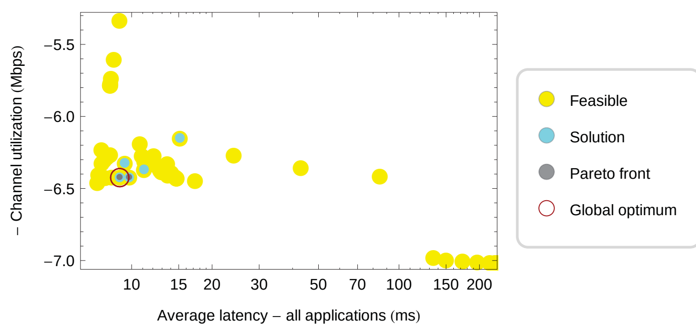


Figure 4.23: MOP 1 and solution – the channel utilization vs the expected latency of the network.

4.10.2 The channel utilization vs the expected latency of the highest priority class

Here we consider the influence of network configurations on the expected latency of the highest priority class. The biobjective space is computed as follows:

$$\begin{aligned} z_1(\mathbf{p}) &= \mathbb{E}\Big(f_0^{\lambda}(\mathbf{p})\Big), \quad \text{refers (4.35)}, \\ z_2(\mathbf{p}) &= -\sum_{k\in\mathcal{K}} f_k^{\theta}(\mathbf{p}) = -\sum_{k\in\mathcal{K}} \theta_k, \quad \text{refers (4.32)}. \end{aligned} \tag{4.47}$$

Figure 4.24 illustrates the solution. GO is computed to be at a decision vector that configures the network using the *CW* driven arbitration,

$$\mathbf{p} = [(1, 4.4/4/1, 1), (1, 4.4/4/2, 1), (1, 4.4/4/3, 1), (1, 4.4/4/4, 1)]. \tag{4.48}$$

Similar to the MOP 1 case, optimization trade-offs can be made between elastic and inelastic applications. Contrary to it, this MOP optimizes specifically for the latency of applications in only the highest priority class. Therefore, the Pareto front offers optimal solutions that are closely tuned to the latency requirements of the highest priority class.

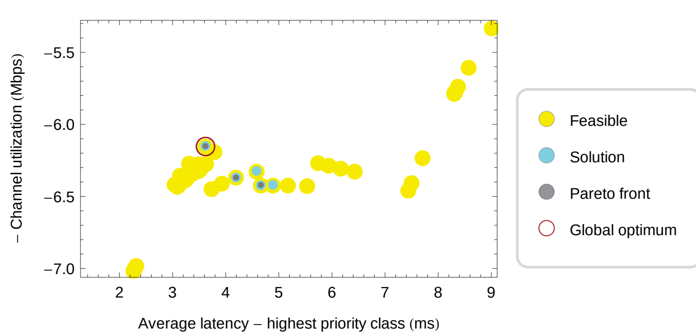


Figure 4.24: MOP 2 and solution: the channel utilization vs the expected latency of the highest priority class.

4.10.3 The channel utilization vs the QoS fulfillment proximity

The optimizations in MOP 1 and MOP 2 may lead to GO values that correspond to a large QoS fulfillment proximity. This proximity is defined over the QoS requirements, $\mathbf{e}$ and the QoS delivered, $\mathbf{d}$. It is determined as an average over all applications and QoS dimensions, the proportional difference between $\mathbf{e}$ and $\mathbf{d}$. Let $\boldsymbol{\Delta} = \mathbf{e} - \mathbf{d}$ be the difference vector, so that using (2.19), the corresponding proportional difference vector for application, k, is $\mathbf{s}_k = \mathtt{ps}(\Delta_k)$. Let $\mathbf{g}$ be a vector of user assigned weights for QoS dimensions, then the QoS fulfillment proximity is

$$\mathtt{ps}(\mathbf{e}, \mathbf{d}) = \frac{1}{N} \sum_{k \in \mathcal{K}} \mathbf{s}_k . \mathbf{g} \; . \qquad [\cdot \text{ is an inner product}] \; . \tag{4.49}$$

Due to the nature of this MOP, the biobjective space is computed for only solutions, $\mathbf{p} \in \mathcal{N}'(\mathbf{e})$:

$$\begin{aligned} \mathtt{z}_1(\mathbf{p}) &= \mathtt{ps}(\mathbf{e}, \mathbf{d}) \, , \quad \text{refers (4.49)}, \\ \mathtt{z}_2(\mathbf{p}) &= - \sum_{k \in \mathcal{K}} \mathtt{f}^{\theta}_k(\mathbf{p}) = - \sum_{k \in \mathcal{K}} \theta_k, \quad \text{refers (4.32)}. \end{aligned} \tag{4.50}$$

Figure 4.25 illustrates the solution with vector of weights for QoS dimensions, $\mathbf{g} = [0, 0.7, 0.2, 0.1]$. Again, GO is computed to be at a decision vector that configures the network using the $AIFSN$ driven arbitration:

$$\mathbf{p} = [(1, 4.3b/3/1, 1) \, , (2, 4.3b/3/2, 1) \, , (6, 4.3b/3/3, 1) \, , (7, 4.3b/3/4, 1)] \, . \tag{4.51}$$

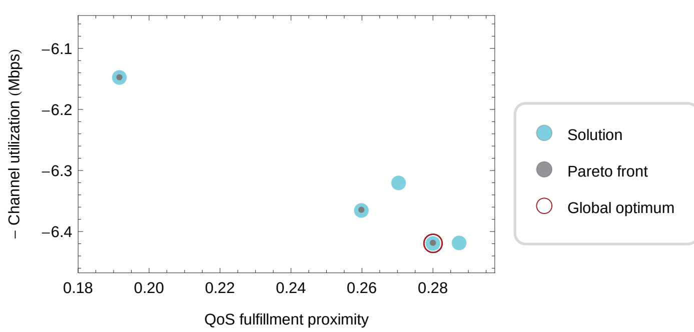


Figure 4.25: MOP 3 and solution: the channel utilization vs the QoS fulfillment proximity. Note that 'Feasible' is absent in this plot due to its nature.

In this MOP, the exactness of the QoS fulfillment is considered in the QoS proximity sense. Optimizing on this objective has multiple benefits:

- The quality of the space, $\mathcal{N}$, can be estimated for given QoS requirements. In other words, it indicates the sparseness of the network configurations in $\mathcal{N}$;
- The effects of thinly/thickly scattered network configurations can be highlighted by considering this objective while forming other MOPs. Hence, it can optimize sparseness of network configurations. Resultantly, optimizations on other objectives can be studied in isolation by selecting a sparseness level. Other objectives may include unused channel capacity and fluctuation scale due to over crowded (contended) regions over time.

4.10.4 A critique on the proposed approach

All the aims set to tackle research **Question 5** are achieved by formulating MOPs and SOPs. The MOP formulation is just an intermediate expression that also allows tradeoffs — hence, it is not necessary for our original quantitative mapping problem. The presented MOPs are based on technical aspects, however other strategic aspects may be considered. Amongst all solutions, the network configurations on the Pareto front are optimal in the sense of less wastage due to collisions and/or idle slots. They offer tradeoff analysis. The Pareto based approach allows to compare network configurations objectively. We assess the suitability of our proposed approach to tackle research **Question 5** by evaluating it within the scope of the following two questions:

q 1. What are the chances of finding a solution to a given set of QoS requirements?

Let $\hat{\mathcal{N}} \subseteq \mathcal{N}$ denote the decision vector space for our experiments: $\left|\hat{\mathcal{N}}\right| = 55$. Intuitively, the chances of finding a solution to a given set of QoS requirements is increasing with the size of $\hat{\mathcal{N}}$. Under the assumption that a solution exists for a given set of QoS requirements and that these QoS requirements are selected without bias, the probability that $\hat{\mathcal{N}}$ holds a solution is estimated as $\frac{|\hat{\mathcal{N}}|}{|\mathcal{N}|}$.

$\mathcal{N}$ is determined within the scope of parameter ranges given in Section 4.6. It is comprised of three types of property based network configurations: (4.25) for $AIFSN$, (4.26) for CW and (4.27) for $TXOP$. In each case, $|\mathcal{Q}|$[2] values are selected from the allowed range of the priority parameter to form $|\mathcal{Q}|$ classes. A collection of these $|\mathcal{Q}|$ values form a strict total ordered set on the allowed range and a $|\mathcal{Q}|$-ary relation. Let $\prec(\mathcal{V},\mathcal{Q})$ and $\mathcal{V} \triangleq \{v_1, v_2, \cdots, v_n\}$ denote the total ordered set and the allowed range, respectively. We aim for counting all possibilities of forming classes due to the priority parameter under consideration by determining $|\prec(\mathcal{V},\mathcal{Q})|$ as follows.

$$
\begin{aligned}
|\prec(\mathcal{V},\mathcal{Q})| &= \mathfrak{g}(|\mathcal{V}|, |\mathcal{Q}|, 0)\,, \\
\mathfrak{g}(n,q,i) &= \begin{cases} 1 & q = 0 \\ \displaystyle\sum_{j=i+1}^{n-q+1} \mathfrak{g}(n, q-1, j) & q > 0 \end{cases}\,.
\end{aligned} \tag{4.52}
$$

Let $\mathcal{V}^a$, $\mathcal{V}^w$, $\mathcal{V}^r$ and $\mathcal{V}^x$ denote the sets of selected values of $AIFSN$, $CWmin$, $RetryLimit$ and $TXOP$.

$$
\begin{aligned}
\mathcal{V}^a &\triangleq \{1, \cdots, 7\}\,, \\
\mathcal{V}^w &\triangleq \{8i - 1 \mid 1 \leqslant i \leqslant 16\}\,, \\
\mathcal{V}^r &\triangleq \{0, \cdots, 7\}\,, \\
\mathcal{V}^x &\triangleq \{1, \cdots, 7\}\,.
\end{aligned}
$$

For $\mathbf{n} = [1, 1, 1, 1]$ and CW growth until three steps, the number of network configurations per MAC parameter arbitration is computed as follows.

- $AIFSN$ driven arbitration: $|\prec(\mathcal{V}^a,\mathcal{Q})| \times |\mathcal{V}^w| \times |\mathcal{V}^r| \times |\mathcal{V}^x| = 31{,}360$;
- CW driven arbitration: We consider strict total orders on $\mathcal{V}^w$ and $\mathcal{V}^r$ separately and their combinations.

[2] Remember $\mathcal{Q}$ is an index set of priority classes, as defined in (2.2).

$$\begin{aligned} & 1 \times |\prec (\mathcal{V}^w, \mathcal{Q})| \times |\mathcal{V}^r| \times |\mathcal{V}^x| \\ + \quad & 1 \times |\mathcal{V}^w| \times |\prec (\mathcal{V}^r, \mathcal{Q})| \times |\mathcal{V}^x| \\ + \quad & 1 \times |\prec (\mathcal{V}^w, \mathcal{Q})| \times |\prec (\mathcal{V}^r, \mathcal{Q})| \times |\mathcal{V}^x| \\ = \quad & 1,001,560; \end{aligned}$$

- $TXOP$ driven arbitration: $1 \times |\mathcal{V}^w| \times |\mathcal{V}^r| \times |\prec (\mathcal{V}^x, \mathcal{Q})| = 4,480$;
- $AIFSN$, CW and $TXOP$ combined arbitration: We consider combinations of strict total orders on $\mathcal{V}^a$, $\mathcal{V}^w$, $\mathcal{V}^r$ and $\mathcal{V}^x$.

  $$|\prec (\mathcal{V}^a, \mathcal{Q})| \times |\prec (\mathcal{V}^w, \mathcal{Q})| \times |\prec (\mathcal{V}^r, \mathcal{Q})| \times |\prec (\mathcal{V}^x, \mathcal{Q})| = 156,065,000.$$

Aggregating these four cases,

$$|\mathcal{N}| = 31,360 + 1,001,560 + 4,480 + 156,065,000 = 157,102,400.$$

q 2. What are the chances that a given set of QoS requirements is fulfilled by the default EDCA parameter set?

Of the three presented MOPs, none is solved by the default setting, rather by settings from $AIFSN/CW$ driven arbitration schemes. The QoS properties of each network configuration point, $\mathbf{p} \in \mathcal{N}$, can satisfy multiple sets of QoS requirements, constituting what is termed as a satisfaction region of $\mathbf{p}$ in $\mathcal{A}$. Let $\mathcal{A}''(\mathbf{p}) \subseteq \mathcal{A}$ denote this region,

$$\mathcal{A}''(\mathbf{p}) \triangleq \{\mathbf{e} \in \mathcal{A} \mid \mathtt{f}(\mathbf{p})\,\underline{\mathtt{sat}}\,\mathbf{e}\}.$$

Further, let $\mathbf{p}'$ denote the default EDCA parameter set, then the chances that it can fulfill an arbitrary set of QoS requirements is given by $\frac{|\mathcal{A}''(\mathbf{p}')|}{|\mathcal{A}|}$. Such low chances, render the default EDCA parameter set inapplicable to meet most of the QoS requirements.

We leave this section by underpinning an observation that due to saturated MAC queues, each network configuration point determines an upper limit of the QoS properties of the network. The cases of unsaturated MAC queues with QoS requirements less than this limit, can be fulfilled by the corresponding network configuration point. Strategically, it allows a choice of allocating the spare capacity to the QoS requester applications.

4.11 Worst case delay performance

The well accepted benefits of WLANs render their increasing usage in industrial [37, 36, 38, 205] and medical [121, 166, 107, 10, 142, 108, 157, 109] environments. Out of many advantages, mobility and relieve from wires rank high in support of tasks that are carried out in these environments. These tasks include monitoring/controlling the devices/patients and alarms on critical events. Generally, these tasks are delay bounded and must be carried out reliably before their deadlines. Thus a delay bounded reliable service is desired over the resources. The signal to trigger such tasks, is termed as a control signal within our context. Due to the inherent randomness in the WLAN protocol, carrying this signal over a WLAN poses challenges of delay boundedness, reliability and performance fluctuations. In order to deal with these issues, a well studied WLAN performance model is mandated.

Ubiquity in industrial and medical environments has caught recent focus. A classification of industrial traffic can be used to map traffic to EDCA classes [37]. A simulation based study of delays in infrastructure WLANs can indicate the performance requirements that can be fulfilled [38, 205]. For better estimation of delays, traffic used for results must match the traffic patterns of the industry [36]. The QoS-related issues in exchanging medical data over wireless technologies, are being addressed both qualitatively and quantitatively. This data is typically characterized as having a small size that can be transported over a low capacity wireless channel. The QoS requirements of medical-related control signals [142] guide towards making engineering choices. As a preliminary study, investigations are made for distortion compensation in the electrocardiogram (ECG) signals that are exchanged over a WLAN operating in an *infrastructure* BSS mode [107, 108]. Investigations into impact of MAC arbitration parameters indicate performance limits for varying load of medical data [166]. An absolute priority scheme, can improve medical-grade reliability and delays of control signals on an EDCA network. In this context, various performance curves are drawn from simulations [121]. Reliability can also be improved by forward error correction codes [109]. The absolute priority scheme reserves slots permanently for the high priority class. These slots are used whenever traffic is available, otherwise, they go wasted. Thus, the system efficiency suffers in the absence of higher priority traffic. Wastage due to permanent reservations can be avoided by demand-based dynamic reservations. This approach can rely on tuning $AIFS/CW$ values. It is demonstrated in [7] and followed by [190] for the medical context.

A bounded delay model using the absolute priority scheme is missing from the existing literature, which we aim for. Our specific contribution is a quantitative model for the worst delay, based on mandatorily deferring the transmissions by competing processes as supported by the $AIFS$ mechanism. We restrict to only model development.

4.11.1 A system model

Our EDCA network hosts applications whose priorities are treated stochastically. Now we allow applications whose priorities are treated absolutely to achieve deterministic worst delays. Consider a joint system, composed of a stochastic priority channel (*spc*) and an absolute priority channel (*apc*). This partitioning is relative to each other, such that application(s) in the latter case have absolute non-preemptive fixed-priority (NPFP) preference over application(s) in the latter case. The models for the QoS properties for the former case still hold for the *spc*, while here we develop analysis only for the *apc*.

Let the applications in the *apc* are indexed by a set, $\mathcal{K}^{\text{abs}}$, in decreasing priority,

$$\mathcal{K}^{\text{abs}} \triangleq \Big\{1, 2, 3, \cdots, N^{\text{abs}}\Big\}. \tag{4.53}$$

Corresponding index set for the *spc* is denoted by $\mathcal{K}^{\text{sto}}$. We set off by creating a partition of the system capacity into the two desired types of channels. Some slots immediately after the system busy state are reserved for the *apc*, while slots beyond the reserved slots constitute the *spc*. The number of these reserved slots is the worst case MAC arbitration slot for the *apc*. This required system behavior is achieved by the $AIFS$ based differentiation. Let ws^{abs} denote the MAC arbitration worst case slot for applications in the *apc*,

$$ws^{\text{abs}} = \mathop{\uparrow}_{k \in \mathcal{K}^{\text{abs}}} (a_k + CWmax_k).$$

The slots for the *spc* begin from ws^{abs}, while their occurrence is subject to the traffic generated in *apc*. Thus the $AIFS$ of all MAC processes in *spc* are shifted by ws^{abs}. The *apc* allocation to multiple applications is carried out on the NPFP basis, as explained next.

In NPFP approach, each application within *apc* has a unique absolute priority, which is enforced by the $AIFS$ differentiation. For $k \in \mathcal{K}^{\text{abs}}$,

$$\begin{aligned} AIFSN_k &= CW_k + \begin{cases} 1 & k = 1 \\ AIFSN_{k-1} + 1 & k > 1 \end{cases}, \\ CWmin_k &= 0, \\ CWmax_k &= 0. \end{aligned} \tag{4.54}$$

An arbitrary moment, t, may see any application in the system being served. The worst delay for an application, $k \in \mathcal{K}^{\text{abs}}$, λ'_k can occur at t if a longest transmission from *spc* is being served, while all applications higher in priority than k have data to transmit. Suppose delays due to these two events are denoted by λ^{sto} and λ^{abs}_k, respectively.

$$\lambda'_k = \lambda^{\text{sto}} + \lambda^{\text{abs}}_k. \tag{4.55}$$

λ^{sto} is determined by the longest expected retention service time, μ_i in (4.21), from *spc*.

$$\lambda^{\text{sto}} = \mathop{\uparrow}_{i \in \mathcal{K}^{\text{sto}}} \mu_i.$$

λ^{abs}_k is determined as an aggregate of expected retention service times of applications higher in priority order than k.

$$\lambda^{\text{abs}}_k = \sum_{j=1}^{j<k} \mu_j \ .$$

4.12 Conclusions

This chapter is the core of the thesis. It addresses three research questions.

Research **Question 3** deals with the modelling of a random-based system such as any variant of a wireless MAC protocol. The question is addressed by developing a Markov model, which facilitates in tackling random behavior systems. It maps network configurations to stationary probability distributions, one per QoS class of MAC processes. An important aspect of the model is realistic assumptions that are compliant with the EDCA protocol specifications. One such assumption is consideration of finite *RetryLimit*. The model is valid for any CW update discipline. It allows correct predictions for the QoS properties. This Markov model, solved iteratively using our algorithm, seeks a fixed point of the system. We have implemented it, realizing our software tool, which is used to validate the hypothesis set on the preferential behavior of the MAC parameters. The probability masses increase in lower backoff/retry counter values. Skewness in probability distributions also increases with priority of processes, while probability distributions gets more uniform towards decreasing priority of processes. In general, the extent of these observations depends on system settings.

The Markov model is comparable in complexity to the zone-based models that suffer from scalability issue, though it is more complex than the slot-based models. The implementation performance of this algorithm is thoroughly reported in terms of memory required, iterations needed and time taken, for various network configurations. For instance, memory requirements are determined by the CW settings, whose higher values form more stable systems. The stability criteria is taken to be the number of iterations needed to reach system fixed point. Further, the time of convergence is determined to be a polynomial of memory requirements. Timing results suggest a need for parallelized hardware for on-the-fly computations in an operational system. Such an implementation performance study offers tradeoff points. For instance, accuracy of a solution versus time of the solution can be optimized against each other.

Research **Question 4** is regarding development of models for the QoS properties. To this end, we have developed mappings from the network steady states as represented by stationary probability distributions, to the QoS properties. In contrast to oversimplified assumption of infinite retry attempts, we consider a finite case. These mappings are validated by corresponding tools, and verified by our custom built EDCA specific discrete event simulator, GRADES [5]. The results from the analysis and the simulations conform well enough to repose our trust in them.

Consideration of finite retry attempts, allows accurate prediction of QoS properties. Specially, the difference from an infinite case is profound when allowed retry attempts are small. We confirm that except MAC latency, which increases, channel access frequency and channel share decrease with $AIFS$ and CW. MAC reliability depends on allowed retry attempts. $TXOP$ has a negative/positive/negative impact on frequency/share/latency, respectively. We also note that a MAC latency model that does not consider MAC losses, gives pessimistic results. This is due to delay accumulation of MPDUs that are dropped due to exhaustion of retry attempts.

We have also analysed a case of achieving a bounded delay on an arbitrary MPDU, to cater requirements of hard real time applications, representing for instance, control signals. Transportation of hard real time applications over WLANs is gaining grounds due to their benefits in industrial and medical environments. As an initial investigation, we contribute a model for the worst case delay of such applications. These applications are treated with the highest priority that is enforced by the lowest $AIFS$ while making MAC reservations. A side effect of this approach is permanent reservations that go wasted when applications are silent.

Achieving a distributed control over QoS properties of processes allows us to deal with research **Question 5**. To this end, the QoS semantics, specifically the satisfiability relation (cf. Section 2.5) is used to search the solution space against a given set of QoS requirements. A side effect of our proposed satisfiability relation is multiple solutions. To tackle this issue, we define objective spaces based on their technical quality properties and formulate multiobjective optimization problems in them. Using Pareto analysis, we find the optimal solutions. They represent tradeoff points, offering choices for system adaptations. Finally, a single-objective optimization problem is formulated, whose solution provides us the global optimal (solution).

Originally, this research question seeks a transformation, from QoS space to network space. Due to technology limitations, the network space is discrete. A transformation mapping to an unreachable point, does not serve the purpose. Overcoming this issue gives rise to another set of problems, which include shifting to a discrete point and dealing with effects of the shift. The shift may lead to a discrete point that does not fulfill the original given QoS requirements. Contrary to this, we effectively discretize the solution space to avoid unreachable point situation.

The objective spaces can also be defined using non-technical quality properties for strategic decisions. We have compared Pareto analysis based approach to determine the solution, against another method, gradient descent, which requires a continuous space. Due to the issue of unreachable points, this method is found to be inapplicable.

The suitability of our approach is assessed in terms of chances of finding a solution to a given set of QoS requirements. As expected, these chances increase with size of the space of decision vectors. In context of our selected network parameter ranges, maximum size of this space is $156,065,000$, which also indicates the memory requirements. Lastly, we judged the suitability of using the default EDCA parameter set to solve an arbitrary QoS problem. The result suggests quite weak chances, hence it is inapplicable for vast majority of QoS problems. This is supported by the observation that none of our three QoS problems are solved by the default set.

In the next chapter, we study the optimization of the channel efficiency.

5

Streaming-aware channel utilization improvement model

The previous chapter addressed the problem of achieving distributed control of an EDCA network. To this end, we have presented quantitative models for steady state behavior and QoS properties of the EDCA network. It also addressed the problem of mapping the transportation related QoS requirements of applications to some desired network behavior such that it fulfills those requirements. Each network behavior determines some expected network bandwidth termed as system channel utilization, which is the effective bandwidth available to applications. Thus, each mapped network behavior has a system channel utilization aspect that can be considered as its quality measure. For convenience, this quality measure is termed as a mapping efficiency. Two mapped network behaviors can be compared in terms of their mapping efficiencies. Given that the wireless channel bandwidth is a scarce resource, the challenge of optimal mapping efficiency is crucial. Its significance as a system concern is further substantiated by an increasing number of consumer electronic devices in modern network setups - especially that majority of them stream multimedia content of varying quality.

The said mapping is supported by the IEEE 802.11e standard [92], which provides cooperative and competitive distributed control for bandwidth assignment through a set of MAC mechanisms. These mechanisms are explained in Chapter 2, while the effect of their settings is modeled in Chapter 4. While these MAC mechanisms leverage priority arbitration for wireless channel usage, they are typically configured without taking into account, the knowledge of streaming services. Thus, even the cooperative sharing of the wireless channel between MAC processes does not automatically lead to optimal mapping efficiency.

In the example of video streaming, switching control to another station may be done halfway the transmission of a video frame. In that case, allowing the station to complete its video frame transmission can be a better choice. By taking properties of the transferred data into account, both utilization and the video frame latency can be improved. We investigate the improvement in utilization by varying $TXOP$ in relation to the video frame size, and also how this is affected by other MAC parameters. This chapter addresses the following research question, as also posed in Chapter 1:

Question 6. How to improve the channel utilization for the wireless network under control of a distributed MAC, by exploiting characteristics of applications?

We present a streaming-aware wireless channel utilization policy. For defining and understanding the policy, the two major concerns, *gaining control* and *retaining control*, duly quantified in Chapter 4, are extended according to application requirements for obtaining the bandwidth utilization. This allows us to consider transmission durations of video frames. The resultant channel utilization model is simpler than previous works.

By applying the extended channel utilization model to known video traces, we determine bandwidth shares and bandwidth utilization of a system of competing video streams. This model is validated by testing different $TXOP$ settings. The validated results are compared and verified with simulations, which are done in GRADES [5]. A utilization improvement of up to 18% is observed to be possible, showing that such cross-layer optimization improves the resource use.

We set off by reviewing the prior approaches to channel utilization in WLAN systems in Section 5.1. Section 5.2 derives a channel utilization model that is applicable to the QoS control mechanism of the IEEE 802.11e standard. In Section 5.3, a set of scenarios are sketched to derive analytical and simulation results, to validate and verify the presented model. Concluding remarks are given in Section 5.4.

5.1 Related work

Wireless networks are powerful enablers for realizing the dream of running application services 'anywhere, anytime'. The transportation of video streams and real-time traffic over the IEEE 802.11 [92] networks has got a lot of attention [21, 236, 140]. Various application services are increasingly categorized based on their QoS requirements, and typically mapped onto the wireless channel using the MAC enforcement schemes [179, 184, 244, 178, 146, 85, 224]. Generally the usage of MAC parameters influences the frequency of channel access. Hence, based on the mechanisms invoked, these schemes exhibit different stochastic performance guarantees for channel shares as well as influence each others priority impact [4]. Nevertheless transforming the mapping of application requirements into a channel partitioning problem does not automatically consider the system-related extra-functional concern of mapping optimality. This leads to wastage of already scarce channel resource, which is undesired. The utilization of channel resource can be improved by increasing the reliability of links using the PHY parameters [88]. Thus, channel utilization can be studied as a function of parameters at various OSI layers. It justifies a case for mapping the application requirements with an additional MAC level objective to prevent the channel wastage.

Therefore a MAC enforced channel arbitration model should consider both aspects of QoS and optimality. The QoS aspect should be correctly predicted over all ranges of $AIFSN$ and CW values [201]. The optimality aspect prevents the wastage, and it is significant especially when it uses the application knowledge.

In this work, we focus on both aspects of QoS and optimality together. Specifically the introduced model exploits the independence between the two concerns, *gaining control* and *retaining control*, thus effectively reducing the complexity of the problem. It enables the separation of the application-to-MAC mapping problem from MAC controlled shares allocation, as presented in this chapter.

5.2 Performance model

This section develops a channel utilization model for a QoS based ad-hoc network. Recapturing from Section 2.7, *gaining control* and *retaining control* have key roles in the bandwidth allocation problem. The dependency of bandwidth on the MAC parameters is restated from (4.32) as

$$\theta_k = \nu_k \cdot (8 \cdot L_k \cdot x_k) \ . \tag{5.1}$$

It is reexpressed using (4.31), (4.16) and (4.20),

$$\theta_k = \rho_k \cdot \frac{8 \cdot L_k \cdot x_k}{\mathcal{C} + \sum\limits_{i \in \mathcal{K}} \rho_i \cdot \mu_i + \rho' \cdot \mu'} \ . \tag{5.2}$$

θ_k leverages the derivation of the new channel utilization model. Hence, it is based on the same assumptions as for the stochastic models (cf. Section 4.2) and uses the same notations as in Table 4.1.

5.2.1 Channel utilization

The channel utilization is defined as an aggregation of the expected channel shares of all MAC processes in the network.

$$\mathcal{U} = \sum_{k \in \mathcal{K}} \theta_k \ . \tag{5.3}$$

In this chapter, we study the effect of varying the transmission opportunity vector, $\mathbf{x}$. In particular, we compare with the standard policy of having $x_k = 1$ for all processes and we denote this policy by $\mathbf{x} = \mathbf{1}$. Hence, the channel utilization is explicitly expressed as a function of $\mathbf{x}$,

$$\mathcal{U}(\mathbf{x}) = \sum_{k \in \mathcal{K}} \rho_k \cdot \frac{8 \cdot L_k \cdot x_k}{\mathcal{C} + \sum\limits_{i \in \mathcal{K}} \rho_i \cdot \mu_i + \rho' \cdot \mu'} \ . \tag{5.4}$$

We also unfold the second term in the denominator of (5.4) using (4.21), to highlight the dependency of the channel utilization on $\mathbf{x}$.

$$\mathcal{U}(\mathbf{x}) = \sum_{k \in \mathcal{K}} \rho_k \cdot \frac{8 \cdot L_k \cdot x_k}{\mathcal{C} + \sum\limits_{i \in \mathcal{K}} \rho_i \left(x_i \left(tt_i + SIFS \right) - SIFS \right) + \rho' \cdot \mu'} \ . \tag{5.5}$$

The utilization improvement is defined as follows.

$$\beta\left(\mathbf{x}\right) = \frac{\mathcal{U}(\mathbf{x}) - \mathcal{U}(\mathbf{1})}{\mathcal{U}(\mathbf{1})} \cdot 100 \ . \tag{5.6}$$

5.2.2 Application mapping

The channel utilization in (5.4) is derived on the underlying assumption that applications transmit all full MPDUs. More specifically, application k transmits x_k MPDUs in each transmission opportunity, each carrying an application payload of L_k bytes. Thus, in each transmission opportunity, it transmits $x_k \cdot L$ bytes. Some applications may not conform to this assumption. Hence, in order to derive the channel utilization in a system of such applications, there is a need to consider application messages of arbitrary lengths. Let application k generate arbitrary length messages of expected size, $\mathbb{E}(M_k)$. For such a system, let $\bar{\mu}_k$ and $\bar{\mu}'$ be the expected retention service time of MAC process k and the expected collision service time, respectively. The channel utilization is then

$$\mathcal{U}(\mathbf{x}) = \sum_{k\in\mathcal{K}} \rho_k \cdot \frac{8\cdot\mathbb{E}(M_k)}{\mathcal{C} + \sum\limits_{i\in\mathcal{K}} \rho_i \cdot \bar{\mu}_i + \rho' \cdot \bar{\mu}'} \ . \tag{5.7}$$

This necessitates computing three parameters: $\mathbb{E}(M_k), \bar{\mu}_k$ and $\bar{\mu}'$.

In this system, we consider as applications, multimedia streams with each one possibly having a different QoS requirement. We assume that the quality of a stream (and corresponding QoS requirement) is characterized by the video frame sizes. These frame sizes vary with the contents and frame types and can be regarded as a mapping, $\mathtt{m}_k(i)$, from a frame sequence number, i, to the size of frame i. We also assume that application payload limits per MPDU are equal, denoted by L. From this we determine $\mathbb{E}(M_k)$ as follows. Frame i may be sent split over several transmission opportunities by using $\lfloor \frac{\mathtt{m}_k(i)}{x_k\cdot L_k} \rfloor$ full opportunities of size $x_k \cdot L_k$ bytes and a remaining one of size, $\mathtt{m}_k(i) \bmod x_q \cdot L_k$. For a video of a given size, we determine the histogram of occurrence of transmission opportunities of certain sizes in $\mathtt{g}(j)\,, 1 \leq j \leq x_k \cdot L$. Let G denote the total number of used transmission opportunities, $G = \sum\limits_{j=1}^{x_k\cdot L_k} \mathtt{g}(j)$. $\mathbb{E}(M_k)$ is now defined as the average size of a transmission opportunity,

$$\mathbb{E}(M_k) = \frac{1}{G} \cdot \sum_{j=1}^{x_k\cdot L_k} j \cdot \mathtt{g}(j) \ . \tag{5.8}$$

$\mathbb{E}(M_k)$ can be regarded as a property of the video. Alternatively, it can be computed on-the-fly over a window of video frames.

In order to derive the corresponding expected retention service time, observe that the time it takes to transmit a single full MPDU is a transmission transaction time, $\mathbb{t}$, expressed in (4.22). For transmitting $\mathbb{E}(M_k)$ bytes of payload, $\left\lceil \frac{\mathbb{E}(M_k)}{L} \right\rceil$ MPDUs are needed, of which the last one may be short of $\left\lceil \frac{\mathbb{E}(M_k)}{L} \right\rceil \cdot L - \mathbb{E}(M_k)$ bytes from being full. This allows us to conveniently express the time of transmission of application payload, m bytes, in terms of transmission transactions as

$$\mathtt{t}^{\mathrm{mac}}(m) = \left(\left\lceil \frac{m}{L} \right\rceil (\mathbb{t}(L) + SIFS) - SIFS \right) - \frac{8\left(\left\lceil \frac{m}{L} \right\rceil \cdot L - m \right)}{\mathscr{R}^{\mathrm{d}}}, \tag{5.9}$$

where

$$\mathbb{t}(L) = \frac{h^{\mathrm{phy}}}{\mathscr{R}^{\mathrm{b}}} + \frac{8\,(L+H)}{\mathscr{R}^{\mathrm{d}}} + t^{\mathrm{p}} + SIFS + \frac{h^{\mathrm{phy}}}{\mathscr{R}^{\mathrm{b}}} + \frac{8\cdot ACK}{\mathscr{R}^{\mathrm{d}}} + t^{\mathrm{p}}, \tag{5.10}$$

is transmission transaction time of L bytes and reexpression of (4.22) to show its explicit dependency on some payload. The second term on the right hand side of (5.9) is the excess transmission time that may have been added to the time of the last transmission transaction. Now

$$\bar{\mu}_k = \mathtt{t}^{\mathrm{mac}}(\mathbb{E}(M_k)) \ . \tag{5.11}$$

Lastly, the expected collision time is determined as follows. The occurrence of a collision is observed at the end of the transmission of the first MPDU in a transmission opportunity. Therefore, the expected collision time is determined by the expected size of the first MPDU in a transmission opportunity. For our system, we assume that this MPDU has maximal application payload, L.

$$\bar{\mu}' = \mathtt{t}^{\mathrm{mac}}(L) \ . \tag{5.12}$$

In the experiments, we examine the difference between these approximations and the actual behavior, as shown by simulations.

5.3 Validation and verification

Our analytical model is validated by predicting shares and utilization values from (5.1) and (5.4), which are then verified against simulations using our EDCA specific custom made discrete event simulator GRADE. For the comparison between model and simulation, as well as for demonstrating the utilization improvement, three experiments are setup, exercising the different priority arbitration mechanisms (i.e., MAC settings) that are applied in a general system context.

5.3.1 General system setup

For the channel utilization model, four different streams share the channel, which we represent by four MAC classes of one stream each. The transmitter stations in Figure 4.3 represent four ad-hoc connections, all of which exchange video information with different properties and requirements in principal. We mostly use the configurations from the general setup in Section 4.6.1. When a parameter is changed for the channel utilization model, it is specified accordingly.

Priority levels According to our description, $\mathcal{Q} = \{0, 1, 2, 3\}$ and $\mathbf{n} = \{1, 1, 1, 1\}$, so that $\mathcal{K} = \{1, 2, 3, 4\}$.

Network mode The network is operated in an ad-hoc mode, assuming that all devices are in the broadcast range of each other.

Channel access mechanism EDCA is used to access the channel at the MAC sublayer, whose parameters draw values from the following sets of values, for obtaining all results:

- $AIFSN \in \{1, \cdots, 7\}$;
- $CW \in \{7, \cdots, 1023\}$. CW grows until three steps.

OSI parameters The PHY/MAC parameters of IEEE 802.11b [92], listed in Table 4.2, are selected, except for the application payload limit per MPDU, $L = 960$ bytes, which is uniform across the board.

Simulation time Excluding the initial start-up and final wind-off times, all simulations are run for 100 seconds. The averages over this interval is reported for all results.

System convergence The system convergence is achieved by a call, $\texttt{FPC}\big(10^{-10}, 0.75\big)$ (cf. Algorithm 6).

Video traffic model H.264 [78] frame sizes are derived from video traces of a Harry Potter movie [209]. We use the same video trace for all four streams. In this study we are not interested in the real-time properties of the streams but in their variability, in the effect of changing the $TXOP$ setting and in the comparison of model and simulation. Due to variability in MAC service rates of streams, there are no unwanted 'synchronization' effects within the simulations.

MAC service time In each of the three experiments we vary the $TXOP$ settings over the values listed in Table 5.1. The first row in this table represents the original case without improvement, which yields $\mathcal{U}(\mathbf{1})$, while the last row (marked with a $*$ vector) represents the case in which the system transmits entire video frames in each opportunity. This means that $TXOP$ can potentially take very large values.

Table 5.1: Set of vectors for setting service times, corresponding to a transmission opportunity in number of MPDUs. The first row represents a retention vector with a single MPDU for all classes, and gives the reference utilization, $\mathcal{U}(\mathbf{1})$. The last row represents transmission of entire video frames.

$\mathbf{x}$	$\sum_{k\in\mathcal{K}} x_k$
$[1,1,1,1]$	4
$[2,1,1,1]$	5
$[3,1,1,1]$	6
$[4,1,1,1]$	7
$[6,2,1,1]$	10
$[8,3,2,1]$	14
$[10,5,3,2]$	20
$[*,*,*,*]$	variable

5.3.2 Experiments, results and discussion

The described system is examined varying the two parameters that influence *gaining control*: $AIFSN$ and CW. These parameters are used to define three priority arbitration schemes for the experiments — their values are given in Table 5.2. For all experiments, $RetryLimit = 7$, across the board. The first scheme is the $AIFSN$ driven arbitration (cf. Section 4.6.2.1) and conforms to the MAC settings in (4.25). According to this relation, classes have identical sets for CW but different settings for $AIFSN$. The second scheme is the CW driven arbitration, (cf. Section 4.6.2.2) and conforms to the MAC settings in (4.26). According to this relation, $AIFSN$ settings are identical for the classes but each has its own CW set. Finally, the third scheme is the default EDCA parameter set driven arbitration (cf. Section 4.6.2.4), in which both $AIFSN$ and CW are changed jointly as described in the IEEE 802.11e specification [92].

We determined analytically and through simulation, the bandwidth shares and utilization (in Mbps) for each of the three schemes while we varied the $TXOP$ (i.e., retention) setting according to Table 5.1. These settings enable non-uniform service times for classes (except the first setting). The relevant plots are shown in Figure 5.1, which presents adjacent bar pairs to compare analytical and simulation results. The bar heights show the actual utilization, whereas utilization improvement ($\beta(\mathbf{x})$ of (5.6)) is shown alongside each bar separately. All simulations are run for 500 seconds and repeated 20 times. 95% of the simulations fall within an error margin of 0.05% of the mean over these 20 repetitions.

The first thing to observe is that the model and the simulation results are remarkably close.

Second, for each priority scheme the channel utilization is seen growing with $\sum_{k\in\mathcal{K}} x_k$ (along the retention vector axis in Figure 5.1), though the same does not hold for the individual class shares, which are shown in each bar. Within a given arbitration scheme, this is due to the fact that the share of a process is proportional to the fraction of the total retention time as given in (5.2).

Third, the improvement is explained as a reduction of the overhead of contention and collision. This reduction is maximal and optimal in the last columns of the three experiments, in which entire frames are transmitted. There is a downside to this, however. With the choice of a large $TXOP$, there is no guarantee anymore for a certain share per process, which shows in the graphs through a much less prioritized division of the shares over the classes. Therefore, the best $TXOP$ setting for streaming is probably significantly larger than 1 to cancel out the overhead, but still fixed.

Table 5.2: *Gaining control* configurations

(a) $AIFSN$ driven arbitration

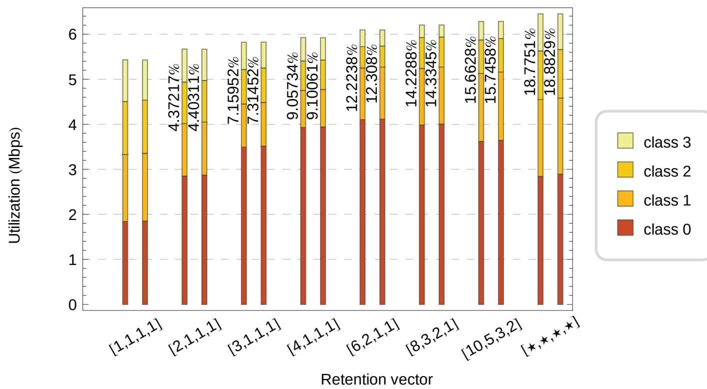
parameter	configuration
$AIFSN$	$[1, 2, 3, 4]$
CW	$\begin{bmatrix} \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \\ \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \\ \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \\ \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \end{bmatrix}$

(b) CW driven arbitration

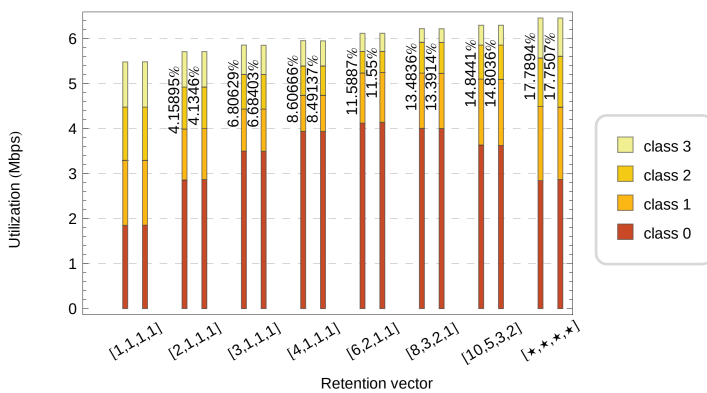
parameter	configuration
$AIFSN$	$[1, 1, 1, 1]$
CW	$\begin{bmatrix} \mathbf{31}, 63, 127, 255, 255, 255, 255, 255 \\ \mathbf{39}, 79, 159, 319, 319, 319, 319, 319 \\ \mathbf{47}, 95, 191, 383, 383, 383, 383, 383 \\ \mathbf{55}, 111, 223, 447, 447, 447, 447, 447 \end{bmatrix}$

(c) EDCA default arbitration

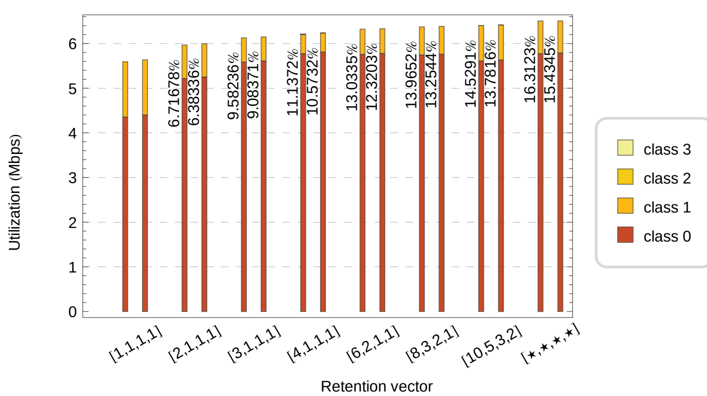
parameter	configuration
$AIFSN$	$[2, 2, 3, 7]$
CW	$\begin{bmatrix} \mathbf{7}, 63, 15, 15, 15, 15, 15, 15, 15 \\ \mathbf{15}, 31, 31, 31, 31, 31, 31, 31 \\ \mathbf{31}, 63, 127, 255, 511, 1023, 1023, 1023 \\ \mathbf{31}, 63, 127, 255, 511, 1023, 1023, 1023 \end{bmatrix}$



(a) *Gaining control* – $AIFSN$ driven arbitration. Configurations in Tables 5.1 & 5.2a.



(b) *Gaining control* – CW driven arbitration. Configurations in Tables 5.1 & 5.2b.



(c) *Gaining control* – EDCA default driven arbitration. Configurations in Tables 5.1 & 5.2c.

Figure 5.1: Utilization, $\mathcal{U}(\mathbf{x})$, as a function of $\mathbf{x}$ in Table 5.1. The adjacent bar pairs represent analytical and simulation results. The utilization improvement, $\beta(\mathbf{x})$ of (5.6), is shown alongside each bar. The first bar pair shows the reference setting without our proposed improvement, i.e., $\mathcal{U}(\mathbf{1})$, while the last bar pair corresponds to transmitting full video frames in each opportunity. The utilization is shown separately.

5.4 Conclusions

This chapter presents a channel utilization predictive model to judge the mapping efficiency of the IEEE 802.11e MAC based QoS control that is used to fulfill the QoS requirements of applications. It investigates a streaming-aware channel assignment policy that improves the utilization. The MAC controls regulate average channel shares of contending processes. The derivation of the model is very simple compared to other works and based on separation of the two concerns, *gaining control* of the channel and *retaining control* of the channel. Using the model, we can systematically investigate the effect of variations in access time on the share a process obtains and on the channel utilization. The aim of these variations is to consider increasingly the application properties — we investigate the case of video streams. We show the quality of the model by comparing with a simulation of the IEEE 802.11e standard. In a series of experiments with varying access probabilities, we show clearly the effect of changing the access times on the utilization. An improvement up to 18% over the normal case is achieved when entire video frames are transmitted. In practice, this top rate is unlikely. Thus, a more realistic estimate is the average, 11%, of the improvement range $4\% - 18\%$.

We have discovered that the extent of improvement in the channel utilization is related to the application. Excessive access times may adversely affect the delay performance of competing applications. The effects of our policies may diminish in erroneous channels, in which case the access times are reduced to one MPDU per transmission opportunity.

In the next chapter, we propose and evaluate a dynamically adaptive model, controlled in a distributed manner, in which QoS-enabled stations can autonomously configure their MAC processes to minimize the channel wastage and to improve the channel utilization.

6

An autonomic self-optimization model

Perceptually appealing high graded streaming services maintain a persistent quality, even though the entropy levels of encoded information vary temporally/spatially. Transporting the streams over the network, imposes fluctuating QoS requirements on the underlying network to cope with these variations. Generally these requirements are mapped statically onto scarce wireless channel bandwidth using the IEEE 802.11e standard [92], which supports prioritized arbitration through a set of MAC mechanisms. In an absolute QoS protection policy, the static mapping considers the maximum share requirements of streams. This approach makes invariant bandwidth reservations by considering maximum requirements of streams. A consequence of this approach is overbookings over time. These overbooked but unused 'holes' remain unnoticed and hereby highlighted as a bandwidth wastage. We consider this invariant reservation approach, as pessimistic.

Hence, due to the time varying disparity between the QoS requirements of streams and static but maximum channel reservations, the system performance remains suboptimal. If the system is adaptive and reactive, then the long-term shares of the competing elastic traffic (time non sensitive) as well as channel utilization can be improved. In context of the research questions posed in Chapter 1, it refers to the following:

Question 7. How to adapt distributed control of MAC autonomously, to avoid channel wastage due to pessimistic reservations?

In order to improve over the static and the pessimistic reservation schemes in a QoS based ad-hoc network, the challenge is to dynamically tune the MAC priority arbitration mechanisms to minimize the channel bandwidth wastage. This chapter has a two-fold contribution.

First, a dynamic channel reservation model is presented, with a three phase functionality. In phase one, dynamic channel demands over time are monitored autonomously by stations. In phase two, monitoring stations decide distributively on moments of overbooked reservations that are precisely wastage instants. These instants are intercepted as signals of potential adaptation of MAC priority arbitration mechanisms, to delegate the unused bandwidth to the elastic traffic in the system. In phase three, deciding stations react timely to adapt the MAC priority arbitration mechanisms for the reallocation of shares in the system, to prevent the channel wastage.

Secondly, the impact over the shares of the streams due to this adaptive reservation model is studied and reported as well.

This model is verified by simulating system behavior comprising of known video traces (inelastic traffic) and a file download (elastic traffic). Only inelastic applications have minimum QoS requirements. The performance of dynamic adaptability of these MAC priority arbitration mechanisms is compared on channel utilization, to feature their feasibility for such dynamic shares reallocation scheme. The reported results are drawn from GRADES [5].

We derive the performance of our proposed model under different MAC priority arbitration mechanisms that enforce the QoS control. The results demonstrate gain in channel utilization due to our proposed dynamic approach. Such a study highlights unintended wastage of channel, which can be avoided. In an ad-hoc network scenario that we consider, these reactive systems are desired for optimized network performance.

6.1 Related work

Automatic network adaptivity has been in focus in recent years. The work carried out in this domain has addressed problems of various complexity and scope. The proposed solutions pertain to various layers of the OSI model. Radio resource management by network monitoring has been studied for large scale systems such as Global System for Mobile Communication (GSM), Universal Mobile Communication System (UMTS) and WLAN [193, 162] as well as for hybrid networks [180]. Challenges such as heterogeneity in wireless networks, diversity in application types and links have been addressed using adaptations at the transport layer [3, 186, 164]. The problem that one TCP variant does not fit well to all dynamic characteristics of operational networks is highlighted in [31]. To alleviate this problem, switching between the TCP variants has been proposed. Adaptivity in wireless networks has also been addressed using learning automata to increase performance [151] and channel allocation [68]. Intelligent algorithms at access points can use polling to cater for the application requirements asymptotically [150, 149].

Due to limited capacity of networks, applications with QoS requirements have to make reservations. When the channel capacity and the QoS requirements do not change over time, static reservations are enough. These limitations may not hold always, especially due to time-variant requirements of inelastic streaming applications. Hence, instead of static reservations, dynamic reservation schemes are feasible that can adapt over time [16, 141, 2, 65, 206, 113, 141, 77]. Specific to WLANs, similar adaptive bandwidth reservation approaches have been studied [188, 144, 100, 58, 242]. These include the multi-rate feature of WLANs, which can adapt network according to the link variations [228, 230, 62, 51, 135, 125, 207, 153, 86]. Adaptivity in WLANs has also implied autonomic and decentralized management [176, 172, 200, 74]. In this regard, some works have considered optimal channel utilization in classless WLANs by overlaying intelligence over DCF backoff scheme [32, 33, 34]. The methods in these works rely on estimating the number of transmitting stations in the network. For large networks, this estimation may easily be inaccurate, hence the methods can be accordingly misguided. An improvement to these methods is proposed in [25], which remains independent of the number of transmitting stations. The improved method adapts the default DCF backoff behavior by allowing probabilistic transmissions to achieve asymptotic network behavior. Similar adaptive schemes are applied for traffic prioritization, admission control and estimating the number of nodes in the network [61, 203]. Though dynamic tuning of MAC parameters has been studied, the question of applying it in an EDCA QoS control to achieve optimality is still interesting [174]. It turns out to be more challenging in unmanaged networks to absolutely preserve the QoS levels of the multimedia streams, while also minimizing the channel wastage due to overbooked shares by them. These overbookings protect shares over time by considering only maximum requirements of the streams.

In this chapter, we propose and evaluate a single method which can transparently adapt various EDCA mechanisms used to enforce the QoS control in unmanaged networks, with the following aims:

a. to restrict the bandwidth wastage due to overbookings of streaming services, so that the gain can be utilized by competing applications;

b. to protect the requirements of real-time applications such as streaming services;

c. to keep the inherent distributed nature of unmanaged networks through distributed decisions;

d. to keep functional separation of OSI layers. Our proposed reactive method requires only MAC level information. This is in contrast to proactive methods that mostly rely on application level information.

Our proposed method is required to detect fine grained overbookings that occur at random moments. Precise detectability and appropriate decisions mandates timely adaptations so that our optimality seeking predictive control steers the system towards the desired behavior.

6.2 Conceptual model

This section builds an intuitive understanding of the optimality seeking predictive control by formally presenting the concepts used and the problem to address. The core objective of this chapter is to study optimization of a QoS based ad-hoc network that satisfies QoS requirements of applications. This allows us to benefit from notations introduced in Section 2.5, though they need to be extended. We proceed by introducing a time series analysis of system dynamics.

Let $\mathbf{e}(t)$ be the QoS requirements of applications at time, t that are satisfied by some network configuration, $\mathbf{p}(t)$, thus $\mathtt{f}(\mathbf{p}(t))\,\underline{\mathtt{sat}}\,\mathbf{e}(t)$. The QoS satisfaction constraint generates a region in $\mathcal{A}$, $\mathcal{A}'(\mathbf{e}(t))$, with centroid $\mathbf{e}(t)$,

$$\mathtt{f}\,|_{\mathcal{N}'(\mathbf{e}(t))}\,(\mathbf{p}(t)) \rightarrow \mathcal{A}'(\mathbf{e}(t))\,. \tag{6.1}$$

where

$$\mathcal{A} \supseteq \mathcal{A}'(\mathbf{e}(t)) \triangleq \{\mathtt{f}(\mathbf{p}(t)) \in \mathcal{A} \mid \mathtt{f}(\mathbf{p}(t))\,\underline{\mathtt{sat}}\,\mathbf{e}(t)\}\,. \tag{6.2}$$

(6.1) and (6.2) extend (2.14) and (2.15), respectively.

As a result of our proposed dynamic QoS control, the system performance is affected. Thus, system analyzability over time is required. Suppose $\mathtt{P}(t_1, t_2)$ be a trajectory of network configurations in $[t_1, t_2]$,

$$\mathtt{P}(t_1, t_2) \triangleq \{\mathbf{p}(t) \mid t \in [t_1, t_2]\}\,. \tag{6.3}$$

The trajectories for the static QoS control and our proposed dynamic QoS control are termed as a static trajectory and a dynamic trajectory, respectively — in the same order, denote them by $\mathtt{P}^{\bullet}(t_1, t_2)$ and $\mathtt{P}^{*}(t_1, t_2)$. We aim at comparing the system performance at these two trajectories. Introducing yet another term, an ideal trajectory, $\mathtt{P}^{\diamond}(t_1, t_2)$ that represents optimal system performance. Obviously, a trajectory will be considered better than other, if it is closer to the ideal trajectory. Proximity between two trajectories at t is determined by their distance — denote it by $\mathtt{d}(\mathbf{p}_1(t)\,, \mathbf{p}_2(t))$. We set the distance function to be the channel utilization, which is also our optimization metric. Denote the channel utilization at $\mathbf{p}(t)$ by $\mathcal{U}(\mathbf{p}(t))$. Now the distance between two trajectories at t is defined as the difference of their corresponding channel utilizations,

$$\mathtt{d}(\mathbf{p}_1(t)\,,\mathbf{p}_2(t)) = \mathcal{U}(\mathbf{p}_1(t)) - \mathcal{U}(\mathbf{p}_2(t))\,. \tag{6.4}$$

Extending above to trajectories spanning in $[t_1, t_2]$,

$$\langle \mathtt{d}(\mathbf{p}_1(t)\,,\mathbf{p}_2(t))\rangle = \langle \mathcal{U}(\mathbf{p}_1(t))\rangle - \langle \mathcal{U}(\mathbf{p}_2(t))\rangle, \tag{6.5}$$

where

$$\langle \mathtt{d}(\mathbf{p}_1(t)\,,\mathbf{p}_2(t))\rangle = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} \mathtt{d}(\mathbf{p}_1(t)\,,\mathbf{p}_2(t)) = \mathtt{d}(\mathtt{P}_1(t_1,t_2)\,,\mathtt{P}_2(t_1,t_2)) \tag{6.6}$$

and

$$\langle \mathcal{U}(\mathbf{p}(t))\rangle = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} \mathcal{U}(\mathbf{p}(t)) = \mathcal{U}(\mathtt{P}(t_1,t_2)) \tag{6.7}$$

are the time averages of their functions. Now

$$\mathtt{d}(\mathtt{P}_1(t_1,t_2)\,,\mathtt{P}_2(t_1,t_2)) = \mathcal{U}(\mathtt{P}_1(t_1,t_2)) - \mathcal{U}(\mathtt{P}_2(t_1,t_2))\,. \tag{6.8}$$

The research question in this chapter can be addressed by obtaining a trajectory that is closer to the ideal trajectory as compared to the static trajectory. Thus we hypothesize that

$$\begin{aligned} &\mathtt{d}(\mathtt{P}^{\diamond}(t_1,t_2)\,,\mathtt{P}^{*}(t_1,t_2)) < \mathtt{d}(\mathtt{P}^{\diamond}(t_1,t_2)\,,\mathtt{P}^{\bullet}(t_1,t_2)) \\ &\implies \mathcal{U}(\mathtt{P}^{*}(t_1,t_2)) > \mathcal{U}(\mathtt{P}^{\bullet}(t_1,t_2))\,. \end{aligned} \tag{6.9}$$

Our system model presents a channel utilization improvement seeking predictive control to obtain the desired dynamic trajectory.

6.3 System model

The presented model is regulated by a three phase functionality that is perpetually invoked in a closed loop: sensing, decision and adaptation (SDA) (cf. Figure 6.1).

This model is based on the same assumptions as stated for the stochastic models in Chapter 4 (cf. Section 4.2). We set off by categorizing each application in our system, as either inelastic or elastic. Presumably, the QoS requirements of the inelastic applications that we consider, fluctuate with time. Based on the type of application hosted by a transmitter node, it is distinguished accordingly. This allows us to form two node index sets. A transmitter node that hosts an elastic application belongs to a set, $\mathcal{K}^{\mathrm{el}}$, while the one hosting an inelastic application, belongs to a set, $\mathcal{K}^{\mathrm{iel}}$.

$$\begin{gathered} \mathcal{K}^{\mathrm{iel}}, \mathcal{K}^{\mathrm{el}} \subseteq \mathcal{K}, \\ \mathcal{K}^{\mathrm{iel}} \bigcup \mathcal{K}^{\mathrm{el}} = \mathcal{K}, \end{gathered}$$

where $\mathcal{K}$, defined in (2.3), is an index set of MAC processes in the network. The nodes in $\mathcal{K}^{\mathrm{el}}$ and $\mathcal{K}^{\mathrm{iel}}$ are termed as elastic traffic nodes and inelastic traffic nodes, respectively. If a transmitter node hosts at least one application from each category, then it is termed as a mixed traffic node.

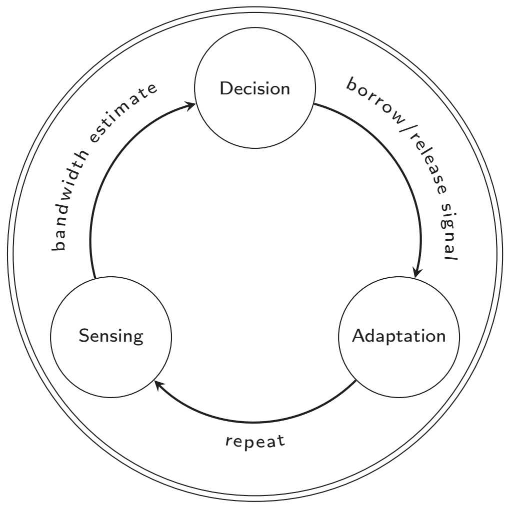


Figure 6.1: Control loop: sensing, decision and adaptation.

In our model, an active node is one that listens to the network to estimate the bandwidth usage over time, of the inelastic traffic nodes that are considered as passive nodes. An active node, $el \in \mathcal{K}^{\mathrm{el}}$, interested in a passive node, $iel \in \mathcal{K}^{\mathrm{iel}}$, maintains related data in a local monitor, $\mathcal{M}_{el}^{iel}$.

Each node arbitrates to the network using its MAC process, which is configured by settings of its respective P-tuple, defined in (2.1). The P-tuple of node $j, j \in \mathcal{K}$ is denoted by p_j. Without loosing generality, it is assumed that an inelastic application is considered more important than an elastic application, hence it has higher priority. A priority order can be specified using the priority semantics (cf. Section 2.6), so that by (2.20), $\mathbf{s}_{iel} \cdot \mathbf{g} > \mathbf{s}_{el} \cdot \mathbf{g}$. SDA control strategy for a monitor, $\mathcal{M}_{el}^{iel}$, is given in Algorithm 7 — explanations next.

6.3.1 Sensing

All nodes operate in a shared broadcast medium, so that a local monitor at an active node can mark the channel busy state due to the traffic generated by one or more target passive nodes, which can be distinguished by their MAC addresses. This allows bandwidth estimation process, carried out as follows.

A monitor, $\mathcal{M}_{el}^{iel}$, at an active node records periodically the channel usage time due to the target passive node, during fixed time intervals, each one of duration δ. The channel usage time during a given δ is termed as a sample. Successive samples are ordered chronologically. The $i^{\mathrm{th}}, i \in \mathbb{N}_0$ sample, $\mathrm{s}_{el}^{iel}(i)$, expresses the channel 'busyness' during δ, due to the target passive node: $\mathrm{s} : \mathcal{K}^{\mathrm{iel}} \times \mathcal{K}^{\mathrm{el}} \times \mathbb{N}_0 \to [0, \delta]$. The time overheads of transmission transactions (cf. Definition 2.7.1) during the sample interval are counted towards the channel usage by the target passive node. The interframe spaces of EDCA are used to delimit the boundaries of the channel usage, as obeyed by all nodes. Alternatively, NAV can be used for this purpose.

6.3.2 Decision

At the end of each sample period, a decision is taken regarding channel overbooking, which is considered as a bandwidth wastage. When considered appropriate, this decision may lead

Algorithm 7 $\texttt{SDA}(el, iel, \delta, \kappa, d, time)$
// SDA predictive control

Require: $el \in \mathcal{K}^{\text{el}}, iel \in \mathcal{K}^{\text{iel}}, \delta \in \mathbb{R}, \kappa \in \mathbb{R}, d \in \mathbb{N}_1, time \in \mathbb{R}$
Ensure: Dynamic trajectory

```
// initialization
p'_el ← p_el; // preserve original priority of active node
i ← 1; // initialize sample index

do
    // sensing
    take a sample, s_el^iel(i);

    // decision
```

// compute threshold

$$\tau_{el}^{iel} = \underset{\max(i-d,0)<j\leqslant i}{\uparrow} \frac{s_{el}^{iel}(i)}{\delta} \cdot \kappa \text{ ; // (6.11) restated}$$

// decide for adaptation

$$\varrho_{el}^{iel}(i) = \begin{cases} 0 & \frac{s_{el}^{iel}(i)}{\delta} \geqslant \tau_{el}^{iel} \\ 1 & \frac{s_{el}^{iel}(i)}{\delta} < \tau_{el}^{iel} \end{cases} \text{ ; // (6.10) restated}$$

// adaptation

$$p_{el} \leftarrow \begin{cases} p_{iel} & \varrho_{el}^{iel}(i) \\ p'_{el} & \neg\varrho_{el}^{iel}(i) \end{cases} ;$$

```
    i ← i + 1;
while (i ⩽ time/δ)

return
```

to reallocation of shares between the nodes, with an aim to avoid this bandwidth wastage. This decision is taken by defining a predicate, $\varrho_{el}^{iel}(i)$, which allows reallocation only when a given sample falls below a threshold value, τ_{el}^{iel}:

$$\varrho_{el}^{iel}(i) = \begin{cases} 0 & \frac{\mathtt{s}_{el}^{iel}(i)}{\delta} \geqslant \tau_{el}^{iel} \\ 1 & \frac{\mathtt{s}_{el}^{iel}(i)}{\delta} < \tau_{el}^{iel} \end{cases} . \tag{6.10}$$

The term $\frac{\mathtt{s}_{el}^{iel}(i)}{\delta}$ represents a normalized channel usage during the sample period.

The threshold value is adaptive to the application behavior at the target passive node. This adaptation considers the irregular and fluctuating behavior of application traffic that is sensed over short time intervals. Selecting too short time intervals, may report spiky behavior which may be only momentous and hence, misleading. Decisions based on observations in such situations, have to be avoided. Therefore, the traffic pattern is captured over another independent dimension, samples history. In this approach, a window over the last d samples is maintained at the monitor. This window moves with time. Based on this information, the threshold, τ_{el}^{iel}, is estimated by down scaling maximum of the normalized channel usage within the current window by a factor, κ:

$$\tau_{el}^{iel} = \mathop{\uparrow}_{\mathtt{max}(i-d,0)<j\leqslant i} \frac{\mathtt{s}_{el}^{iel}(i)}{\delta} \cdot \kappa, 0 < \kappa \leqslant 1 . \tag{6.11}$$

κ is used to tradeoff between too optimistic/pessimistic decisions.

6.3.3 Adaptation

At the end of each sample period, a decision for adaptation is taken for the reallocation of shares. If an adaptation is required, it is carried out by changing the P-tuple of the MAC process at the active node, p_{el}.

Whenever the predicate, $\varrho_{el}^{iel}(i)$, holds, it triggers an adaptation process at the active node. During this process, the channel arbitration priority of the MAC process at the active node is increased to a limit. This limit is the channel arbitration priority of the MAC process at the passive node being monitored. The selection of this limit minimizes two desired model aspects: one, the chances of failure for the demands of the inelastic application at the passive node, due to impact of the current decision. By failure, we imply a demand that exceeds the allocated channel during the next sample period; two, the channel wastage in the temporal locality. Equating priorities of the MAC processes for inelastic/elastic applications is expressed as, $p_{el} \leftarrow p_{iel}$. Before this priority assignment, the original priority is also kept, $p'_{el} \leftarrow p_{el}$. Otherwise, whenever the predicate, $\varrho_{el}^{iel}(i)$, does not hold, it signals an increased demand from the inelastic application during the last sample period. This signal is intercepted as to lower the priority of the MAC process at the active node, to its original value, $p_{el} \leftarrow p'_{el}$. This process is termed as priority restoration, which allows preferential treatment of the inelastic application in the current sample period. The priority restoration is neither needed nor carried out if the predicate, $\varrho_{el}^{iel}(i)$, does not hold in succession, which reflects continuous high demand from the inelastic application. The number of such successive monotonous decisions is directly related to the interval of continuous high demands.

6.3.4 Channel utilization

The performance of the dynamic trajectory obtained from the SDA model is evaluated using our hypothesis set in (6.9). It is based on channel utilization that is defined in Chapter 5 (cf. Section 5.3) as an aggregation of the expected channel shares of all MAC processes in the network.

$$\mathcal{U} = \sum_{k \in \mathcal{K}} \theta_k \ . \tag{6.12}$$

For convenience, we denote $\mathcal{U}(\mathtt{P}^*(t_1, t_2))$ and $\mathcal{U}(\mathtt{P}^\bullet(t_1, t_2))$ of (6.9) by $\mathcal{U}^*$ and $\mathcal{U}^\bullet$, respectively. In our validation tests, we establish $\mathcal{U}^* > \mathcal{U}^\bullet$. The time integral in (6.6) assumes continuous intervals for trajectories. They will be discretized by periodic sampling.

6.4 Validation

The SDA model is assessed and validated by simulations performed in GRADES [5]. Preference of GRADES over ns2 [199] is discussed in Section 4.9. The reported results are drawn from two QoS based ad-hoc network configurations in the general system setup.

6.4.1 General system setup

In order to highlight the channel utilization improvement by the SDA model, a small network comprising of two applications is constituted. These applications are represented by two MAC classes of one member each. The transmitters in Figure 6.2 represent two ad-hoc connections, one of which represent a passive node streaming a video, while the other represent an active node running a file download application using File Transfer Protocol (FTP). The MAC process for the video is assigned a higher priority than the MAC process for the file download. We mostly use the configurations from the general setup in Chapter 4 (cf. Section 4.6.1). When a parameter is changed for the SDA model, it is specified accordingly.

Priority levels According to our description, $\mathcal{Q} = \{0, 1\}$ and $\mathbf{n} = \{1, 1\}$, so that $\mathcal{K} = \{1, 2\}$.

Network mode The network is operated in an ad-hoc mode, assuming that all devices are in the broadcast range of each other.

Channel access mechanism EDCA is used to access the channel at the MAC sublayer, whose parameters draw values from the following sets of values, for obtaining all results:

- $AIFSN \in \{1, \cdots, 3\}$;
- $CW \in \{15, \cdots, 511\}$. CW grows until three steps;
- $TXOP \in \{1\}$.

OSI parameters The PHY/MAC parameters of IEEE 802.11b [92], listed in Table 4.2, are selected, except for the application payload limit in bytes per MPDU vector, $\mathbf{L} = [960, 960]$ for MAC processes at the passive node and the active node.

Simulation time Excluding the initial start-up and final wind-off times, all simulations are run for 10 seconds, which is considered enough to exhibit the fluctuating requirements of the video and the improvements due to the SDA model. The simulation time determines the invocation time, $time = 10$ of Algorithm 7.

Video traffic model The video traces of Tokyo Olympics are taken to model H.264 [78] frame sizes [177, 209]. Due to variability in MAC service rates and MPDU sizes of the two applications, there are no unwanted 'synchronization' effects within the simulations.

Sample period The bandwidth is sampled at every $\delta = 5$ ms. In accordance with aim d. in Section 6.1, selection of the sample period is based on keeping preciseness that holds for various application types, including streaming services.

Sample window The sample window of (6.11) is set as $d = 10$.

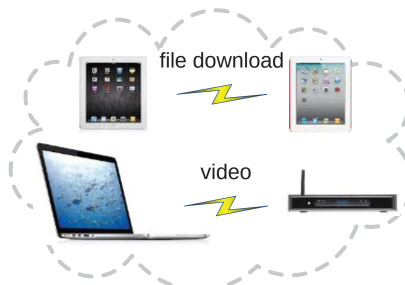


Figure 6.2: QoS based autonomic self-optimization system setup with four devices on the same ad-hoc network, arranged in two transmitter/receiver pairs.

6.4.2 Experiments, results and discussion

The SDA model is tested by defining two *gaining control* configurations for the experiments, each one exercising a different priority arbitration — their settings are given in Table 6.1. For all experiments, $RetryLimit = 7$, across the board. Each plot presents the individual shares of the file download application and the video along with the channel utilization, both for the statically configured MAC without the SDA model and the dynamically configured MAC with the SDA model. The results due to static configurations are marked with a superscript $*$ in the plot legends. Algorithm 7 is invoked as $\texttt{SDA}(2, 1, 0.005, \kappa, 10, 10)$.

6.4.2.1 $AIFSN$ driven arbitration

The $AIFSN$ driven arbitration scheme conforms to the MAC settings in (4.25) (cf. Section 4.6.2.1). According to this relation, classes have identical sets for CW but different settings for $AIFSN$. The network configurations used for the experiments are given in Table 6.1a. The results are reported in Figure 6.3a for $\kappa = 0.75$ of (6.11), the tradeoff parameter. On the onset, following two observations are noticable:

- The file download application share and the channel utilization with the SDA model mostly dominate their counterparts without the SDA model, thus showing the effectiveness in terms of the improvement in the channel utilization of a dynamic scheme such as proposed in the SDA model. The arbitration strengths of the two applications are selected to be not far apart, to study the effectiveness of the SDA model in comparable MAC competition. An improvement in the channel utilization is found to be 1.7% over the static case. Whenever results in the static configuration dominate the results in the dynamic configuration, it spots instances of delayed bandwidth return to the video node. This delay is dependent on the sample period, δ, which controls sensing the exact moment of returning the borrowed bandwidth. Hence, the SDA model performance indeed varies with δ, though a very small setting might widely and too frequently change the threshold,

Table 6.1: *Gaining control* configurations

(a) $AIFSN$ driven arbitration

parameter	configuration
$AIFSN$	$[1, 3]$
CW	$\begin{bmatrix} \mathbf{31}, 31, 31, 31, 31, 31, 31, 31 \\ \mathbf{31}, 31, 31, 31, 31, 31, 31, 31 \end{bmatrix}$
$TXOPLimit$	$[1, 1]$

(b) CW driven arbitration

parameter	configuration
$AIFSN$	$[1, 1]$
CW	$\begin{bmatrix} \mathbf{15}, 31, 63, 127, 255, 255, 255, 255 \\ \mathbf{31}, 63, 127, 255, 511, 511, 511, 511 \end{bmatrix}$
$TXOPLimit$	$[1, 1]$

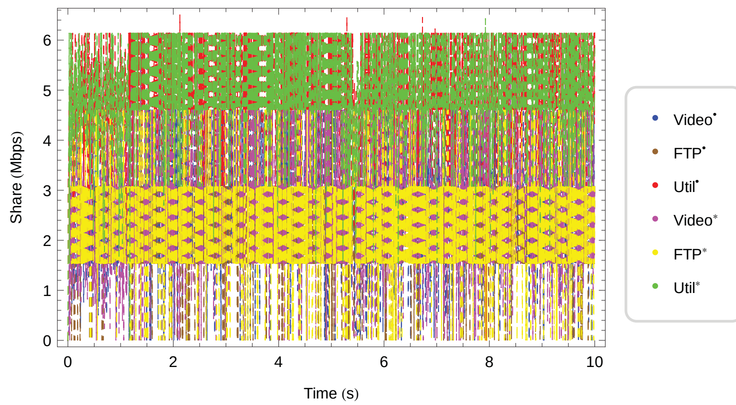
τ_{el}^{iel} in (6.11). Thus, the SDA model might anticipate and capture small bandwidth improvement opportunities, though such optimistic decisions might have a shortfall over longer periods. Even such optimistic decisions might lead to increased contention in the system, which incurs more collisions, hence bandwidth wastage.

- The SDA model has no effect on the high priority traffic, as evident from the video shares, which are fulfilled on demand.

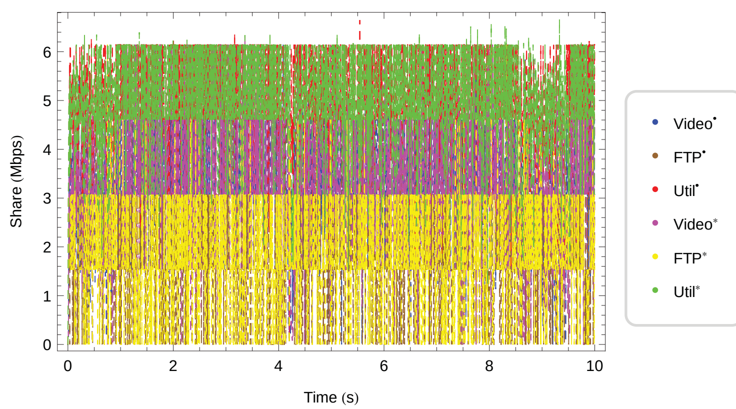
6.4.2.2 CW driven arbitration

The CW driven arbitration scheme conforms to the MAC settings in (4.26) (cf. Section 4.6.2.2). According to this relation, classes have identical sets for $AIFSN$ but different settings for CW. The network configurations used for the experiments are given in Table 6.1b. Again, the arbitration strengths of the two applications are comparable. The results, reported in Figure 6.3a for $\kappa = 0.65$ of (6.11), adhere to the two observations presented in the $AIFSN$ driven arbitration case, though the improvement in the channel utilization is less than the $AIFSN$ case, 1.0%.

This is due to the fact that any $AIFNS$ change impacts the system performance quickly and more frequently. Also that $AIFSN$ guarantees priority protection better than CW. Due to this reason, the tradeoff parameter, κ, is set to a lower value than the $AIFSN$ case.



(a) *Gaining control* – $AIFSN$ driven arbitration. Configurations in Tables 6.1a.



(b) *Gaining control* – CW driven arbitration. Configurations in Tables 6.1b.

Figure 6.3: Comparing the performance of dynamically configured MAC using the SDA model (dynamic trajectory) with a statically configured MAC (static trajectory). The former case has superscripts, $*$, in the legends, is distinguished from the latter case that has superscripts, $\bullet$. Channel utilization, video and FTP points due to the dynamic trajectory mostly dominate points due to the static trajectory.

6.5 Conclusions

The requirements of inelastic applications may vary over time, hence statically made pessimistic reservations on the channel to fulfill them, triggers suboptimality. These reservations are made through network configuration, which determine the channel utilization. This study addresses this suboptimality issue by making reservations that adjust to dynamic requirements.

We have demonstrated that resource management in a QoS controlled ad-hoc WLAN is possible by autonomously monitoring activities in the network — specifically transmissions on the channel. By allowing nodes to adapt their contention strength objectively, network optimality can be achieved.

This chapter presents formally a reactive model for distributed QoS control, wherein MAC configuration is done dynamically and autonomously by processes, with the aim to improve the channel utilization. The results demonstrate the advantage of such dynamic

MAC configuration approach over the static MAC configuration approach. Our dynamic approach has realized a closed-loop predictive control, which can exhibit instability — a well known characteristic of such controls [137, 60, 35, 218, 219].

The model is validated in two MAC *gaining control* configurations: $AIFSN$ driven arbitration and CW driven arbitration. Evaluating the effectiveness of our model in the two MAC arbitration schemes, the performance is comparable, though it is better in the $AIFSN$ case. Improvements of 1.7% and 1.0% are found possible for the $AIFSN$ and the CW cases, respectively. The higher rate in the $AIFSN$ case is because of the fact that the effect of the $AIFS$ deferral is activated more often than the effect of backoff slots in the CW. Thus, the MAC behavior adapts to the desired network configuration quicker in the $AIFSN$ case — in fact, the current $AIFSN$ change affects the share of the FTP node exactly when the channel is idle again. Consequently, the priority gain has a faster impact in favor of the FTP node. Another important observation is regarding preservation of the video priority. In both schemes, its shares are preserved.

The proposed model estimates short-term and long-term fluctuations, so as to make optimal MAC reservations accordingly. Staying on this optimal trajectory is not always possible. The results reflect an overall improvement considered over the durations of the experiments. The extent of the improvement is dominantly determined by three factors: one, the profile of applications with pessimistic reservations; two, the extent of network suboptimality, which in our tests is not much because of comparable MAC arbitration strengths, hence lower rates of improvement; three, the size of time windows for observing the short-term and the long-term fluctuations.

This study highlights unnoticed wastage of channel in static reservations, which can be avoided. In an ad-hoc network scenario, the network performance improvement due to such reactive systems is significant, especially in high rate WLANs.

7

Conclusions and recommendations

Provisioning of QoS in wireless networks to deliver desired performance of applications is arduous. It becomes even more challenging if the considered network lacks a central controller. This thesis has addressed the problem of QoS-aware channel bandwidth reservation in such unmanaged wireless networks — this problem is termed as the QoS challenge. Research in this area is motivated by the realization of a lack of a QoS based predictive control to manage channel allocation. The existing IEEE 802.11 standard [92] has introduced QoS classes in the preferential treatment context, using a single recommended network configuration point, termed as default setting. We have identified the default setting to be insufficient and incomplete in the following sense:

- The set of QoS requirements of applications that is satisfied by the default setting, is absent;
- The QoS properties of applications that emerge as a result of this default setting, are not mentioned.

To overcome these two deficiencies, we have provided functions, which together map the given QoS requirements of applications to some network configuration point, ensuring their satisfaction. Thus, our approach towards the QoS problem allows the desired manageability of the channel. The problem statement of this thesis poses three challenges:

- How to specify the communication related QoS requirements of applications?
- In order to fulfill the communication related QoS requirements of applications, network resources have to be reserved through relevant enforcement mechanisms. Such reservations call for asserting a control over the network. Hence, the problem to address is — how to control the network behavior in a distributed manner?
- How to map the communication related QoS requirements of applications to some desired network behavior such that it can fulfill those requirements?

These challenges are further split into research questions. The first challenge concerns the specification of QoS requirements of applications. The second challenge considers investigations on achieving a distributed control of an unmanaged network. The third challenge concerns mapping of QoS requirements of applications to some network configuration point that satisfies them. These three challenges are coped with by offering a performance framework that admits quantitative analysis for predicting the network configuration to deliver the desired performance. Challenge specific contributions are described as follows.

In the area of QoS specification, we proposed the QoS semantics with focus on accommodating all main types of applications, representing various domains. Generality is maintained by allowing applications with distinguished QoS concerns, which are channel access frequency, channel share, MAC latency and MAC reliability. An integral part of the QoS semantics is a relation that captures satisfiability of given QoS requirements of applications by network configuration points. These points control MAC reservations distributively. In case of no solution due to QoS requirements that exceed the channel capacity, we propose priority semantics to order applications by their criticality. The aim is to renegotiate the QoS requirements on the network to have a solution.

In the area of distributed control, a conceptual model of a distributed network is introduced. The proposed model is descriptive enough and complies with IEEE 802.11 [92]. It abstracts MAC arbitration mechanisms into two coherent system concerns, namely *gaining control* of the channel and *retaining control* of the channel. The identified concerns leverage identification of system cyclic behavior that facilitates development of quantitative analysis for system steady state performance. Based on this, analytical models are developed for QoS properties, one per QoS concern.

In the area of determining some network configuration point against a set of QoS requirements of applications, our proposed mapping generates a set of solutions. Based on their quality properties, some of these solutions are found to be optimal. They are collected in the set called Pareto front. These quality properties are mapped to by corresponding functions of network configuration points. Solutions in the Pareto front offer tradeoff points over multiple objectives. Out of the Pareto front, a global optimal solution is determined by formulating a single combined objective.

For all the problems tackled in this thesis, we have developed software tools. These include implementation of the proposed theoretical models to validate the presented analysis. The results are verified by developing a new custom made discrete event simulator, GRADES [5]. The work has resulted in related publications, given as a separate list. We leave this part with concluding remarks on our performance framework.

- Regarding the objectives set for it (cf. Section 2.2), we assess that all of them are met. Aims $1-4$ are approached with corresponding analytical models, while aim 5 is met only to an extent of formal specifications.

- The third challenge in the problem statement regarding function, mapping QoS requirements to some network configuration point, $\mathtt{F}(\cdot)$ (cf. Section 2.5.2) is resolved by considering a limited set of network configurations. Discretization of $\mathtt{F}(\cdot)$ is, thus constrained by the considered set. Consequently, determination of optimal choices as offered by the Pareto front and the global optimal network configuration point, exhibit strong dependence on the considered set. By its nature, the Pareto front based predictive approach is proactive, though it is able to find solutions of only limited regions of QoS requirements. In contrast to adaptive approaches that are considered reactive, we rank limitations in our approach to be the cost of early predictability.

7.1 A discussion on the research findings

This section expresses the findings and implications in context of contributions per research question.

Question 1. What is the conceptual model of the desired QoS based MAC reservations on the channel?

It has two subquestions:

a. How to abstract the MAC behavior towards a performance model?
b. How to specify the communication related QoS requirements of applications?

Addressing this question has resulted in defining our proposed performance framework. The necessary foundational concepts are presented, around which other questions are formulated and studied. Further details per subquestion are as follows.

a. The EDCA behavior is generalized to a simplified algorithm and a State Transition Diagram. This offers the understanding needed for system abstraction towards a performance model. Given that CSMA/CA based network protocols such as EDCA have unpredictable execution patterns, we transcend these by identifying system cyclic behavior, expressed as cycles. These cycles repeat transparently over specific MAC arbitration mechanisms and their settings. Such an approach has led us to derive performance metrics in the trace framework (research **Question 2**) and stochastic models related to achieving a distributed control of the network (research **Question 3** and research **Question 4**).

   In the literature, system cycle based approaches for performance derivation are reported [17, 168, 201], though the details captured and assumptions made vary in accordance with the IEEE 802.11 standard [92]. Our high level conceptual model, expressing cycles formally is new. It is based on identification of two fundamental system concerns, namely *gaining control* of the channel and *retaining control* of the channel. It supports prediction of short-term fluctuations in system QoS properties, which is one of the objectives of our performance framework. This support is demonstrated in the analysis leading to the theoretical tools in the trace framework. Predictability of short-term fluctuations in systems allows conciseness in provisioning of resources, in our case for instance, the MAC buffers — thus, offering cost-effective solutions. It also leverages isolation of extreme non-conformal system behaviors.

b. We have introduced the QoS semantics to specify the QoS requirements. To the best of our knowledge, our contributions are new in QoS-aware WLANs. The QoS semantics can accommodate applications with distinguished QoS concerns, thus covering major applications types. To judge quantitatively, if QoS requirements are fulfilled, we propose a satisfiability relation. Supplementary to the QoS semantics is our proposed priority semantics that can handle the issue of insufficiency in the network capacity. To this end, applications can be ordered by their criticality for admission control — thus lowering the QoS requirements on the network to permit their satisfaction.

Overall, the study of this question has revealed the following:

- Identification of system cyclic behavior facilities performance analysis to a great extent while keeping generality intact. Formal specification of cycles can be extended to accommodate more details, such as consecutive transmissions by a MAC process, reflecting MAC capture effect due to extreme asymmetric settings.
- Priority assignment and priority implementation are two separate issues, hence they should be treated accordingly within their usage context. The former refers to obtaining a preference order based on criticality. The latter refers to the settings of enforcements mechanisms as provided by a given technology. Concerning our introduced function for the

distributed control and mapping the QoS requirements of applications to network configuration points, there is no need to even consider any preferential order of applications. This is even true for most of resource reservation approaches applied to other technologies. As such, priority numbers are irrelevant, unless their implementation details towards resource reservations are known.

- Distinctive even functionally independent system partial behaviors, can be compared by appropriate functions — in our case, we proposed a weighted average of QoS concerns.
- A mapping problem can be solved by converting it into corresponding reverse mapping problem(s), especially if it is a transformation problem with discrete range space. Such spaces render infeasibility of approaches based on continuous spaces. Thus, a reverse mapping from such a discrete space can effectively discretize the original continuous range, to allow only reachable points.
- Bringing in more QoS concerns to specify the QoS requirements, restricts the solution space. Relaxation of some of the QoS concerns, can bring about potentially interesting network behaviors, towards opening of more optimization regions. Consider network behaviors that allow one-to-many relationships between QoS properties. For instance, two network configurations may result in nearly the same shares but different latencies or vice versa.

Question 2. How to track the EDCA network evolution over time?

It has four subquestions:

a. How to discover MAC states resulting from arbitrary subtraces of processes?
b. How to derive low-level performance metrics such as occurrence probabilities of process states, from an arbitrary subtrace of a process?
c. How to identify and specify process-specific and MAC event-specific cyclic behaviors in traces?
d. How to examine QoS properties for both, short-term and long-term durations, from an arbitrary subtrace of a process?

This question is addressed by our proposed trace framework. To the best of our knowledge, there is no other related work that focuses on trace analysis of CSMA/CA based MAC protocols. Our proposed theoretical tools allow to formally study the trace properties towards QoS based performance derivation over short-term and long-term durations. Detectability of QoS properties on short-term duration leverage ensuring compliance to requirements on QoS fluctuations. Contributions and findings per subquestion are as follows.

a. Given subtraces of processes collected independently, this subquestion studies their combined effect at the network level. To this end, we have determined a relation from states of processes to states of network using two independent approaches. The first approach, based on istate attributes, identifies more states of the network than the second approach, which is based on process istate transition rate. On the other hand, the second approach relies on system distributed view, hence needs less information. In general, distributed views lack information to deduce all system level behaviors.

b. Low-level details reflect more accurately on the system properties, while they can be used to derive high-level performance metrics, such as the QoS properties, which tend to abstract internal behaviors of a process. We record events with attributes to derive low-level performance metrics. The aim of studying such details from traces is to be able to verify stochastic models at the same level. These details are key indicators of process internal behaviors, and support validation of hypothesis set on EDCA mechanisms. For instance, CW settings influence the effect of $AIFS$ settings, and probability of collision changes with every retry attempt. Further, it also allows to check if stochastic models and simulation models conform to each other. For instance, an update of the backoff counter at the beginnnig/end of the PHY slot boundary has different performance implications. In general, capturing in-depth system details offers more performance metrics and isolation of unexpected system irregularities.

c. Studying the properties of recurrence patterns provides insights into QoS fluctuations. We introduce a formal querying system that can underpin performance fluctuations from multiple perspectives. These details support identification of performance anomalies — for instance, in hidden/exposed node situations.

d. We provide functions, one per QoS concern, mapping events in traces to QoS properties, derived on both short-term and long-term durations. Short-term QoS properties are transient behaviors. Knowing them allows to estimate over time, the demands on resources, such as MAC buffers. Such an estimate is more precise than an average made over a long-term duration. Predictability over small time windows can be followed by the actual resource usage over the same time scale, thus offering more cost-effective resource provisioning. Confidence in such a predictability, would still be questionable for randomness based systems.

Question 3. How to develop a quantitative model for network steady state?

A network steady state represents an average network behavior. It emerges as a combined effect of steady states of individual MAC processes. To this end, we develop a Markov based model similar to [201]. It maps network configurations to stationary probability distributions, one per QoS class of MAC processes. This Markov model is solved by a fixed point seeking approach based on our algorithm. The implementation performance of this algorithm is thoroughly reported in terms of memory required, iterations needed and time taken, for various network configurations.

Existing works in literature in this domain can be classified in three categories in an increasing order of details captured and realistic assumptions: slot-based [17, 216], zone-based [168, 211] and istate-based [201]. Due to objectives of our performance framework, we adopted an istate-based approach, which stays valid in extreme network configurations. The Markov model is validated over an extensive set of four priority classes setups, exercising different arbitration mechanisms. Some observations are as follows.

- The distribution of probability masses, as evident from the probability distributions of processes, is related to the arbitration mechanisms driving the priority. The extent of this effect is related to the network configurations used. In general, lower backoff/retry counter values have high masses. Both $AIFS$ and CW driven arbitration schemes assure preferential treatment, though its extent is only relative to the system settings. The order of invocation of $AIFS$ and CW is related to the ability of effecting each others impact: earlier in order can not impact those later. Only CW settings can effect the impact of $AIFS$.

- Our software tool for the Markov model allows an accelerated exploration of the network configuration space. Capturing more details and taking realistic assumptions brought in complexity, which is not affected by the selected network configurations. In comparison to the slot-based [17] approaches, ours is more complex but is comparable to the zone-based [168] approaches, which even suffer from scalability and accuracy issues.

- The memory requirements of our fixed point seeking algorithm are only related to the system CW settings. Based on the number of iterations needed to converge, system is more stable for higher CW settings. The time of convergence is polynomially related to the memory requirements.

- To realize our algorithm for on-the-fly computations in an operational system, results suggest a need for parallelized hardware.

- Time restricted execution of our algorithm is possible by relaxing the convergence criteria. We consider it as a trafeoff between solution time and error corresponding to system instability.

Question 4. How to develop quantitative models for QoS properties of MAC processes?

For this question, we derive functions, one per QoS concern, mapping network steady states to QoS properties. Most of the works reported in literature, derive shares and latencies based on availability of infinite retry attempts to the MAC processes. We avoid this oversimplified and unrealistic assumption. The results demonstrate that cases of finite retry attempts, especially the low range of values, influence the QoS properties more profoundly as compared to corresponding cases of infinite retry attempts. This highlights the need of a realistic assumption when model accuracy is a concern.

During the study of this question, we also considered controlling QoS performance fluctuations due to inherent randomness in MAC protocols, such as EDCA. Without such a control, for instance, delay on each MPDU can not be bounded. Hence, the usage of these protocols becomes questionable in transmitting traffic with hard real time requirements. This type of traffic has an increasing usage in, for instance, industrial and medical environments. Existing literature in this context reports usage of arbitration mechanisms and simulation studies to achieve absolute priority treatment to each MPDU in infrastructure WLANs [121, 190], but without relevant models. To this end, we have developed a quantitative model for the worst case delay an MPDU can experience in an ad-hoc network. This model is based on $AIFS$ mechanism that can be used to mandatorily defer the transmissions by competing processes. We take into account, the *retaining control* system concern, thus keeping the model more general. Investigation of this question provide the following insights:

- Except MAC latency, which is increasing, channel access frequency and channel share are decreasing with $AIFS$ and CW. MAC reliability, as expected is only related to allowed retry attempts. On the other hand, $TXOP$, influencing the system concern, *retaining control*, has a negative/positive/negative impact on frequency/share/latency, respectively. A system studied long enough to be considered stable, $TXOP$ does not influence reliability.

- A MAC latency model that does not consider MAC losses, gives pessimistic results. It is because of additional delays of intermediate MPDUs that are dropped on exhaustion of retry attempts. For time sensitive traffic, such as video frames, it implies lower video QoS prediction on the receiver side of the link.

- More realistic latency prediction offers better management decisions for usage of limited resources, such as MAC buffers.

- To cater for hard real time traffic requirements, our worst case delay model calls for permanent MAC based reservations through $AIFS$. In case of no hard real time traffic, the reserved slots are wasted, thus channel utilization is decreased. This suggests modelling dynamic network configuration on demand, in the sense whenever hard real time traffic appears.

- For the worst case delay model, the PHY channel is split into two logical channels on time sharing: stochastic priority channel (*spc*) and an absolute priority channel (*apc*). The performance of traffic in *spc* is directly affected by the busyness of *apc*. In fact a high priority subchannel in *apc* affects all those with lower priority, whether they be in *apc* or not. This model offers bounded delays only for the highest priority traffic.

Question 5. How to map the communication related QoS requirements of applications to some desired network behavior that fulfills those requirements?

Prior to addressing this question, we have formally specified a set of QoS requirements of applications within the QoS context of our system, a relation to ascertain on their satisfiability by network configurations and obtained a distributed control over QoS properties of processes. This allows us to define a function, mapping the QoS requirements of applications to some network configuration point. A side effect of our proposed satisfiability relation is multiple solutions, out of which the optimal ones are selected based on their quality properties. While the optimal set offers tradeoff points, we formulate a distance function, argument of whose minimal value is taken to be the best solution, termed as the global optimal solution.

Prior works in this problem domain have presented reactive [122, 12] and proactive [116, 178] approaches. Most of these approaches are not QoS requirements driven. They are characterized by their consideration of, what is termed by us, implicit QoS requirements. These imply optimization seeking approaches, which whenever offer solution(s), they are not guaranteed to fulfil a set of given QoS requirements. Sometimes, this term implies only performance curves that can fulfil unknown QoS requirements. Contrary to this, our approach can determine solutions for explicit QoS requirements. Investigations in this problem area brought us the following insights:

- To tackle this problem, we seek a transformation, from QoS space to network space. Due to technology limitations, not all points in the transformation range are accessible, hence we deem them infeasible in that sense. Any approach giving unreachable points, considered as infeasible, does not serve the purpose. These infeasible points have to be appropriately dealt with by shifting towards a feasible point. This may have side effects, such as no solution or need to cope with multiple solutions. Contrary to this, our proposed satisfiability relation based approach avoids the unreachable point situation.

- The Pareto analysis is carried out in a space of technical properties of the system. More spaces can be formulated that consider other strategic and management aspects.

- Other than applying the Pareto analysis to determine a single solution, we have also considered another method, gradient descent, which requires a continuous space. To this end, inapplicability of this method confirms the issue of unreachable points.

- We carried out a suitability assessment of our approach in terms of chances of finding a solution to a given set of QoS requirements. Indeed, these chances can be increased by building a larger space of decision vectors. In context of network parameter ranges as selected for our experiments, the largest space would have $156,065,000$ decision vectors. Computing this space seems feasible, however it is too detailed.

- We have also assessed the suitability of the default EDCA parameter set to satisfy a given set of QoS requirements. The result indicates weak chances, hence renders it inapplicable for solving most of the QoS requirements. Assessment also confirms our initial identification about its incapability. In fact, out of three experiments carried out, none of the solutions are provided by it.

Investigations into the QoS challenge is the functional requirement of the research undertaken in this thesis. The quality of meeting the QoS challenge by our proposed approach can be assessed by an extra-functional property of the system, namely the channel utilization. Improving channel utilization in resource scarce systems has significance in multiple dimensions such as constraint/user satisfaction problems and business optimization. Thus, we are also led into exploring channel utilization optimization. Research **Question 6** and research **Question 7** are formulated and studied in this context.

Question 6. How to improve the channel utilization for the wireless network under control of a distributed MAC, by exploiting characteristics of applications?

To address this question, we consider a proactive approach, wherein MAC decisions are enriched with the knowledge of applications. The approach is policy-based, with the assumption that the policy is already propagated to all MAC processes in the network. The impact of various policies is studied and their comparison is made. Channel utilization improvements in the range $4\% - 18\%$ are observed, which we associate to wastage avoidance from excessive contention slots and collisions. Due to practical limitations of networks, the extent of avoidance might be less, hence we estimate an average rate, 11%.

In contrast to works reported in literature [178, 146, 85], we consider two aspects together: QoS mapping and optimality of mapping. Investigations provide the following insights:

- The gain in channel utilization is related to the application property.

- Excessive retention durations may have negative delay impact on competing applications, though channel utilization is improved. This suggests a delay-aware policy, to the effect of fulfilling more sets of QoS requirements, though with less improvement on channel utilization.

- In erroneous channels, the impact of any policy would drop, approaching with the extent of errors, to a case of transmitting one MPDU per transmission opportunity.

- The proposed approach is based on accessing information of two OSI layers: the application layer and the MAC layer. Consequently, the independence of MAC decisions is affected. To keep independence, an application unaware, delay-aware MAC policy can offer allowed retention durations. It may result in application packets split across multiple transmission opportunities, or data from multiple application packets multiplexed in a single transmission opportunity.

Question 7. How to adapt distributed control of MAC autonomously, to avoid channel wastage due to pessimistic reservations?

To address this question, we consider a reactive approach, wherein MAC processes monitor channel utilization independently and autonomously, with an aim to estimate pessimistic reservations by their competitor processes. These pessimistic reservations are considered to be channel wastage. On detection of channel wastage, MAC processes can autonomously adapt their channel arbitration strength with an aim to use available slots. From the system perspective, network configurations are dynamically updated with an aim to improve channel utilization. Channel utilization improvements of $1.0\% - 1.7\%$ are observed for the settings we tested. These settings are close to the optimal settings, hence they determine lower limits of improvement. Further insights are as follows:

- Our proposed approach seeks system optimization by predicting channel usage over small time windows. The resultant MAC reservations should not over/under provision, ideally. The extent of optimization is dominantly determined by two factors: one, the properties of applications with pessimistic reservations; two, the extent of network suboptimality as ascertained by the network configuration.

- Our channel usage estimation algorithm aims for short-term and long-term fluctuations. These two durations, are yet another factor that influence the extent of optimization achieved. For short-term duration, too small time windows may easily push the system into an oscillating behavior, while for long-term duration, too large time windows may already average out improvement opportunities.

- It would be interesting to apply this approach towards studying an improvement in video QoS.

- In case multiple nodes assess and react to pessimistic reservations by a single node, probably our proposed approach is insufficient. There might be a need to distribute among the nodes, the bandwidth gain.

7.2 Recommendations for future research

Since the turn of the 21st century, wireless technology has emerged as a game changer. It has effectively changed the world. Wireless networks offer cost-effective solutions, enhance reachability to information and allow access to remote areas previously held back by technical or financial limitations. Given that QoS has a key role in any network technology, the provisioning of QoS in unmanaged wireless networks is poised to rise in the coming years. Despite relevant advancements in QoS for networks, IP QoS still challenges the scientific community. Based on the work done in this thesis, we identify a few QoS-related open issues that need investigations.

7.2.1 Open issue – mathematical

In general, our fixed point driven mapping from network configurations to the system steady states that are represented by stationary probability distributions, exhibits contraction. For a few network configurations in the default EDCA parameter set case, we observed system bouncing back and forth around the fixed points (cf. Section 4.7.3.4). These network configurations, suggesting points of instability, are already pointed out. It would be interesting to study mathematical properties of our system at these outlier configurations, where it alternates between contraction and expansion.

7.2.2 Open issue – modelling short-term fluctuations in QoS properties

The analysis for the stochastic behavior of the system presented in Chapter 4 assumed independence of MAC system cycles (cf. Definition 2.4.1). It would be interesting to lift this assumption to define correlated MAC system cycles. They reflect on short-term fluctuations in QoS properties due to possible MAC capture effects [201].

7.2.3 Open issue – modelling unsaturated networks

The second challenge in the problem statement is addressed by deriving a model for a distributed control of wireless unmanaged networks with saturated MAC queues. This assumption on the MAC queues turned out to be critical, resulting in limited applicability of the analysis for the case of unsaturated MAC queues. Hence, the motivation is to model distributed control for unmanaged wireless networks with unsaturated MAC queues. We aim to derive a schedule for applications that is optimized on low average delay and low delay variation. The model will be based on delay analysis. Specifically we intend to study the following research question:

Question How to derive best, average and worst delays for a set of applications in an unsaturated wireless network?

Thus we recommend future extension of our work in this direction. To this end, an outlook on our preliminary approach on delay analysis is presented in Appendix A.

7.2.4 Open issue – technological

Wireless devices are ever increasing, so is the application traffic, which is projected to have a 13-fold increase from 2012 to 2017 [53]. Anticipating that WLANs will have a pivotal role in the future wireless communication systems, a quest for next generation gigabit wireless (WiGig) technologies continues [210, 55, 40]. To this end, IEEE 802.11ad and 802.11ac working groups [90] already aim for WiGig. The insights gained from our work on QoS, should encourage development of a similar QoS-aware performance framework for WiGig.

7.2.5 Open issue – modelling distributed control of multihop wireless ad-hoc networks

Finally, we propose extension of our distributed control model towards multihop wireless ad-hoc networks. They are characterized by overlapping regions at network boundaries where hidden/exposed node problems give rise to asymmetric QoS behaviors [72].

7.3 Final remarks

This thesis has addressed the QoS challenge set in Chapter 1. The issue with QoS is management, for which we have obtained a predictive distributed control over wireless networks. Amidst the continuing vicious battle between ever-hungry applications and emergence of new wireless technologies, theory on QoS-aware wireless networks is under focus now. The significance of QoS provisioning in wireless technologies is enormous, though this parallel race between the ever-increasing QoS requirements of applications and new wireless technologies promising higher data transmission rates is not allowing a settlement.

In an era of constant change such as the current one, many approaches can be taken to address the QoS problem. Our approach is based on theory, which is verified by simulations. However, there are other research groups that obtain data from wireless test beds, which is more realistic. Collaborations between theory based and test bed based groups, will offer benefits to both — theoretical results in form of toolboxes, can be verified by data obtained from test beds, while more realistic theories can be aimed at by sharing data collected from test beds.

A

Unsaturated network delay predictor

A.1 System model

Our system is composed of N periodic applications that are indexed by a set, $\mathcal{K}$, and a single domain wireless network upon which applications compete for access.

$$\mathcal{K} \triangleq \{1, 2, 3, \cdots, N\}\,. \tag{A.1}$$

To proceed with analysis, we tag an arbitrary application to be the reference application. OSI layers processes corresponding to the reference application are termed accordingly. Within our current scope, these are reference transport process and reference MAC process. The requirements of reference application, k, are expressed in terms of its period, d_k, and message length, m_k. We assume that applications establish a transport connection over a wireless unmanaged network of MAC processes, which are configured according to their MAC settings. The requirements of applications are translated into lower layers specifications, till MAC, where performance metrics for delay are derived. Application performance metrics are determined by aggregating MAC performance metrics.

Application to transport layer mapping

Payload of the reference application is transmitted in transport layer packets, termed as data segments, each one with an application payload limit that we denote by G_k bytes. The delivery of data segments is acknowledged by transport layer ACKs, termed as control segments. Thus an application payload, m_k, is split into $\left\lceil \frac{m_k}{G_k} \right\rceil$ segments, of which $\left\lfloor \frac{m_k}{G_k} \right\rfloor$ are full while the last one carries $m_k - G_k \cdot \left\lfloor \frac{m_k}{G_k} \right\rfloor$ application payload. If the transport connection is reliable, a transport layer ACK is needed for each segment separately or together for a sequence of segments. Let transport layer header, size of a transport layer ACK and length of sequence of segments be denoted by h^{tp}, ACK^{tp} and seq^{tp}, respectively.

Transport layer to MAC layer mapping

Similarly each segment is passed to DLL, which establishes a connection with an adjacent network node. Its MAC sublayer forms MPDUs, each one with a payload limit that we denote by L_k. In case of larger segments, they are split into $\left\lceil \frac{g_k}{L_k} \right\rceil$ MPDUs, where g_k is the payload reaching DLL — it includes the segment with its header, h^{tp} and IP header, h^{ip}. Out of these MPDUs, $\left\lfloor \frac{g_k}{L_k} \right\rfloor$ are full while the last one carries $g_k - L_k \cdot \left\lfloor \frac{g_k}{L_k} \right\rfloor$ segment payload. Recalling from Section 4.5, the time to transmit m bytes of application payload over DLL is given by $\mathtt{t}^{\text{mac}}(m)$ in (5.9). In this case, segment payload is transmitted over DLL, so that $\mathtt{t}^{\text{mac}}(g_k)$ and $\mathtt{t}^{\text{mac}}\big(ACK^{\text{tp}}\big)$ are the times to transmit the segment and its acknowledgement.

MAC arbitration

A brief recap of the MAC arbitration allows us to introduce new terms whenever needed for the current analysis. During its execution, the MAC reference process, may access the channel exclusively to transmit its MPDU, thus experiencing minimal delay. Contrary to this, if multiple MAC processes access the channel simultaneously, the situation is termed as a collision. All MAC processes involved in a given collision, have a failed transmission that incurs a wasted busy state on the channel. The situation is followed by a collision resolution process during which the colliding MAC processes attempt again after waiting longer but random time durations. The reference process may undergo multiple collisions before it either transmits or discards the MPDU. The collision resolution process after an i^{th} collision, is considered an order-i resolution. In this analysis, we restrict to only order-1 resolutions — also called first-order resolution. The restriction is justified by the fact that in a system with unsaturated MAC queues, the probability of collision is much lower than a system with saturated MAC queues. The size of a collision, c, is the number of MAC processes involved in it: $2 \leqslant c \leqslant N$. Collisions of large sizes are even less probable.

A.1.1 Performance metrics

The three delay types are defined for the reference application. Their derivation is based on mean field analysis. In general, the transport layer interleaves data segments and control segments to ensure an ordered data delivery service. At a given time, a source will have either of these two types of segments riding the channel. Time exclusiveness property allows us to assume that both segment types do not compete against each other for channel, rather segments from other sources. Thus, our model has a virtual source that can emit both types of segments. A model source represents corresponding application source, so that our system is composed of model sources. Each model source contends for the channel with a strength that we measure by the corresponding application period and the mean size of MPDU emitted. The contention strength is needed to quantify the MAC arbitration, which has a key contribution to delay variations — hence, required to estimate an average case.

The mean MPDU size, $\mathbb{E}(l_k)$, is determined as follows. Given application message, m_k, our model source fragments it into a sequence of MPDUs. Let $data_k$ and n_k denote total data sent in this sequence and the length of this sequence, then

$$\mathbb{E}(l_k) = \frac{data_k}{n_k}. \tag{A.2}$$

$data_k$ is composed of data segments and control segments, with each one including h^{tp}. As they are passed downwards, h^{ip} and h^{mac} are also added to them. A control segment is a data segment with zero payload: $ACK^{\text{tp}} = h^{\text{tp}}$. The process of m_k fragmentation into MPDUs must consider the data segment payload limit[1]. To proceed, let $\mathtt{f}(z_k, L_k)$ and $\mathtt{e}(z_k, L_k)$ map the data segment payload, z_k, and the MPDU payload limit, L_k, to total data sent in the related MPDUs and their count, respectively.

$$\begin{aligned} \mathtt{f}(z_k, L_k) &= z_k + h^{\mathrm{tp}} + h^{\mathrm{ip}} + \mathtt{e}(z_k, L_k) \cdot h^{\mathrm{mac}}, \\ \mathtt{e}(z_k, L_k) &= \left\lceil \frac{z_k + h^{\mathrm{tp}} + h^{\mathrm{ip}}}{L_k} \right\rceil \ . \end{aligned} \tag{A.3}$$

z_k represents,

- $\left\lfloor \frac{m_k}{G_k} \right\rfloor$ full data segments, each holding a payload G_k;
- at most one incomplete data segment, holding a payload $m_k - G_k \cdot \left\lfloor \frac{m_k}{G_k} \right\rfloor$;
- $\left\lceil \frac{m_k}{G_k} \right\rceil$ transport layer ACKs.

Thus,

$$\begin{aligned} data_k &= \left\lfloor \frac{m_k}{G_k} \right\rfloor \cdot \mathtt{f}(G_k, L_k) + \mathtt{f}\left(m_k - G_k \cdot \left\lfloor \frac{m_k}{G_k} \right\rfloor, L_k \right) \\ &\qquad + \left\lceil \frac{m_k}{G_k} \right\rceil \cdot \mathtt{f}(0, L_k)\,, \\ n_k &= \left\lfloor \frac{m_k}{G_k} \right\rfloor \cdot \mathtt{e}(G_k, L_k) + \mathtt{e}\left(m_k - G_k \cdot \left\lfloor \frac{m_k}{G_k} \right\rfloor, L_k \right) \\ &\qquad + \left\lceil \frac{m_k}{G_k} \right\rceil \cdot \mathtt{e}(0, L_k) \ . \end{aligned} \tag{A.4}$$

$\mathbb{E}(l_k)$ includes the headers, so that (5.10) can not be used directly to compute its time of transmission. Instead this time is given by $\mathbb{t}(\mathbb{E}(l_k))$, where

$$\mathbb{t}(l) = \frac{h^{\mathrm{phy}}}{\mathscr{R}^{\mathrm{b}}} + \frac{8 \cdot l}{\mathscr{R}^{\mathrm{d}}} + t^{\mathrm{p}} + SIFS + \frac{h^{\mathrm{phy}}}{\mathscr{R}^{\mathrm{b}}} + \frac{8 \cdot ACK}{\mathscr{R}^{\mathrm{d}}} + t^{\mathrm{p}}. \tag{A.5}$$

All the notations used above, retain their meanings defined earlier (cf. Section 4.5).

Best delay

It is defined as the time taken to transmit in the first attempt, all MPDUs related to the reference application message, m_k. Denoted by λ'_k, it is determined as

$$\frac{\lambda'_k}{n_k} = \mathbb{t}(\mathbb{E}(l_k))\,. \tag{A.6}$$

[1]Nowadays, IP packets are not fragmented by the MAC layer due to higher payload allowed by the latter. We keep our model general.

Average delay

It is defined as an average time taken to transmit all MPDUs related to the reference application message, m_k. Its derivation is based on the best case and the cases that arise after the first-order collision resolutions. We assume that the first-order collision resolution is done sequentially, with each involved MAC process being equally favored after the collision. Over longer duration, the system achieves behavior closer to this assumption.

Formally, the average delay, λ''_k, is computed as the weighted mean of times for the best delay and the first-order collision resolutions. For the latter events, we consider collisions of all possible sizes that are resolved equally in favor of each MAC process involved in them.

$$\frac{\lambda''_k}{n_k} = \mathtt{P}(\{k\}) \cdot \lambda'_k + \sum_{c=2}^{n} \sum_{s \in \mathtt{S}_k(c)} \mathtt{P}(s) \left[\mathtt{t'}_k(s) + \mathbb{E}(\mathtt{t''}_k(s))\right], \tag{A.7}$$

where $\mathtt{P}(\cdot)$ maps a set of MAC processes to probability of their transmission. The introduced terms are defined as follows.

- $\mathtt{P}(\{k\})$ is the probability of transmission by MAC process, k only, which is the best delay case;
- $\mathtt{S}_k(c)$ is a collection of collision sets of size c that contains the reference MAC process, k;
- $\mathtt{P}(s)$ is the probability of a collision by MAC processes in s;
- $\mathtt{t'}_k(s)$ is the duration of the collision state on the channel by MAC processes in s — it excludes the collision resolution time;
- $\mathbb{E}(\mathtt{t''}_k(s))$ is the expected time of first-order collision resolution — it excludes the duration of the collision state.

The event space considered for the average delay comprises a successful transmission and all possible collision cases. Therefore, $\mathtt{P}(\{k\})$ and $\mathtt{P}(s)$ are normalized within this space. These terms are derived as follows.

- $\mathtt{P}(\{k\})$ — this probability term is computed by deriving a mapping, $\mathtt{P}(z)$, of all MAC processes in z, to a probability of their simultaneous transmission. MAC processes not in z, $\mathcal{K} \setminus z$, are considered silent. In our system, we assume that application phasings are not known. In an unsaturated network with periodic applications, the probability of transmission by MAC process, k, is given by $\frac{1}{d_k/aSlotTime}$, where $aSlotTime$ is the PHY slot duration as determined by the wireless technology. d_k and $aSlotTime$ are expressed in same time units. Now

  $$\mathtt{P}(z) = \prod_{i \in z} \frac{1}{d_i/aSlotTime} \cdot \prod_{j \in \mathcal{K} \setminus z} \left(1 - \frac{1}{d_j/aSlotTime}\right).$$

  $\mathtt{P}(z)$ at $z = \{k\}$ gives $\mathtt{P}(\{k\})$.

- $\mathtt{S}_k(c)$ — an arbitrary subset of $\mathcal{K}$ with cardinality c, may or may not contain k. Hence, we make a partition of $\mathcal{K}$ into two subsets, $\mathcal{K}'_k = \mathcal{K} \setminus \{k\}$ and $\{k\}$. A union of an arbitrary subset of $\mathcal{K}'_k$ with $\{k\}$, yields a set which necessarily contains k. Now,

  $$\mathtt{S}_k(c) \triangleq \left\{ e \cup \{k\} \mid e \subseteq \mathcal{K}'_k, |e| = c - 1 \right\}.$$

Also,

$$|\mathtt{S}_k(c)| = \binom{n-1}{c-1}.$$

this probability term is computed by evaluating $\mathtt{P}(z)$ at $z = s$.

- $\mathtt{t}'_k(s)$ — the duration of the collision state on the channel by MAC processes in s depends on the size of the largest MPDU in s, hence

$$\mathtt{t}'_k(s) = tt\left(\mathop{\uparrow}_{i \in s} \mathbb{E}(l_i)\right).$$

- $\mathbb{E}(\mathtt{t}''_k(s))$ — this time term is computed by considering all possible orders of collision resolution that allow the MAC process, k, to transmit. Each order represents a transmission schedule of MAC processes, while all orders are assumed to occur equally likely. All processes ahead of the MAC process, k, incur delay to it. Given a collision set, s, suppose $\mathtt{O}_k(s)$ be a set of all orders: $|\mathtt{O}_k(s)| = |s|!$.

$$\mathbb{E}(\mathtt{t}''_k(s)) = \frac{1}{|s|!} \sum_{\langle o \rangle \in \mathtt{O}_k(s)} \sum_{i=1}^{\mathtt{pos}(\langle o \rangle, k)} (tt(\mathbb{E}(l_i)) + bk_i), \tag{A.8}$$

  where,

  - $\mathtt{pos}(\langle o \rangle, k) \in \{1, 2, \cdots, |s|\}$ maps to the position of k within the order $\langle o \rangle$;
  - $bk_i = aSlotTime \cdot \frac{CW_1}{2}$ is an average backoff time of MAC process i with contention window, CW_1, during the first-order collision resolution.

Worst delay

It is defined as the time taken to transmit last in the schedule during the first-order collision resolution involving all applications, all MPDUs related to the reference application message, m_k. Denoted by λ'''_k, it is determined as

$$\frac{\lambda'''_k}{n_k} = tt\left(\mathop{\uparrow}_{i \in \mathcal{K}} \mathbb{E}(l_i)\right) + \sum_{j \in \mathcal{K}} (tt(\mathbb{E}(l_j)) + bk_j). \tag{A.9}$$

The first term above is the duration of a size N collision.

A.2 Model validation and discussion

We validated our model by the requirements of an experimental autonomous navigation system [204] that communicates in a wireless unmanaged network composed of three nodes, all within a single broadcast domain: a robot that transmits navigation signals to a base station, and a data processing unit. This real-time distributed system hosts five periodic applications, with requirements stated in Table A.1.

For the transport layer, TCP is selected, whose implementation in Robot Operating System (ROS) mandates a TCP ACK for each segment, thus $seq^{\text{tp}} = 1$. Other specifications taken from ROS are the segment payload limit, $G = 1448$ bytes, TCP ACK size, $ACK^{\text{tp}} =$

Table A.1: Experimental setup: requirements of applications.

application	period, d (ms)	message length, m (bytes)
LaserData	79.0	136
FrameData	50.0	2736
OdomFrameGM	50.0	144
VelCmd	50.0	144
OdomData	37.0	1024

66 bytes, the MPDU payload limit, $L = 2304$ bytes and OSI headers sizes, $h^{\text{tp}} = 32$ bytes, $h^{\text{ip}} = 20$ bytes and $h^{\text{mac}} = 34$ bytes. Validation tests are conducted for three WLAN technologies [92], IEEE 802.11b, IEEE 802.11g and IEEE 802.11a tuned at $DIFS = SIFS + 2 \times aSlotTime$ and contention windows, $CW_0 = 32$ and $CW_1 = 64$. Other PHY/MAC parameter values are listed in Table A.2. For each WLAN technology, we report two results: the three types of delays for each application, and probabilities of three types of events: network being idle, cumulative success of all MAC processes and network collisions of various sizes. These probabilities are computed as follows. Let $\mathtt{P}^{\text{i}}$, $\mathtt{P}^{\text{s}}$ and $\mathtt{P}(c)$ denote these three probability terms with c being the collisions size, then

$$\mathtt{P}^{\text{i}} = \mathtt{P}(\varnothing)\,, \tag{A.10}$$

$$\mathtt{P}^{\text{s}} = \sum_{k \in \mathcal{K}} \mathtt{P}(\{k\}), \tag{A.11}$$

$$\begin{aligned} \mathtt{P}(c) &= \sum_{s \in \mathtt{S}(c)} \mathtt{P}(s), \\ \mathtt{S}(c) &\triangleq \{e \cup \{k\} \mid e \subset \mathcal{K}, |e| = c - 1\}\,, \\ |\mathtt{S}(c)| &= \binom{N-1}{c-1}. \end{aligned} \tag{A.12}$$

Table A.2: IEEE 802.11 b,g,a PHY/MAC parameters values.

parameter	b	g	a
$aSlotTime$ (µs)	20	9	9
$SIFS$ (µs)	10	10	16
Data rate (bits/µs)	11	54	54
ACK (bytes)	14	14	14
PHY header (bits)	192	144	40

The results are reported in Figure A.1. On the onset, it appears that the average delays for all WLAN technologies are larger than twice the best delays. This is due to the collisions time overhead that incurs an extra delay of at least the same order as the best delays.

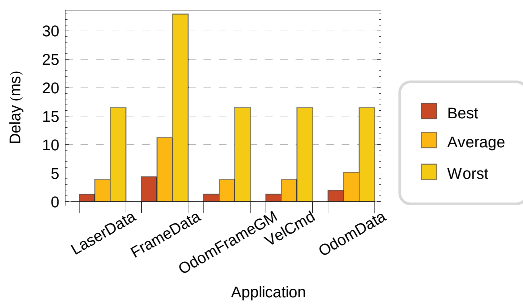
Another observation is the relatively large values of the worst delays. We underpin the effect of our mean-field approach on the two terms in (A.9). The first term strongly depends on the size of the largest MPDU involved in a given collision. The fact that MPDU sizes reflect only the mean payload generated at the transport layer, the first term has a low impact. The second term strongly depends on the number of sources. Our model reduces the transmitters of data segments and corresponding control segments, to one virtual source per application. Thus the length of the longest schedule during the first-order collisions resolution is effectively reduced by an order of mean MPDU count per application. Consequently, the second term has a low impact, as well. Given the conditions that lead to the worst case delay, their occurrence probability is almost negligible. Their occurrence represent system hyper periods, which is estimated to be 4.22888 days at the application and our model levels. Lastly, the lower delays of IEEE 802.11g and IEEE 802.11a are due to their higher transmission rates.

For the events probabilities, we present three observations. First, the probability of the wireless network being idle is quite high in an unsaturated MAC queues case, like the one we evaluated. Second, natural logarithms of collisions probabilities are almost negative exponentially distributed. This is due to an almost negligible probabilities of occurrence of large size collisions. In fact, collisions beyond size 3 have no impact on system performance. Third, the probabilities of collisions for IEEE 802.11b are higher than IEEE 802.11g and IEEE 802.11a. This owes to more slots per application period due to smaller $aSlotTime$ durations in IEEE 802.11g and IEEE 802.11a, so that probability mass resides more towards network idle event in them.

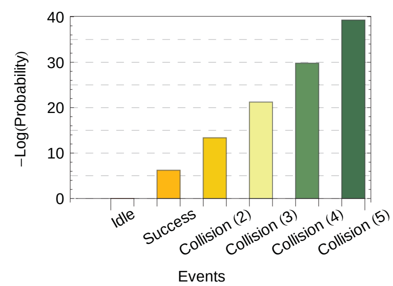
A.3 Next direction

Validation results from the current analysis are encouraging. Next, we intend to take the following steps:

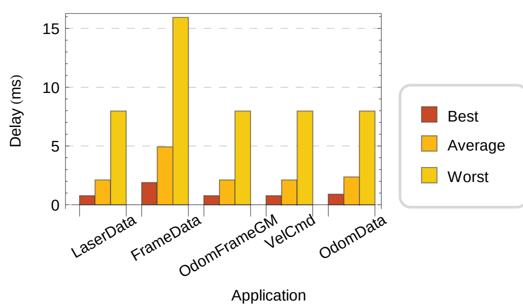
- Verification of our model by data obtained from the experimental autonomous navigation system [204];
- As aimed for this work, we seek a schedule for applications that is optimized on delay. To achieve this, application phasing will be used.



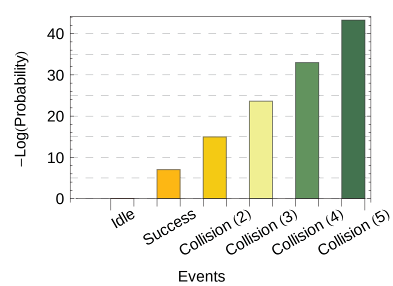
(a) IEEE 802.11b – delays.



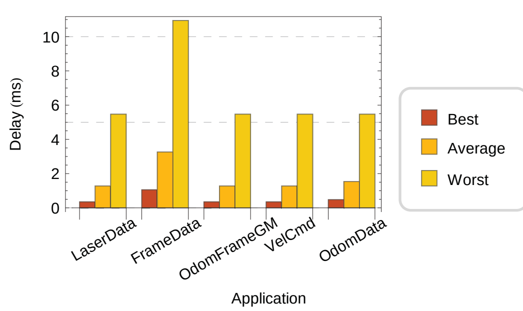
(b) IEEE 802.11b – events probabilities.



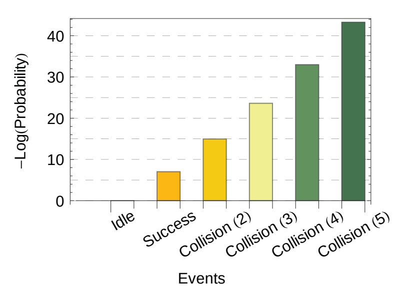
(c) IEEE 802.11g – delays.



(d) IEEE 802.11g – events probabilities.



(e) IEEE 802.11a – delays.



(f) IEEE 802.11a – events probabilities.

Figure A.1: Unmanaged wireless network delays and probabilities predictions for autonomous navigation system, evaluated for three WLAN technologies. Each probability chart plots $\mathbb{P}^{\mathbf{i}}$ in (A.10), $\mathbb{P}^{\mathbf{s}}$ in (A.11) and $\mathbb{P}(c)$ in (A.12) for $2 \leqslant c \leqslant 5$. The probability values are zoomed in by taking their natural logarithms.

Symbol Index

state	general MAC state, shown as bold	9
$AIFS$	arbitration interframe space	14
$CWmin$	minimum contention window size	14
$CWmax$	maximum contention window size	14
$RetryLimit$	number of retries allowed	14
$TXOPLimit$	number of MPDUs allowed per transmission opportunity	14
$aSlotTime$	duration of the PHY slot	15
$AIFSN$	arbitration interframe space number	15
$SIFS$	short interframe space	15
bc	backoff counter	15
$\mathbb{N}_0$	set of natural numbers, i.e. $\{0, 1, 2, \cdots\}$	15
$\mathbb{N}_1$	set of positive natural numbers, i.e. $\{1, 2, \cdots\}$	15
$\mathbb{Z}$	set of integers, i.e. $\{\cdots, -2, -1, 0, 1, 2, \cdots\}$	15
CW_r	contention window at retry attempt, r	15
$\texttt{cw}(r)$	function, mapping to the contention window at retry attempt, r	15
NoP	*network of processes*	16
AC	Access Category, for MAC queues	16
P-tuple	parameter settings, per MAC process	22
$\mathbf{n}$	vector of the number of processes per QoS class	22
$\mathcal{Q}$	set of indices for QoS classes	22
$\mathcal{K}$	set of indices for MAC processes	22
$\underset{\text{mac}}{\mathcal{S}}$	set of backoff process states	23
$\underset{\text{phy}}{\mathcal{S}}$	set of PHY states	23
$\underset{\text{mac}}{\mathcal{O}}^{(i)}$	i^{th} MAC system cycle	23
$\underset{\text{phy}}{\mathcal{O}}^{(i)}$	i^{th} PHY system cycle	24
ν	expected channel access frequency	27

$\mathcal{X}^x$	set of $TXOP$ settings for the $TXOP$ driven performance	69
$\mathcal{W}^x$	set of CW settings for the CW driven performance	69
$\mathcal{W}^d$	set of CW settings for the default EDCA parameter set arbitration	70
$\mathcal{R}^d$	set of *RetryLimit* settings for the default EDCA parameter set arbitration	70
$\mathcal{N}^d$	set of the number of processes per class for the default EDCA parameter set arbitration	70
$\texttt{time}\,(s)$	time polynomial of the system state space size, s	76
$\mathrm{z}_i(\mathbf{p})$	function, mapping an ncp, $\mathbf{p}$, to a quality property, i	97
$\mathcal{Z}$	objective space	97
$\mathcal{Z}'(\cdot)$	solution region in the objective space for the given QoS requirements of applications	97
$\mathcal{PF}$	Pareto front	97
up	utopia point	98
$\hat{\mathcal{N}}$	decision vector space for our experiments	101
$\lvert\prec(V,\mathcal{Q})\rvert$	function, mapping a set of selected parameter values, $\mathcal{V}$, and the number of classes, $\lvert\mathcal{Q}\rvert$, to the size of the strict total order	101
$\mathcal{U}(\mathbf{x})$	function, mapping a transmission opportunity vector, $\mathbf{x}$, to channel utilization	109
$\mathcal{K}^{\mathrm{el}}$	index set of elastic applications	120
$\mathcal{K}^{\mathrm{iel}}$	index set of inelastic applications	120
λ'_k	best delay of the MAC process, k, in an unsaturated network	141
λ''_k	average delay of the MAC process, k, in an unsaturated network	142
λ'''_k	worst delay of the MAC process, k, in an unsaturated network	143

Bibliography

[1] I. Aad and C. Castelluccia. 'Differentiation mechanisms for IEEE 802.11'. In: *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Vol. 1. 2001, 209–218 vol.1. DOI: 10.1109/INFCOM.2001.916703.

[2] L. Abeni and G. Buttazzo. 'Adaptive bandwidth reservation for multimedia computing'. In: *Real-Time Computing Systems and Applications, 1999. RTCSA '99. Sixth International Conference on*. 1999, pp. 70–77. DOI: 10.1109/RTCSA.1999.811195.

[3] O.B. Akan and I.F. Akyildiz. 'ATL: an adaptive transport layer suite for next-generation wireless Internet'. In: *Selected Areas in Communications, IEEE Journal on* 22.5 (2004), pp. 802–817. ISSN: 0733-8716. DOI: 10.1109/JSAC.2004.826919.

[4] W. Aslam, J. Lukkien and I. Radovanovic. 'Allocation of in-home network resources to competing multimedia streams using EDCA'. In: *Consumer Electronics, 2009. ICCE '09. Digest of Technical Papers International Conference on*. Jan. 2009, pp. 1–2. DOI: 10.1109/ICCE.2009.5012376.

[5] W. Aslam and J.J. Lukkien. 'Streaming-aware channel utilization improvement for wireless home networks'. In: *Consumer Electronics, IEEE Transactions on* 58.1 (2012), pp. 69–77. ISSN: 0098-3063. DOI: 10.1109/TCE.2012.6170057.

[6] Waqar Aslam and Johan J. Lukkien. 'Delay Constrained Resource Allocation for Wireless Home Networks'. In: *2012 IEEE International Conference on Consumer Electronics (ICCE) (ICCE 2012)*. Las Vegas, USA, Jan. 2012, pp. 98–99.

[7] Waqar Aslam and Johan J. Lukkien. 'Scavenging the Channel Share: Performance on Sensing and Allocation'. In: *Proceedings of the first Taibah University International Conference on Computing and Information Technology (ICCIT 2012)*. Al-Madinah Al-Munawwarah, Saudi Arabia, Dec. 2012, pp. 834–837.

[8] A. Autenrieth and A. Kirstadter. 'Engineering end-to-end IP resilience using resilience-differentiated QoS'. In: *Communications Magazine, IEEE* 40.1 (2002), pp. 50–57. ISSN: 0163-6804. DOI: 10.1109/35.978049.

[9] Yan Bai and M.R. Ito. 'QoS control for video and audio communication in conventional and active networks: Approaches and comparison'. In: *Communications Surveys Tutorials, IEEE* 6.1 (2004), pp. 42–49. ISSN: 1553-877X. DOI: 10.1109/COMST.2004.5342233.

[10] S.D. Baker and D.H. Hoglund. 'Medical-Grade, Mission-Critical Wireless Networks [Designing an Enterprise Mobility Solution in the Healthcare Environment]'. In: *Engineering in Medicine and Biology Magazine, IEEE* 27.2 (Mar. 2008), pp. 86–95. ISSN: 0739-5175. DOI: 10.1109/EMB.2008.915498.

[11] Hari Balakrishnan, Srinivasan Seshan and RandyH. Katz. 'Improving reliable transport and handoff performance in cellular wireless networks'. English. In: *Wireless Networks* 1.4 (1995), pp. 469–481. ISSN: 1022-0038. DOI: 10.1007/BF01985757. URL: http://dx.doi.org/10.1007/BF01985757.

[12] A. Banchs, P. Serrano and L. Vollero. 'Providing Service Guarantees in 802.11e EDCA WLANs with Legacy Stations'. In: *Mobile Computing, IEEE Transactions on* 9.8 (2010), pp. 1057–1071. ISSN: 1536-1233. DOI: 10.1109/TMC.2010.52.

[13] Albert Banchs and Pablo Serrano. 'Revisiting 802.11e EDCA performance analysis'. In: *Wireless Personal Communications* 43.4 (2007), pp. 1145–1149. ISSN: 0929-6212. DOI: 10.1007/s11277-007-9290-3. URL: http://dx.doi.org/10.1007/s11277-007-9290-3.

[14] Jakob Bernoulli. 'Ars Conjectandi: Usum & Applicationem Praecedentis Doctrinae in Civilibus, Moralibus & Oeconomicis'. In: 1713. Chap. 4.

[15] K. Bhargavan et al. 'Verisim: formal analysis of network simulations'. In: *Software Engineering, IEEE Transactions on* 28.2 (2002), pp. 129–145. ISSN: 0098-5589. DOI: 10.1109/32.988495.

[16] V. Bharghavan et al. 'The TIMELY adaptive resource management architecture'. In: *Personal Communications, IEEE* 5.4 (1998), pp. 20–31. ISSN: 1070-9916. DOI: 10.1109/98.709366.

[17] G. Bianchi. 'Performance analysis of the IEEE 802.11 distributed coordination function'. In: *Selected Areas in Communications, IEEE Journal on* 18.3 (Mar. 2000), pp. 535–547. ISSN: 0733-8716. DOI: 10.1109/49.840210.

[18] G. Bianchi and I. Tinnirello. 'Remarks on IEEE 802.11 DCF performance analysis'. In: *Communications Letters, IEEE* 9.8 (2005), pp. 765–767. ISSN: 1089-7798. DOI: 10.1109/LCOMM.2005.1496609.

[19] G. Bianchi, I. Tinnirello and L. Scalia. 'Understanding 802.11e contention-based prioritization mechanisms and their coexistence with legacy 802.11 stations'. In: *Network, IEEE* 19.4 (July 2005), pp. 28–34. ISSN: 0890-8044. DOI: 10.1109/MNET.2005.1470680.

[20] Benny Bing. *Emerging Technologies in Wireless LANs*. Cambridge Books Online. Cambridge University Press, 2007.

[21] F. Birlik, O. Gurbuz and O. Ercetin. 'IPTVhome networking via 802.11 wireless mesh networks: an implementation experience'. In: *Consumer Electronics, IEEE Transactions on* 55.3 (Aug. 2009), pp. 1192–1199. ISSN: 0098-3063. DOI: 10.1109/TCE.2009.5277975.

[22] S. Blake et al. 'An Architecture for Differentiated Services'. In: *RFC 2475, IETF* (Dec. 1998). URL: http://tools.ietf.org/html/rfc2475.

[23] G.V. Bochmann, R. Dssouli and J. R. Zhao. 'Trace analysis for conformance and arbitration testing'. In: *Software Engineering, IEEE Transactions on* 15.11 (1989), pp. 1347–1356. ISSN: 0098-5589. DOI: 10.1109/32.41328.

[24] J. Bolliger and T. Gross. 'A framework based approach to the development of network aware applications'. In: *Software Engineering, IEEE Transactions on* 24.5 (1998), pp. 376–390. ISSN: 0098-5589. DOI: 10.1109/32.685260.

[25] L. Bononi, M. Conti and E. Gregori. 'Runtime Optimization of IEEE 802.11 Wireless LANs Performance'. In: *IEEE Transactions on Parallel and Distributed Systems* 15.1 (Jan. 2004), pp. 66–80. ISSN: 1045-9219. DOI: 10.1109/TPDS.2004.1264787.

[26] A. Bouch, M.-A. Sasse and H. DeMeer. ‘Of packets and people: a user-centered approach to quality of service’. In: *Quality of Service, 2000. IWQOS. 2000 Eighth International Workshop on*. 2000, pp. 189–197. DOI: 10.1109/IWQOS.2000.847955.

[27] R. Braden, D. Clark and S. Shenker. ‘Integrated Services in the Internet Architecture: an Overview’. In: *RFC 1633, IETF* (June 1994). URL: http://tools.ietf.org/html/rfc1633.

[28] Martin Burkhart and Xenofontas Dimitropoulos. ‘Privacy-preserving distributed network troubleshooting – bridging the gap between theory and practice’. In: *ACM Trans. Inf. Syst. Secur.* 14.4 (Dec. 2008), 31:1–31:30. ISSN: 1094-9224. DOI: 10.1145/2043628.2043632. URL: http://doi.acm.org/10.1145/2043628.2043632.

[29] J.N. Buxton and B. Randell, eds. *Software Engineering Techniques: Report on a Conference sponsored by the NATO Science Committee*. Rome, Italy: Brussels, Scientific Affairs Division, NATO (1970), 27th to 31st October 1969, p. 164.

[30] L.X. Cai et al. ‘Voice capacity analysis of WLAN with unbalanced traffic’. In: *Vehicular Technology, IEEE Transactions on* 55.3 (2006), pp. 752–761. ISSN: 0018-9545. DOI: 10.1109/TVT.2006.874145.

[31] C. Caini, R. Firrincieli and D. Lacamera. ‘The TCP “Adaptive-Selection” Concept’. In: *Systems Journal, IEEE* 2.1 (2008), pp. 83–89. ISSN: 1932-8184. DOI: 10.1109/JSYST.2007.914744.

[32] F. Cali, M. Conti and E. Gregori. ‘IEEE 802.11 wireless LAN: capacity analysis and protocol enhancement’. In: *INFOCOM ’98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Vol. 1. 1998, 142–149 vol.1. DOI: 10.1109/INFCOM.1998.659648.

[33] F. Cali, M. Conti and Enrico Gregori. ‘Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit’. In: *Networking, IEEE/ACM Transactions on* 8.6 (2000), pp. 785–799. ISSN: 1063-6692. DOI: 10.1109/90.893874.

[34] F. Cali, M. Conti and Enrico Gregori. ‘IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism’. In: *Selected Areas in Communications, IEEE Journal on* 18.9 (2000), pp. 1774–1786. ISSN: 0733-8716. DOI: 10.1109/49.872963.

[35] E. Camponogara et al. ‘Distributed model predictive control’. In: *Control Systems, IEEE* 22.1 (2002), pp. 44–52. ISSN: 1066-033X. DOI: 10.1109/37.980246.

[36] G. Cena et al. ‘Evaluation of Response Times in Industrial WLANs’. In: *Industrial Informatics, IEEE Transactions on* 3.3 (Aug. 2007), pp. 191–201. ISSN: 1551-3203. DOI: 10.1109/TII.2007.903219.

[37] G. Cena et al. ‘Industrial applications of IEEE 802.11e WLANs’. In: *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*. May 2008, pp. 129–138. DOI: 10.1109/WFCS.2008.4638739.

[38] G. Cena et al. ‘On the Performance of IEEE 802.11e Wireless Infrastructures for Soft-Real-Time Industrial Applications’. In: *Industrial Informatics, IEEE Transactions on* 6.3 (Aug. 2010), pp. 425–437. ISSN: 1551-3203. DOI: 10.1109/TII.2010.2052058.

[39] P.A. Chaparro et al. ‘Supporting Scalable Video Transmission in MANETs through Distributed Admission Control Mechanisms’. In: *Parallel, Distributed and Network-Based Processing, 2010 18th Euromicro International Conference on*. Feb. 2010, pp. 238–245. DOI: 10.1109/PDP.2010.44.

[40] E. Charfi, L. Chaari and L. Kamoun. ‘PHY/MAC Enhancements and QoS Mechanisms for Very High Throughput WLANs: A Survey’. In: *Communications Surveys Tutorials, IEEE* 15.4 (2013), pp. 1714–1735. ISSN: 1553-877X. DOI: 10.1109/SURV.2013.013013.00084.

[41] P. Chatzimisios, A.C. Boucouvalas and V. Vitsas. ‘IEEE 802.11 Packet Delay - A Finite Retry Limit Analysis’. In: *Global Telecommunications Conference, 2003. GLOBECOM ’03. IEEE*. Vol. 2. 2003, 950–954 Vol.2. DOI: 10.1109/GLOCOM.2003.1258379.

[42] P. Chatzimisios, A.C. Boucouvalas and V. Vitsas. ‘Influence of channel BER on IEEE 802.11 DCF’. In: *Electronics Letters* 39.23 (2003), pages. ISSN: 0013-5194. DOI: 10.1049/el:20031081.

[43] Xiaoping Che and S. Maag. ‘Passive Testing on Performance Requirements of Network Protocols’. In: *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*. 2013, pp. 1439–1444. DOI: 10.1109/WAINA.2013.12.

[44] Jyh-Cheng Chen and Kai-wen Cheng. ‘EDCA/CA: Enhancement of IEEE 802.11e EDCA by Contention Adaption for Energy Efficiency’. In: *Wireless Communications, IEEE Transactions on* 7.8 (2008), pp. 2866–2870. ISSN: 1536-1276. DOI: 10.1109/TWC.2008.070168.

[45] Rui Chen, Gergely Acs and Claude Castelluccia. ‘Differentially private sequential data publication via variable-length n-grams’. In: *Proceedings of the 2012 ACM conference on Computer and communications security*. CCS ’12. Raleigh, North Carolina, USA: ACM, 2012, pp. 638–649. ISBN: 978-1-4503-1651-4. DOI: 10.1145/2382196.2382263.

[46] Ruichuan Chen et al. ‘Towards statistical queries over distributed private user data’. In: *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. NSDI’12. San Jose, CA: USENIX Association, 2012, pp. 13–13.

[47] X. Chen et al. ‘Supporting QoS in IEEE 802.11e wireless LANs’. In: *Wireless Communications, IEEE Transactions on* 5.8 (2006), pp. 2217–2227. ISSN: 1536-1276. DOI: 10.1109/TWC.2006.1687738.

[48] Zhenzhong Chen, Maodong Li and Yap-Peng Tan. ‘Perception-Aware Multiple Scalable Video Streaming Over WLANs’. In: *Signal Processing Letters, IEEE* 17.7 (July 2010), pp. 675–678. ISSN: 1070-9908. DOI: 10.1109/LSP.2010.2046193.

[49] Yu Cheng et al. ‘A cross-layer approach for WLAN voice capacity planning’. In: *Selected Areas in Communications, IEEE Journal on* 25.4 (2007), pp. 678–688. ISSN: 0733-8716. DOI: 10.1109/JSAC.2007.070505.

[50] Jeong-woo Cho, J.-Y. Le Boudec and Yuming Jiang. ‘On the Asymptotic Validity of the Decoupling Assumption for Analyzing 802.11 MAC Protocol’. In: *Information Theory, IEEE Transactions on* 58.11 (2012), pp. 6879–6893. ISSN: 0018-9448. DOI: 10.1109/TIT.2012.2208582.

[51] Jaehyuk Choi et al. ‘Adaptive Optimization of Rate Adaptation Algorithms in Multi-Rate WLANs’. In: *Network Protocols, 2007. ICNP 2007. IEEE International Conference on*. 2007, pp. 144–153. DOI: 10.1109/ICNP.2007.4375845.

[52] Jaehyuk Choi et al. ‘EBA: an enhancement of the IEEE 802.11 DCF via distributed reservation’. In: *Mobile Computing, IEEE Transactions on* 4.4 (2005), pp. 378–390. ISSN: 1536-1233. DOI: 10.1109/TMC.2005.57.

[53] Cisco. 'Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2017'. In: [Online; accessed 18-December-2013], Americas Headquarters Cisco Systems Inc., 170 West Tasman Drive San Jose CA 95134-1706, USA: Cisco.

[54] Cisco. 'Enterprise Mobility 7.3 Design Guide'. In: [Online; accessed 9-May-2014]. Americas Headquarters Cisco Systems Inc., 170 West Tasman Drive San Jose CA 95134-1706, USA: Cisco, Sept. 2013. Chap. Cisco Unified Wireless QoS.

[55] C. Cordeiro. 'The pursuit of tens of gigabits per second wireless systems [Industry Perspectives]'. In: *Wireless Communications, IEEE* 20.1 (2013), pp. 3–5. ISSN: 1536-1284. DOI: 10.1109/MWC.2013.6472190.

[56] Scott E. Coull et al. 'Playing Devil's Advocate: Inferring Sensitive Information from Anonymized Network Traces'. In: *in Proceedings of the Network and Distributed System Security Symposium*. 2007, pp. 35–47.

[57] Scott E. Coull et al. 'The Challenges of Effectively Anonymizing Network Data'. In: *Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security*. CATCH '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 230–236. ISBN: 978-0-7695-3568-5. DOI: 10.1109/CATCH.2009.27.

[58] F. De Rango, P. Fazio and S. Marano. 'Mobility Prediction and Resources Reservation in WLAN Networks Under a 2D Mobility Model'. In: *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*. 2006, pp. 1–5. DOI: 10.1109/VTCF.2006.551.

[59] 'Definitions of terms related to quality of service'. In: *Recommendation ITU-T E.800* (Sept. 2008). [Online; accessed 29-October-2013].

[60] David F. Delchamps. 'Stabilizing a linear system with quantized state feedback'. In: *Automatic Control, IEEE Transactions on* 35.8 (1990), pp. 916–924. ISSN: 0018-9286. DOI: 10.1109/9.58500.

[61] Der Jiunn Deng and Hsu Chun Yen. 'Quality-of-service Provisioning System for Multimedia Transmission in IEEE 802.11 Wireless LANs'. In: *IEEE Journal on Selected Areas in Communications* 23.6 (June 2005), pp. 1240–1252. ISSN: 0733-8716. DOI: 10.1109/JSAC.2005.845632.

[62] H. Dong et al. 'Selective bit-error checking at the MAC layer for voice over mobile ad hoc networks with IEEE 802.11'. In: *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*. Vol. 2. 2004, 1240–1245 Vol.2. DOI: 10.1109/WCNC.2004.1311366.

[63] X.J. Dong and P. Varaiya. 'Saturation throughput analysis of IEEE 802.11 wireless LANs for a lossy channel'. In: *Communications Letters, IEEE* 9.2 (2005), pp. 100–102. ISSN: 1089-7798. DOI: 10.1109/LCOMM.2005.02011.

[64] C. Dovrolis, D. Stiliadis and P. Ramanathan. 'Proportional differentiated services: delay differentiation and packet scheduling'. In: *Networking, IEEE/ACM Transactions on* 10.1 (2002), pp. 12–26. ISSN: 1063-6692. DOI: 10.1109/90.986503.

[65] Z. Duan et al. 'A core stateless bandwidth broker architecture for scalable support of guaranteed services'. In: *Parallel and Distributed Systems, IEEE Transactions on* 15.2 (2004), pp. 167–182. ISSN: 1045-9219. DOI: 10.1109/TPDS.2004.1264799.

[66] DV02. *How to choose a Wireless Lan solution*. http://dv02.co.uk/Subjects/wireless-lan/. [Online; accessed 23-October-2013]. 2008.

[67] Mustafa Ergen and Pravin Varaiya. 'Throughput Analysis and Admission Control for IEEE 802.11a'. English. In: *Mobile Networks and Applications* 10.5 (2005), pp. 705–716. ISSN: 1383-469X. DOI: 10.1007/s11036-005-3364-9. URL: http://dx.doi.org/10.1007/s11036-005-3364-9.

[68] Behdis Eslamnour, S. Jagannathan and MaciejJ. Zawodniok. 'Dynamic Channel Allocation in Wireless Networks Using Adaptive Learning Automata'. In: *International Journal of Wireless Information Networks* 18.4 (2011), pp. 295–308. ISSN: 1068-9605. DOI: 10.1007/s10776-011-0146-0.

[69] Linghang Fan et al. 'A Cross-Layer Delay Differentiation Packet Scheduling Scheme for Multimedia Content Delivery in 3G Satellite Multimedia Systems'. In: *Broadcasting, IEEE Transactions on* 54.4 (Dec. 2008), pp. 806–815. ISSN: 0018-9316. DOI: 10.1109/TBC.2008.2005397.

[70] J. Francois et al. 'PTF: Passive Temporal Fingerprinting'. In: *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*. 2011, pp. 289–296. DOI: 10.1109/INM.2011.5990703.

[71] H. Fu and E.W. Knightly. 'A simple model of real-time flow aggregation'. In: *Networking, IEEE/ACM Transactions on* 11.3 (2003), pp. 422–435. ISSN: 1063-6692. DOI: 10.1109/TNET.2003.813045.

[72] M. Garetto, T. Salonidis and E.W. Knightly. 'Modeling Per-Flow Throughput and Capturing Starvation in CSMA Multi-Hop Wireless Networks'. In: *Networking, IEEE/ACM Transactions on* 16.4 (Aug. 2008), pp. 864–877. ISSN: 1063-6692. DOI: 10.1109/TNET.2007.902687.

[73] Shantanu Gattani and Thomas E. Daniels. 'Reference models for network data anonymization'. In: *Proceedings of the 1st ACM workshop on Network data anonymization*. NDA '08. Alexandria, Virginia, USA: ACM, 2008, pp. 41–48. ISBN: 978-1-60558-301-3. DOI: 10.1145/1456441.1456454. URL: http://doi.acm.org/10.1145/1456441.1456454.

[74] V. Gazis et al. 'Self-Management Capacities in Future Internet Wireless Systems'. In: *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns. Computation World:* 2009, pp. 9–14.

[75] M.A. El-Gendy, A. Bose and K.G. Shin. 'Evolution of the Internet QoS and Support for Soft Real-Time Applications'. In: *Proceedings of the IEEE* 91.7 (2003), pp. 1086–1104. ISSN: 0018-9219.

[76] Daqing Gu and Jinyun Zhang. 'QoS enhancement in IEEE 802.11 wireless local area networks'. In: *Communications Magazine, IEEE* 41.6 (2003), pp. 120–124. ISSN: 0163-6804. DOI: 10.1109/MCOM.2003.1204758.

[77] Hai-Bo Guo and Geng-Sheng Kuo. 'A dynamic and adaptive bandwidth management scheme for QoS support in wireless multimedia networks'. In: *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*. Vol. 3. 2005, 2081–2085 Vol. 3. DOI: 10.1109/VETECS.2005.1543690.

[78] 'H.264: Advanced video coding for generic audiovisual services'. In: *Recommendation ITU-T H-Series* (Mar. 2009). [Online; accessed 29-October-2013].

[79] Chong Han et al. 'Analytical Study of the IEEE 802.11p MAC Sublayer in Vehicular Networks'. In: *Intelligent Transportation Systems, IEEE Transactions on* 13.2 (2012), pp. 873–886. ISSN: 1524-9050. DOI: 10.1109/TITS.2012.2183366.

[80] Wenbo He, Hoang Nguyen and K. Nahrstedt. 'Experimental Validation of Middleware-based QoS Control in 802.11 Wireless Networks'. In: *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*. Oct. 2006, pp. 1–9. DOI: 10.1109/BROADNETS.2006.4374407.

[81] M. Heusse et al. 'Performance anomaly of 802.11b'. In: *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. Vol. 2. 2003, 836–843 vol.2. DOI: 10.1109/INFCOM.2003.1208921.

[82] C. T E R Hewage et al. 'Prioritized 3D video distribution over IEEE 802.11e'. In: *Future Network and Mobile Summit, 2010*. June 2010, pp. 1–9.

[83] Wei-jen Hsu and A. Helmy. 'On Nodal Encounter Patterns in Wireless LAN Traces'. In: *Mobile Computing, IEEE Transactions on* 9.11 (2010), pp. 1563–1577. ISSN: 1536-1233. DOI: 10.1109/TMC.2010.123.

[84] Chunyu Hu et al. 'A Distributed Approach of Proportional Bandwidth Allocation for Real-Time Services in UltraWideBand (UWB) WPANs'. In: *Parallel and Distributed Systems, IEEE Transactions on* 21.11 (2010), pp. 1626–1643. ISSN: 1045-9219. DOI: 10.1109/TPDS.2010.32.

[85] Chunyu Hu et al. 'A Distributed Approach of Proportional Bandwidth Allocation for Real-Time Services in UltraWideBand (UWB) WPANs'. In: *Parallel and Distributed Systems, IEEE Transactions on* 21.11 (Nov. 2010), pp. 1626–1643. ISSN: 1045-9219. DOI: 10.1109/TPDS.2010.32.

[86] Qing-yan Huang et al. 'A Multirate Adaptive Protocol for IEEE 802.11'. In: *Communication Technology, 2006. ICCT '06. International Conference on*. 2006, pp. 1–4. DOI: 10.1109/ICCT.2006.342001.

[87] Jie Hui and M. Devetsikiotis. 'A unified model for the performance analysis of IEEE 802.11e EDCA'. In: *Communications, IEEE Transactions on* 53.9 (2005), pp. 1498–1510. ISSN: 0090-6778. DOI: 10.1109/TCOMM.2005.855013.

[88] O. Hundt et al. 'Methods to Improve the Efficiency of Wireless LAN for Multimedia Home Networks'. In: *Consumer Electronics, IEEE Transactions on* 53.2 (May 2007), pp. 397–404. ISSN: 0098-3063. DOI: 10.1109/TCE.2007.381707.

[89] G-H Hwang and Dong-H Cho. 'Performance Analysis on Coexistence of EDCA and Legacy DCF Stations in IEEE 802.11 Wireless LANs'. In: *Wireless Communications, IEEE Transactions on* 5.12 (2006), pp. 3355–3359. ISSN: 1536-1276. DOI: 10.1109/TWC.2006.256955.

[90] 'IEEE Approved Draft Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: TV White Spaces Operation'. In: *IEEE P802.11af/D6.0, October 2013 (Amendment to IEEE Std 802.11-2012)* (2013), pp. 1–388.

[91] 'IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications'. In: *ANSI/IEEE Std 802.11, 1999 Edition (R2003)* (), pp. i–513. DOI: 10.1109/IEEESTD.2003.95617.

[92] 'IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications'. In: *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)* (Dec. 2007), pp. C1–1184. DOI: 10.1109/IEEESTD.2007.373646.

[93] A. Iera and A. Molinaro. 'Designing the interworking of terrestrial and satellite IP-based networks'. In: *Communications Magazine, IEEE* 40.2 (2002), pp. 136–144. ISSN: 0163-6804. DOI: 10.1109/35.983920.

[94] A. Iera et al. 'Improving QoS and throughput in single- and multihop WLANs through dynamic traffic prioritization'. In: *Network, IEEE* 19.4 (July 2005), pp. 35–44. ISSN: 0890-8044. DOI: 10.1109/MNET.2005.1470681.

[95] K. Ijiri, S. Ohzahata and K. Kawashima. 'TCP window control for QoS improvement based on channel occupancy information over wireless LAN'. In: *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*. July 2010, pp. 1016–1021. DOI: 10.1109/INDIN.2010.5549601.

[96] I. Inan, F. Keceli and E. Ayanoglu. 'Analysis of the 802.11e enhanced distributed channel access function'. In: *Communications, IEEE Transactions on* 57.6 (2009), pp. 1753–1764. ISSN: 0090-6778. DOI: 10.1109/TCOMM.2009.06.0701132.

[97] I. Inan, F. Keceli and E. Ayanoglu. 'Modeling the 802.11e Enhanced Distributed Channel Access Function'. In: *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*. 2007, pp. 2546–2551. DOI: 10.1109/GLOCOM.2007.484.

[98] ISO/IEC 7498-1:1994. *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*. 2006. Geneva, Switzerland: ISO, Geneva, Switzerland, June 1994. URL: http://www.iso.org.

[99] ISPreview. *Cisco Forecasts Quadruple Jump in Global Internet Traffic by 2014*. http://www.ispreview.co.uk/. [Online; accessed 5-November-2013].

[100] B. Issac, K.A. Hamid and C.E. Tan. 'Pre-Authenticated and Adaptive IP Mobility for WLAN nodes'. In: *Distributed Frameworks for Multimedia Applications, 2006. The 2nd International Conference on*. 2006, pp. 1–8. DOI: 10.1109/DFMA.2006.296900.

[101] K. Jaffres-Runser, Jean-Marie Gorce and S. Ubeda. 'QoS constrained wireless LAN optimization within a multiobjective framework'. In: *Wireless Communications, IEEE* 13.6 (Dec. 2006), pp. 26–33. ISSN: 1536-1284. DOI: 10.1109/MWC.2006.275195.

[102] R. Jain. 'Myths About Congestion Management in High-Speed Networks'. In: *Internetworking: Research and Experience* 3 (1992), pp. 101–113.

[103] Rajendra K. Jain, Dah-Ming W. Chiu and William R. Hawe. *A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems*. Tech. rep. Digital Equipment Corporation, 26th Sept. 1984. arXiv: cs.NI/9809099. URL: http://arxiv.org/abs/cs.NI/9809099.

[104] S. Jaiswal et al. 'Inferring TCP connection characteristics through passive measurements'. In: *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*. Vol. 3. 2004, 1582–1592 vol.3. DOI: 10.1109/INFCOM.2004.1354571.

[105] Hai Jiang, Ping Wang and Weihua Zhuang. 'A Distributed Channel Access Scheme with Guaranteed Priority and Enhanced Fairness'. In: *Wireless Communications, IEEE Transactions on* 6.6 (2007), pp. 2114–2125. ISSN: 1536-1276. DOI: 10.1109/TWC.2007.05746.

[106] T. Joshi et al. 'Airtime Fairness for IEEE 802.11 Multirate Networks'. In: *Mobile Computing, IEEE Transactions on* 7.4 (2008), pp. 513–527. ISSN: 1536-1233. DOI: 10.1109/TMC.2007.70740.

[107] Kyungtae Kang and Dong Kun Noh. 'Performance Assessment of Wireless ECG Transmission over IEEE 802.11 WLANs'. In: *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*. July 2011, pp. 1–5. DOI: 10.1109/ICCCN.2011.6005891.

[108] Kyungtae Kang et al. 'A Medical-Grade Wireless Architecture for Remote Electrocardiography'. In: *Information Technology in Biomedicine, IEEE Transactions on* 15.2 (Mar. 2011), pp. 260–267. ISSN: 1089-7771. DOI: 10.1109/TITB.2011.2104365.

[109] Kyungtae Kang et al. 'Design and QoS of a Wireless System for Real-Time Remote Electrocardiography'. In: *Biomedical and Health Informatics, IEEE Journal of* 17.3 (May 2013), pp. 745–755. ISSN: 2168-2194. DOI: 10.1109/JBHI.2013.2237782.

[110] P. Kemper and C. Tepper. 'Automated Trace Analysis of Discrete-Event System Models'. In: *Software Engineering, IEEE Transactions on* 35.2 (2009), pp. 195–208. ISSN: 0098-5589. DOI: 10.1109/TSE.2008.75.

[111] N. Khademi, Michael Welzl and S. Gjessing. 'Experimental evaluation of TCP performance in multi-rate 802.11 WLANs'. In: *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*. June 2012, pp. 1–9. DOI: 10.1109/WoWMoM.2012.6263696.

[112] Hyogon Kim et al. 'Resolving 802.11 performance anomalies through QoS differentiation'. In: *Communications Letters, IEEE* 9.7 (2005), pp. 655–657. ISSN: 1089-7798. DOI: 10.1109/LCOMM.2005.1461695.

[113] M. Kishore and Yao Liang. 'An empirical study on renegotiated CBR for VBR video services based on network testbed'. In: *Broadcasting, IEEE Transactions on* 52.3 (2006), pp. 362–367. ISSN: 0018-9316. DOI: 10.1109/TBC.2006.879857.

[114] Cristian Koliver et al. 'Specification, Mapping and Control for QoS Adaptation'. In: *Real-Time Syst.* 23.1/2 (July 2002), pp. 143–174. ISSN: 0922-6443. DOI: 10.1023/A:1015302604246. URL: http://dx.doi.org/10.1023/A:1015302604246.

[115] Zhen-Ning Kong et al. 'Performance analysis of IEEE 802.11e contention-based channel access'. In: *Selected Areas in Communications, IEEE Journal on* 22.10 (2004), pp. 2095–2106. ISSN: 0733-8716. DOI: 10.1109/JSAC.2004.836019.

[116] Ioannis Koukoutsidis and V.A. Siris. '802.11e EDCA Protocol Parameterization: A Modeling and Optimization Study'. In: *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*. 2007, pp. 1–9. DOI: 10.1109/WOWMOM.2007.4351767.

[117] Byung-Jae Kwak, Nah-Oak Song and M.E. Miller. 'Performance analysis of exponential backoff'. In: *Networking, IEEE/ACM Transactions on* 13.2 (2005), pp. 343–355. ISSN: 1063-6692. DOI: 10.1109/TNET.2005.845533.

[118] F. Lalanne and S. Maag. 'A Formal Data-Centric Approach for Passive Testing of Communication Protocols'. In: *Networking, IEEE/ACM Transactions on* 21.3 (2013), pp. 788–801. ISSN: 1063-6692. DOI: 10.1109/TNET.2012.2210443.

[119] D. Lee and K. Sabnani. 'Reverse-engineering of communication protocols'. In: *Network Protocols, 1993. Proceedings., 1993 International Conference on*. 1993, pp. 208–216. DOI: 10.1109/ICNP.1993.340917.

[120] H.-J. Lee et al. 'A QoS Provisioning Technique in a Residential Gateway using Heterogeneous Network Access Technology'. In: *TENCON 2005 2005 IEEE Region 10*. Nov. 2005, pp. 1–6. DOI: 10.1109/TENCON.2005.301170.

[121] Hyungho Lee et al. 'Wireless LAN with medical-grade QoS for e-healthcare'. In: *Communications and Networks, Journal of* 13.2 (Apr. 2011), pp. 149–159. ISSN: 1229-2370. DOI: 10.1109/JCN.2011.6157414.

[122] Jae Young Lee, Hwang-Soo Lee and Joong Soo Ma. 'Model-based QoS parameter control for IEEE 802.11e EDCA'. In: *Communications, IEEE Transactions on* 57.7 (July 2009), pp. 1914–1918. ISSN: 0090-6778. DOI: 10.1109/TCOMM.2009.07.070147.

[123] Bo Li et al. 'QoS enabled voice support in the next generation Internet: issues, existing approaches and challenges'. In: *Communications Magazine, IEEE* 38.4 (2000), pp. 54–61. ISSN: 0163-6804. DOI: 10.1109/35.833557.

[124] Maodong Li, Zhenzhong Chen and Yap-Peng Tan. 'Joint packet prioritization and QoS mapping for SVC over wlans'. In: *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. Mar. 2010, pp. 2298–2301. DOI: 10.1109/ICASSP.2010.5495965.

[125] Ming Li et al. 'QOAR: Adaptive QoS Scheme in Multi-Rate Wireless LANs'. In: *Communications, 2008. ICC '08. IEEE International Conference on*. 2008, pp. 2900–2904. DOI: 10.1109/ICC.2008.546.

[126] Qiong Li and M. van der Schaar. 'Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation'. In: *Multimedia, IEEE Transactions on* 6.2 (Apr. 2004), pp. 278–290. ISSN: 1520-9210. DOI: 10.1109/TMM.2003.822792.

[127] Zhao Li and T. Herfet. 'MAC Layer Multicast Error Control for IPTV in Wireless LANs'. In: *Broadcasting, IEEE Transactions on* 55.2 (2009), pp. 353–362. ISSN: 0018-9316. DOI: 10.1109/TBC.2009.2016502.

[128] Sam Liang and David Cheriton. 'TCP-RTM: Using TCP for real time multimedia applications'. In: *in International Conference on Network Protocols*. 2002.

[129] Wan-Seon Lim, Dong-Wook Kim and Young-Joo Suh. 'Design of Efficient Multicast Protocol for IEEE 802.11n WLANs and Cross-Layer Optimization for Scalable Video Streaming'. In: *Mobile Computing, IEEE Transactions on* 11.5 (2012), pp. 780–792. ISSN: 1536-1233. DOI: 10.1109/TMC.2011.95.

[130] Jia-Shi Lin and Kai-Ten Feng. 'QoS-Based Adaptive Contention/Reservation Medium Access Control Protocols for Wireless Local Area Networks'. In: *Mobile Computing, IEEE Transactions on* 10.12 (2011), pp. 1785–1803. ISSN: 1536-1233. DOI: 10.1109/TMC.2010.235.

[131] Jia-Shi Lin et al. 'Novel Design and Analysis of Aggregated ARQ Protocols for IEEE 802.11n Networks'. In: *Mobile Computing, IEEE Transactions on* 12.3 (2013), pp. 556–570. ISSN: 1536-1233. DOI: 10.1109/TMC.2012.34.

[132] A. Lombardo and G. Schembra. 'Performance evaluation of an adaptive-rate MPEG encoder matching IntServ traffic constraints'. In: *Networking, IEEE Transactions on* 11.1 (2003), pp. 47–65. ISSN: 1063-6692. DOI: 10.1109/TNET.2002.804830.

[133] R. MacKenzie, D. Hands and T. O'Farrell. 'An evaluation of quality of service for H.264 over 802.11e WLANs'. In: *Packet Video Workshop, 2009. PV 2009. 17th International*. May 2009, pp. 1–10. DOI: 10.1109/PACKET.2009.5152163.

[134] R. MacKenzie, D. Hands and T. O'Farrell. 'Packet handling strategies to improve video QoS over 802.11e WLANs'. In: *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*. Sept. 2009, pp. 1173–1177. DOI: 10.1109/PIMRC.2009.5449894.

[135] Jianbing Mao et al. 'An Adaptive MAC Scheme to Achieve Throughput Optimization and Airtime Fairness for IEEE 802.11 Multirate Networks'. In: *Wireless Networks and Information Systems, 2009. WNIS '09. International Conference on*. 2009, pp. 97–101. DOI: 10.1109/WNIS.2009.83.

[136] G. Marfia and M. Roccetti. 'TCP at last: reconsidering TCP's role for wireless entertainment centers at home'. In: *Consumer Electronics, IEEE Transactions on* 56.4 (Nov. 2010), pp. 2233–2240. ISSN: 0098-3063. DOI: 10.1109/TCE.2010.5681095.

[137] D.Q. Mayne et al. 'Constrained model predictive control: Stability and optimality'. In: *Automatica* 36.6 (2000), pp. 789–814. ISSN: 0005-1098. DOI: http://dx.doi.org/10.1016/S0005-1098(99)00214-9.

[138] Frank McSherry and Ratul Mahajan. 'Differentially-private network trace analysis'. In: *SIGCOMM Comput. Commun. Rev.* 41.4 (Aug. 2010), pages. ISSN: 0146-4833. URL: http://dl.acm.org/citation.cfm?id=2043164.1851199.

[139] H. Al-Mefleh and J.M. Chang. 'A New ACK Policy To Mitigate the Effects of Coexisting IEEE 802.11/802.11e Devices'. In: *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. 2008, pages. DOI: 10.1109/INFOCOM.2008.271.

[140] V. Miguel et al. 'Distribution of high-definition video in 802.11 wireless home networks'. In: *Consumer Electronics, IEEE Transactions on* 57.1 (Feb. 2011), pp. 53–61. ISSN: 0098-3063. DOI: 10.1109/TCE.2011.5735481.

[141] M. Mirhakkak, N. Schult and D. Thomson. 'Dynamic bandwidth management and adaptive applications for a variable bandwidth wireless environment'. In: *Selected Areas in Communications, IEEE Journal on* 19.10 (2001), pp. 1984–1997. ISSN: 0733-8716. DOI: 10.1109/49.957313.

[142] J. Misic and V.B. Misic. 'Bridge performance in a multitier wireless network for healthcare monitoring'. In: *Wireless Communications, IEEE* 17.1 (Feb. 2010), pp. 90–95. ISSN: 1536-1284. DOI: 10.1109/MWC.2010.5416355.

[143] Sungwook Moon and A. Helmy. 'Understanding Periodicity and Regularity of Nodal Encounters in Mobile Networks: A Spectral Analysis'. In: *Global Telecommunications Conference (GLOBECOM 2010)*. 2010, pp. 1–5. DOI: 10.1109/GLOCOM.2010.5684252.

[144] R. Mudumbai, G. Barriac and U. Madhow. 'Optimizing medium access control for rapid handoffs in pseudocellular networks'. In: *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*. Vol. 2. 2004, 1098–1102 Vol. 2. DOI: 10.1109/VETECF.2004.1400191.

[145] Sai Shankar N, J. del Prado Pavon and P. Wienert. 'Optimal packing of VoIP calls in an IEEE 802.11 a/e WLAN in the presence of QoS constraints and channel errors'. In: *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*. Vol. 5. 2004, 2974–2980 Vol.5. DOI: 10.1109/GLOCOM.2004.1378898.

[146] A. Nafaa and A. Ksentini. 'On Sustained QoS Guarantees in Operated IEEE 802.11 Wireless LANs'. In: *Parallel and Distributed Systems, IEEE Transactions on* 19.8 (Aug. 2008), pp. 1020–1033. ISSN: 1045-9219. DOI: 10.1109/TPDS.2007.70785.

[147] Anthony C. H. Ng, David Malone and Douglas J. Leith. 'Experimental Evaluation of TCP Performance and Fairness in an 802.11e Test-bed'. In: *Proceedings of the 2005 ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis*. E-WIND '05. Philadelphia, Pennsylvania, USA: ACM, 2005, pp. 17–22. ISBN: 1-59593-026-4. DOI: 10.1145/1080148.1080152. URL: http://doi.acm.org/10.1145/1080148.1080152.

[148] S.H. Nguyen, H.L. Vu and L.L.H. Andrew. 'Performance Analysis of IEEE 802.11 WLANs With Saturated and Unsaturated Sources'. In: *Vehicular Technology, IEEE Transactions on* 61.1 (2012), pp. 333–345. ISSN: 0018-9545. DOI: 10.1109/TVT.2011.2174069.

[149] P. Nicopolitidis, G.I. Papadimitriou and A.S. Pomportsis. 'Distributed protocols for ad hoc wireless LANs: a learning-automata-based approach'. In: *Ad Hoc Networks* 2.4 (2004), pp. 419–431. ISSN: 1570-8705. DOI: http://dx.doi.org/10.1016/j.adhoc.2003.09.004. URL: http://www.sciencedirect.com/science/article/pii/S1570870503000593.

[150] P. Nicopolitidis, G.I. Papadimitriou and A.S. Pomportsis. 'Learning automata-based polling protocols for wireless LANs'. In: *Communications, IEEE Transactions on* 51.3 (2003), pp. 453–463. ISSN: 0090-6778. DOI: 10.1109/TCOMM.2003.809788.

[151] P. Nicopolitidis et al. 'Adaptive wireless networks using learning automata'. In: *Wireless Communications, IEEE* 18.2 (2011), pp. 75–81. ISSN: 1536-1284. DOI: 10.1109/MWC.2011.5751299.

[152] A. Odlyzko. 'Paris Metro pricing: the minimalist differentiated services solution'. In: *Quality of Service, 1999. IWQoS '99. 1999 Seventh International Workshop on*. 1999, pp. 159–161. DOI: 10.1109/IWQOS.1999.766491.

[153] Byung Joon Oh and Chang Wen Chen. 'An Opportunistic Multi Rate MAC for reliable H.264/AVC video streaming over Wireless Mesh Networks'. In: *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*. 2009, pp. 1241–1244. DOI: 10.1109/ISCAS.2009.5117987.

[154] Eng Hwee Ong, J.Y. Khan and K. Mahata. 'Radio Resource Management of Composite Wireless Networks: Predictive and Reactive Approaches'. In: *Mobile Computing, IEEE Transactions on* 11.5 (2012), pp. 807–820. ISSN: 1536-1233. DOI: 10.1109/TMC.2011.87.

[155] Ruoming Pang et al. 'The devil and packet trace anonymization'. In: *SIGCOMM Comput. Commun. Rev.* 36.1 (Jan. 2006), pp. 29–38. ISSN: 0146-4833. DOI: 10.1145/1111322.1111330. URL: http://doi.acm.org/10.1145/1111322.1111330.

[156] Maria Papadopouli, Haipeng Shen and Manolis Spanakis. 'Characterizing the duration and association patterns of wireless access in a campus'. In: *Wireless Conference 2005 - Next Generation Wireless and Mobile Communications and Services (European Wireless), 11th European*. 2005, pp. 1–7.

[157] Juyoung Park, Jaemyoun Lee and Kyungtae Kang. 'Designing and predicting QoS of a wireless system for medical telemetry'. In: *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*. Oct. 2012, pp. 737–744. DOI: 10.1109/LCNW.2012.6424058.

[158] Christina Parsa and J.J. Garcia-Luna-Aceves. 'Improving TCP performance over wireless networks at the link layer'. In: *Mobile Networks and Applications* 5.1 (2004), pp. 57–71. ISSN: 1383-469X. DOI: 10.1023/A:1019131822786. URL: http://dx.doi.org/10.1023/A:1019131822786.

[159] W. Pattara-Atikom, P. Krishnamurthy and S. Banerjee. 'Distributed mechanisms for quality of service in wireless LANs'. In: *Wireless Communications, IEEE* 10.3 (June 2003), pp. 26–34. ISSN: 1536-1284. DOI: 10.1109/MWC.2003.1209593.

[160] Fei Peng. 'Theoretical performance evaluation of EDCA in IEEE 802.11e wireless LANs'. In: *European Transactions on Telecommunications* 21.3 (2010), pp. 266–275. ISSN: 1541-8251. DOI: 10.1002/ett.1386. URL: http://dx.doi.org/10.1002/ett.1386.

[161] Fei Peng, H.M. Alnuweiri and V.C.M. Leung. 'Analysis of Burst Transmission in IEEE 802.11e Wireless LANs'. In: *Communications, 2006. ICC '06. IEEE International Conference on*. Vol. 2. 2006, pp. 535–539. DOI: 10.1109/ICC.2006.254850.

[162] A.G. Prieto and R. Stadler. 'Adaptive real-time monitoring for large-scale networked systems'. In: *Integrated Network Management, 2009. IM '09. IEEE International Symposium on*. 2009, pp. 790–795. DOI: 10.1109/INM.2009.5188884.

[163] Wei Pu, Zixuan Zou and Chang Wen Chen. 'New TCP video streaming proxy design for last-hop wireless networks'. In: *Image Processing (ICIP), 2011 18th IEEE International Conference on*. Sept. 2011, pp. 2225–2228. DOI: 10.1109/ICIP.2011.6116079.

[164] I. Radovanovic, R. Verhoeven and J. Lukkien. 'Improving TCP performance over last-hop wireless networks for live video delivery'. In: *Consumer Electronics, IEEE Transactions on* 54.3 (2008), pp. 1139–1147. ISSN: 0098-3063. DOI: 10.1109/TCE.2008.4637599.

[165] S. Rashwand and J. Misic. 'IEEE 802.11e EDCA Under Bursty Traffic — How Much TXOP Can Improve Performance'. In: *Vehicular Technology, IEEE Transactions on* 60.3 (2011), pp. 1099–1115. ISSN: 0018-9545. DOI: 10.1109/TVT.2011.2107928.

[166] S. Rashwand and J. Misic. 'Two-tier WBAN/WLAN healthcare networks; priority considerations'. In: *Global Communications Conference (GLOBECOM), 2012 IEEE*. Dec. 2012, pp. 5398–5403. DOI: 10.1109/GLOCOM.2012.6503979.

[167] J.W. Robinson and T.S. Randhawa. 'A practical model for transmission delay of IEEE 802.11e enhanced distributed channel access'. In: *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*. Vol. 1. 2004, 323–328 Vol.1. DOI: 10.1109/PIMRC.2004.1370887.

[168] J.W. Robinson and T.S. Randhawa. 'Saturation throughput analysis of IEEE 802.11e enhanced distributed coordination function'. In: *J. Select. Areas Commun., IEEE Journal on* 22.5 (June 2004), pp. 917–928. ISSN: 0733-8716. DOI: 10.1109/JSAC.2004.826929.

[169] L. Romdhani and C. Bonnet. 'Performance Analysis and Optimization of the 802.11e EDCA Transmission Opportunity (TXOP) Mechanism'. In: *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on*. 2007, pp. 68–68. DOI: 10.1109/WIMOB.2007.4390862.

[170] T. Sanguankotchakorn and K. Pudashine. 'A cross-layer framework for efficient MPEG-4 video streaming over IEEE 802.11e in MANETs'. In: *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2011 8th International Conference on*. May 2011, pp. 369–372. DOI: 10.1109/ECTICON.2011.5947851.

[171] C.M. Sarraf et al. 'Quality of service mapping between UMTS, WiMAX and IP-based network in Heterogeneous Wireless Networks'. In: *Intelligent and Advanced Systems (ICIAS), 2012 4th International Conference on*. Vol. 1. June 2012, pp. 291–295. DOI: 10.1109/ICIAS.2012.6306204.

[172] G. Sawma et al. 'Autonomic Management for Capacity Improvement in Wireless Networks'. In: *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*. 2009, pp. 1–6. DOI: 10.1109/CCNC.2009.4784696.

[173] L. Scalia and I. Tinnirello. 'A Low-level Simulation Study of Prioritization in IEEE 802.11e Contention-based Networks'. In: *Communication System Software and Middleware, 2006. Comsware 2006. First International Conference on*. 2006, pp. 1–7. DOI: 10.1109/COMSWA.2006.1665237.

[174] L. Scalia et al. 'Dynamic MAC Parameters Configuration for Performance Optimization in 802.11e Networks'. In: *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*. 2006, pp. 1–6. DOI: 10.1109/GLOCOM.2006.751.

[175] M. van der Schaar et al. 'Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs'. In: *Selected Areas in Communications, IEEE Journal on* 21.10 (2003), pp. 1752–1763. ISSN: 0733-8716. DOI: 10.1109/JSAC.2003.815231.

[176] S. Schuetz et al. 'Autonomic and decentralized management of wireless access networks'. In: *Network and Service Management, IEEE Transactions on* 4.2 (2007), pp. 96–106. ISSN: 1932-4537. DOI: 10.1109/TNSM.2007.070905.

[177] P. Seeling and M. Reisslein. 'Video Transport Evaluation With H.264 Video Traces'. In: *IEEE Communications Surveys Tutorials* PP.99 (2011), pp. 1–24. ISSN: 1553-877X. DOI: 10.1109/SURV.2011.082911.00067.

[178] P. Serrano et al. 'Optimal Configuration of 802.11e EDCA for Real-Time and Data Traffic'. In: *Vehicular Technology, IEEE Transactions on* 59.5 (June 2010), pp. 2511–2528. ISSN: 0018-9545. DOI: 10.1109/TVT.2010.2043274.

[179] R. Shankar et al. 'Quality of Service Enhancement for Converging Traffic in EDCA Based IEEE 802.11'. In: *Process Automation, Control and Computing (PACC), 2011 International Conference on*. July 2011, pp. 1–6. DOI: 10.1109/PACC.2011.5978859.

[180] Chong Shen, Dirk Pesch and James Irvine. 'A Framework for Self-Management of Hybrid Wireless Networks Using Autonomic Computing Principles'. In: *Proceedings of the 3rd Annual Communication Networks and Services Research Conference*. CNSR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 261–266. ISBN: 0-7695-2333-1. DOI: 10.1109/CNSR.2005.8. URL: http://dx.doi.org/10.1109/CNSR.2005.8.

[181] A.A. El-Sherif and K.J.R. Liu. 'Cooperation in Random Access Networks: Protocol Design and Performance Analysis'. In: *Selected Areas in Communications, IEEE Journal on* 30.9 (Oct. 2012), pp. 1694–1702. ISSN: 0733-8716. DOI: 10.1109/JSAC.2012.121013.

[182] Shiann T. Sheu and Tzu F. Sheu. 'A bandwidth allocation/sharing/extension protocol for multimedia over IEEE 802.11 ad hoc wireless LANs'. In: *Selected Areas in Communications, IEEE Journal on* 19.10 (2001), pp. 2065–2080. ISSN: 0733-8716. DOI: 10.1109/49.957320.

[183] H. Shimonishi, T. Hama and T. Murase. 'TCP Congestion Control Enhancements for Streaming Media'. In: *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE*. Jan. 2007, pp. 303–307. DOI: 10.1109/CCNC.2007.66.

[184] Pilgyu Shin and Kwangsue Chung. 'A Cross-Layer Based Rate Control Scheme for MPEG-4 Video Transmission by Using Efficient Bandwidth Estimation in IEEE 802.11e'. In: *Information Networking, 2008. ICOIN 2008. International Conference on*. Jan. 2008, pp. 1–5. DOI: 10.1109/ICOIN.2008.4472761.

[185] Guoqiang Shu and D. Lee. 'A Formal Methodology for Network Protocol Fingerprinting'. In: *Parallel and Distributed Systems, IEEE Transactions on* 22.11 (2011), pp. 1813–1825. ISSN: 1045-9219. DOI: 10.1109/TPDS.2011.26.

[186] J.P. Singh et al. 'TCP Performance Dynamics and Link-Layer Adaptation Based Optimization Methods for Wireless Networks'. In: *Wireless Communications, IEEE Transactions on* 6.5 (2007), pp. 1864–1879. ISSN: 1536-1276. DOI: 10.1109/TWC.2007.360388.

[187] R. Sivakumar et al. 'Achieving per-flow weighted rate fairness in a core stateless network'. In: *Distributed Computing Systems, 2000. Proceedings. 20th International Conference on*. 2000, pp. 188–196. DOI: 10.1109/ICDCS.2000.840929.

[188] D. Skyrianoglou, N. Passas and A. Salkintzis. 'Traffic scheduling for multimedia QoS over wireless LANs'. In: *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*. Vol. 2. 2005, 1266–1270 Vol. 2. DOI: 10.1109/ICC.2005.1494550.

[189] Adam Slagell, Kiran Lakkaraju and Katherine Luo. 'FLAIM: a multi-level anonymization framework for computer and network logs'. In: *Proceedings of the 20th conference on Large Installation System Administration*. LISA '06. Washington, DC: USENIX Association, 2006, pp. 6–6. URL: http://dl.acm.org/citation.cfm?id=1267793.1267799.

[190] Sunghwa Son, Kyung-Joon Park and Eun-Chan Park. 'Design of adaptive IEEE 802.11 WLAN in hospital environments'. In: *e-Health Networking, Applications Services (Healthcom), 2013 IEEE 15th International Conference on*. Oct. 2013, pp. 722–724. DOI: 10.1109/HealthCom.2013.6720771.

[191] Ion Stoica and Hui Zhang. 'Providing Guaranteed Services Without Per Flow Management'. In: *SIGCOMM Computer Communication Review* 29.4 (Aug. 1999), pp. 81–94. ISSN: 0146-4833. DOI: 10.1145/316194.316208. URL: http://doi.acm.org/10.1145/316194.316208.

[192] Peter van der Stok et al. 'Hierarchical Resource Allocation for Robust In-home Video Streaming'. In: *J. Syst. Softw.* 80.7 (July 2007), pp. 951–961. ISSN: 0164-1212. DOI: 10.1016/j.jss.2006.09.026. URL: http://dx.doi.org/10.1016/j.jss.2006.09.026.

[193] P. Stuckmann et al. 'The EUREKA Gandalf project: monitoring and self-tuning techniques for heterogeneous radio access networks'. In: *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*. Vol. 4. 2005, 2570–2574 Vol. 4. DOI: 10.1109/VETECS.2005.1543799.

[194] Christian Hoene Sven Wietholter Marc Emmelmann and Adam Wolisz. *TKN EDCA Model for ns-2*. TKN Technical Report Series TKN-06-003. Telecommunication Networks Group, Technische Universitat Berlin, June 2006.

[195] R. Taank and Xiao-Hong Peng. 'An Experimental Evaluation of Sender-Side TCP Enhancements for Wired-to-Wireless Paths: A Real-World Home WLAN Case Study'. In: *Advanced Information Networking and Applications, 2009. AINA '09. International Conference on*. May 2009, pp. 323–330. DOI: 10.1109/AINA.2009.51.

[196] N.C. Taher et al. 'First step towards an efficient admission control: A complete analytical model for 802.11e EDCA for throughput and delay prediction'. In: *Integrated Network Management-Workshops, 2009. IM '09. IFIP/IEEE International Symposium on*. 2009, pp. 217–224. DOI: 10.1109/INMW.2009.5195963.

[197] Zhifeng Tao and S. Panwar. 'Throughput and Delay Analysis for the IEEE 802.11e Enhanced Distributed Channel Access'. In: *Communications, IEEE Transactions on* 54.4 (2006), pp. 596–603. ISSN: 0090-6778. DOI: 10.1109/TCOMM.2006.873066.

[198] Microsoft TechNet. *What Is QoS?* http://technet.microsoft.com. [Online; accessed 29October2013].

[199] 'The network simulator (ns2)'. In: (). URL: http://www.isi.edu/nsnam/ns.

[200] 'Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks'. In: *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2005. WIOPT 2005. Third International Symposium on*. 2005, pages. DOI: 10.1109/WIOPT.2005.48.

[201] I. Tinnirello and G. Bianchi. 'Rethinking the IEEE 802.11e EDCA Performance Modeling Methodology'. In: *Networking, IEEE/ACM Transactions on* 18.2 (Apr. 2010), pp. 540–553. ISSN: 1063-6692. DOI: 10.1109/TNET.2009.2029101.

[202] I. Tinnirello, G. Bianchi and Yang Xiao. 'Refinements on IEEE 802.11 Distributed Coordination Function Modeling Approaches'. In: *Vehicular Technology, IEEE Transactions on* 59.3 (2010), pp. 1055–1067. ISSN: 0018-9545. DOI: 10.1109/TVT.2009.2029118.

[203] A.L. Toledo, T. Vercauteren and Xiaodong Wang. 'Adaptive Optimization of IEEE 802.11 DCF Based on Bayesian Estimation of the Number of Competing Terminals'. In: *IEEE Transactions on Mobile Computing* 5.9 (Sept. 2006), pp. 1283–1296. ISSN: 1536-1233. DOI: 10.1109/TMC.2006.124.

[204] Konstantinos Triantafyllidis, Egor Bondarev and Peter H.N. de With. 'Performance analysis method for RT systems: Promartes for autonomous robot'. In: *Specification Design Languages (FDL), 2013 Forum on*. 2013, pp. 1–8.

[205] H. Trsek et al. 'A flexible approach for real-time wireless communications in adaptable industrial automation systems'. In: *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*. Sept. 2011, pp. 1–4. DOI: 10.1109/ETFA.2011.6059152.

[206] Li-Ping Tung, Y.S. Sun and Meng Chang Chen. 'Dynamic bandwidth reservation scheme in 802.11 and 802.16 interworking networks'. In: *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*. 2011, pp. 386–391. DOI: 10.1109/WCNC.2011.5779194.

[207] M.F. Tuysuz and H.A. Mantar. 'Evaluation of cross layer QoS aproachs for improving voice quality over multi-rate WLANs'. In: *Computer Engineering and Systems (ICCES), 2010 International Conference on*. 2010, pp. 73–78. DOI: 10.1109/ICCES.2010.5674893.

[208] M. Umlauft and P. Reichl. 'Experiences with the ns-2 network simulator - explicitly setting seeds considered harmful'. In: *Wireless Telecommunications Symposium, 2007. WTS 2007.* Apr. 2007, pp. 1–5. DOI: 10.1109/WTS.2007.4563316.

[209] G. Van der Auwera, P.T. David and M. Reisslein. 'Traffic and Quality Characterization of Single-Layer Video Streams Encoded with the H.264/MPEG-4 Advanced Video Coding Standard and Scalable Video Coding Extension'. In: *IEEE Transactions on Broadcasting* 54.3 (Sept. 2008), pp. 698–718. ISSN: 0018-9316. DOI: 10.1109/TBC.2008.2000422.

[210] R. Van Nee. 'Breaking the Gigabit-per-second barrier with 802.11AC'. In: *Wireless Communications, IEEE* 18.2 (2011), pp. 4–4. ISSN: 1536-1284. DOI: 10.1109/MWC.2011.5751287.

[211] Gianluca Vannuccini et al. 'Performance evaluation of a link adaptation scheme in IEEE 802.11e over IEEE 802.11a WLANs with QoS support'. In: *European Transactions on Telecommunications* 18.3 (2007), pp. 305–317. ISSN: 1541-8251. DOI: 10.1002/ett.1154. URL: http://dx.doi.org/10.1002/ett.1154.

[212] A. Veres et al. 'Supporting service differentiation in wireless packet networks using distributed control'. In: *Selected Areas in Communications, IEEE Journal on* 19.10 (2001), pp. 2081–2093. ISSN: 0733-8716. DOI: 10.1109/49.957321.

[213] S. Veres and D. Ionescu. 'A Performance Model and Measurement Framework for DiffServ Implementations'. In: *Instrumentation and Measurement, IEEE Transactions on* 56.4 (2007), pp. 1473–1480. ISSN: 0018-9456. DOI: 10.1109/TIM.2007.900422.

[214] José Villalón, Pedro Cuenca and Luis Orozco-Barbosa. 'On the capabilities of IEEE 802.11e for multimedia communications over heterogeneous 802.11/ 802.11e WLANs'. In: *Telecommunication Systems* 36.1-3 (2007), pp. 27–38. ISSN: 1018-4864. DOI: 10.1007/s11235-007-9059-8. URL: http://dx.doi.org/10.1007/s11235-007-9059-8.

[215] José Villalón et al. 'B-EDCA: A QoS mechanism for multimedia communications over heterogeneous 802.11/802.11e {WLANs}'. In: *Computer Communications* 31.17 (2008), pp. 3905–3921. ISSN: 0140-3664. DOI: http://dx.doi.org/10.1016/j.comcom.2008.07.012. URL: http://www.sciencedirect.com/science/article/pii/S0140366408004209.

[216] V.M. Vishnevsky and A.I. Lyakhov. 'IEEE 802.11 Wireless LAN: Saturation Throughput Analysis with Seizing Effect Consideration'. In: *Cluster Computing* 5.2 (2002), pp. 133–144. ISSN: 1386-7857. URL: http://dx.doi.org/10.1023/A:1013977425774.

[217] Mehmet C. Vuran and I.F. Akyildiz. 'Error Control in Wireless Sensor Networks: A Cross Layer Analysis'. In: *Networking, IEEE/ACM Transactions on* 17.4 (2009), pp. 1186–1199. ISSN: 1063-6692. DOI: 10.1109/TNET.2008.2009971.

[218] G.C. Walsh and Hong Ye. 'Scheduling of networked control systems'. In: *Control Systems, IEEE* 21.1 (2001), pp. 57–65. ISSN: 1066-033X. DOI: 10.1109/37.898792.

[219] G.C. Walsh, Hong Ye and L.G. Bushnell. 'Stability analysis of networked control systems'. In: *Control Systems Technology, IEEE Transactions on* 10.3 (2002), pp. 438–446. ISSN: 1063-6536. DOI: 10.1109/87.998034.

[220] Bing Wang et al. 'Multimedia Streaming via TCP: An Analytic Performance Study'. In: *ACM Trans. Multimedia Comput. Commun. Appl.* 4.2 (May 2008), 16:1–16:22. ISSN: 1551-6857. DOI: 10.1145/1352012.1352020. URL: http://doi.acm.org/10.1145/1352012.1352020.

[221] Chonggang Wang, Bo Li and Lemin Li. 'A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF'. In: *Vehicular Technology, IEEE Transactions on* 53.4 (2004), pp. 1235–1246. ISSN: 0018-9545. DOI: 10.1109/TVT.2004.830951.

[222] Haidong Wang, Guizhong Liu and Qinli Wang. 'An improved cross-layer mapping mechanism for packet video delivery over WLAN'. In: *Image Processing (ICIP), 2011 18th IEEE International Conference on*. Sept. 2011, pp. 941–944. DOI: 10.1109/ICIP.2011.6116716.

[223] Ping Wang, Hai Jiang and Weihua Zhuang. 'IEEE 802.11e enhancement for voice service'. In: *Wireless Communications, IEEE* 13.1 (2006), pp. 30–35. ISSN: 1536-1284. DOI: 10.1109/MWC.2006.1593522.

[224] Shao C. Wang and A. Helmy. 'Performance Limits and Analysis of Contention-based IEEE 802.11 MAC'. In: *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*. Nov. 2006, pp. 418–425. DOI: 10.1109/LCN.2006.322129.

[225] M. Weiser. 'The computer for the 21st Century'. In: *Pervasive Computing, IEEE* 1.1 (2002), pp. 19–25. ISSN: 1536-1268. DOI: 10.1109/MPRV.2002.993141.

[226] M. Welzl and M. Muhlhauser. 'Scalability and quality of service: a trade-off?' In: *Communications Magazine, IEEE* 41.6 (2003), pp. 32–36. ISSN: 0163-6804. DOI: 10.1109/MCOM.2003.1204745.

[227] M. Woehrle et al. 'EvAnT: Analysis and Checking of Event Traces for Wireless Sensor Networks'. In: *Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC '08. IEEE International Conference on*. 2008, pp. 201–208. DOI: 10.1109/SUTC.2008.24.

[228] J.-L.C. Wu, Hung-Huan Liu and Yi-Jen Lung. 'An adaptive multirate IEEE 802.11 wireless LAN'. In: *Information Networking, 2001. Proceedings. 15th International Conference on*. 2001, pp. 411–418. DOI: 10.1109/ICOIN.2001.905459.

[229] Mingquan Wu et al. 'IPTV Multicast Over Wireless LAN Using Merged Hybrid ARQ With Staggered Adaptive FEC'. In: *Broadcasting, IEEE Transactions on* 55.2 (2009), pp. 363–374. ISSN: 0018-9316. DOI: 10.1109/TBC.2009.2016500.

[230] Yong Xi et al. 'Adaptive Multirate Auto Rate Fallback Protocol for IEEE 802.11 WLANS'. In: *Military Communications Conference, 2006. MILCOM 2006. IEEE*. 2006, pp. 1–7. DOI: 10.1109/MILCOM.2006.302449.

[231] Yang Xiao. 'Performance analysis of priority schemes for IEEE 802.11 and IEEE 802.11e wireless LANs'. In: *Wireless Communications, IEEE Transactions on* 4.4 (2005), pp. 1506–1515. ISSN: 1536-1276. DOI: 10.1109/TWC.2005.850328.

[232] Dongxia Xu et al. 'An Access Delay Model for IEEE 802.11e EDCA'. In: *Mobile Computing, IEEE Transactions on* 8.2 (2009), pp. 261–275. ISSN: 1536-1233. DOI: 10.1109/TMC.2008.108.

[233] Duck-Yong Yang et al. 'Performance enhancement of multirate IEEE 802.11 WLANs with geographically scattered stations'. In: *Mobile Computing, IEEE Transactions on* 5.7 (2006), pp. 906–919. ISSN: 1536-1233. DOI: 10.1109/TMC.2006.101.

[234] Jie Yang et al. 'A flexible and distributed architecture for adaptive end-to-end QoS provisioning in next-generation networks'. In: *Selected Areas in Communications, IEEE Journal on* 23.2 (2005), pp. 321–333. ISSN: 0733-8716. DOI: 10.1109/JSAC.2004.839422.

[235] Xiaoyan Yin, Xiaomin Ma and K.S. Trivedi. 'An Interacting Stochastic Models Approach for the Performance Evaluation of DSRC Vehicular Safety Communication'. In: *Computers, IEEE Transactions on* 62.5 (2013), pp. 873–885. ISSN: 0018-9340. DOI: 10.1109/TC.2012.37.

[236] Hayoung Yoon and Jongwon Kim. 'Collaborative streaming-based media content sharing in WiFi-enabled home networks'. In: *Consumer Electronics, IEEE Transactions on* 56.4 (Nov. 2010), pp. 2193–2200. ISSN: 0098-3063. DOI: 10.1109/TCE.2010.5681090.

[237] Fang Yu and D. Lee. 'Internet Attack Traceback - Cross-validation and Pebble Tracing'. In: *Technologies for Homeland Security, 2008 IEEE Conference on*. 2008, pp. 378–383. DOI: 10.1109/THS.2008.4534481.

[238] Xiaobo Yu, P. Navaratnam and K. Moessner. 'Resource Reservation Schemes for IEEE 802.11-Based Wireless Networks: A Survey'. In: *Communications Surveys Tutorials, IEEE* 15.3 (2013), pp. 1042–1061. ISSN: 1553-877X. DOI: 10.1109/SURV.2012.111412.00029.

[239] Ganzhao Yuan et al. 'Low-rank mechanism: optimizing batch queries under differential privacy'. In: *Proc. VLDB Endow.* 5.11 (July 2012), pp. 1352–1363. ISSN: 2150-8097. URL: http://dl.acm.org/citation.cfm?id=2350229.2350252.

[240] Zhenhui Yuan and G.-M. Muntean. 'A Prioritized Adaptive Scheme for Multimedia Services over IEEE 802.11 WLANs'. In: *Network and Service Management, IEEE Transactions on* 10.4 (Dec. 2013), pp. 340–355. ISSN: 1932-4537. DOI: 10.1109/TNSM.2013.110513.130490.

[241] Hongqiang Zhai, X. Chen and Yuguang Fang. 'How well can the IEEE 802.11 wireless LAN support quality of service?' In: *Wireless Communications, IEEE Transactions on* 4.6 (2005), pp. 3084–3094. ISSN: 1536-1276. DOI: 10.1109/TWC.2005.857994.

[242] Zhensheng Zhang. 'DTRA: directional transmission and reception algorithms in WLANs with directional antennas for QoS support'. In: *Network, IEEE* 19.3 (2005), pp. 27–32. ISSN: 0890-8044. DOI: 10.1109/MNET.2005.1453396.

[243] Zhi-Li Zhang, Z. Duan and Y.T. Hou. 'Virtual time reference system: a unifying scheduling framework for scalable support of guaranteed services'. In: *Selected Areas in Communications, IEEE Journal on* 18.12 (2000), pp. 2684–2695. ISSN: 0733-8716. DOI: 10.1109/49.898750.

[244] Liqiang Zhao et al. 'Game-Theoretic EDCA in IEEE 802.11e WLANs'. In: *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*. Sept. 2008, pp. 1–5. DOI: 10.1109/VETECF.2008.428.

[245] Yu Zheng et al. 'Performance Analysis of IEEE 802.11 DCF in Imperfect Channels'. In: *Vehicular Technology, IEEE Transactions on* 55.5 (2006), pp. 1648–1656. ISSN: 0018-9545. DOI: 10.1109/TVT.2006.878606.

[246] Hao Zhu and Guohong Cao. 'rDCF: A Relay-Enabled Medium Access Control Protocol for Wireless Ad Hoc Networks'. In: *Mobile Computing, IEEE Transactions on* 5.9 (2006), pp. 1201–1214. ISSN: 1536-1233. DOI: 10.1109/TMC.2006.137.

[247] Hua Zhu and I. Chlamtac. 'Performance analysis for IEEE 802.11e EDCF service differentiation'. In: *Wireless Communications, IEEE Transactions on* 4.4 (2005), pp. 1779–1788. ISSN: 1536-1276. DOI: 10.1109/TWC.2005.847113.

[248] Eustathia Ziouva and Theodore Antonakopoulos. 'CSMA/CA performance under high traffic conditions: throughput and delay analysis'. In: *Computer Communications* 25.3 (2002), pp. 313–321. ISSN: 0140-3664. DOI: http://dx.doi.org/10.1016/S0140-3664(01)00369-3. URL: http://www.sciencedirect.com/science/article/pii/S0140366401003693.

Publications

[1] W. Aslam, J. Lukkien and I. Radovanovic. 'Allocation of in-home network resources to competing multimedia streams using EDCA'. In: *Consumer Electronics, 2009. ICCE '09. Digest of Technical Papers International Conference on*. Jan. 2009, pp. 1–2. DOI: 10.1109/ICCE.2009.5012376.

[2] W. Aslam and J.J. Lukkien. 'Streaming-aware channel utilization improvement for wireless home networks'. In: *Consumer Electronics, IEEE Transactions on* 58.1 (2012), pp. 69–77. ISSN: 0098-3063. DOI: 10.1109/TCE.2012.6170057.

[3] Waqar Aslam and Johan J. Lukkien. 'Delay Constrained Resource Allocation for Wireless Home Networks'. In: *2012 IEEE International Conference on Consumer Electronics (ICCE) (ICCE 2012)*. Las Vegas, USA, Jan. 2012, pp. 98–99.

[4] Waqar Aslam and Johan J. Lukkien. 'Scavenging the Channel Share: Performance on Sensing and Allocation'. In: *Proceedings of the first Taibah University International Conference on Computing and Information Technology (ICCIT 2012)*. Al-Madinah Al-Munawwarah, Saudi Arabia, Dec. 2012, pp. 834–837.

[5] K. Triantafyllidis and W. Aslam and E. Bondarev and P.H.N. de With and J.J. Lukkien. 'Reliable Performance Prediction for Component-Based Distributed Software Architectures'. In: *Software Engineering, IEEE Transactions on* 40.7 (submitted - July 2014).

To be submitted

[6] W. Aslam, J.J. Lukkien and T. Özçelebi. 'A formal performance analysis framework for wireless networks'. In: *Formal Aspects of Computing* (2014).

[7] W. Aslam, J.J. Lukkien and T. Özçelebi. 'A trace formalization to enable performance analysis of wireless networks'. In: *Formal Aspects of Computing* (2014).

[8] W. Aslam, J.J. Lukkien and T. Özçelebi. 'Finding and optimizing network settings in unmanaged wireless networks to satisfy QoS requirements'. (2014).

[9] W. Aslam, J.J. Lukkien and T. Özçelebi. 'Mapping QoS requirements to network settings in unmanaged wireless networks'. In: *Consumer Electronics, IEEE Transactions on* (2014).

Acknowledgements

الحمد لله رب العلمين

I have never been short of a motivation to undertake a research activity. Looking back at the journey that I embarked upon a few years ago at the TU/e, there has been one permanent scene: confrontation with yet more challenges amidst an air of uncertainty. This journey has offered me a chance of self-actualization, towards better understanding science, mathematics, world, its people and my limitations. In this journey, I have received support from a few people to whom I owe many thanks.

Foremost, I would like to express sincere gratitude to my promotor and advisor, prof.dr. J.J. Lukkien — for giving me a chance to grow directly under his great scientific advice, mentoring, mathematician's perspective, pulling me out from those tons of details to the system level, support and enthusiasm. Johan, I highly value your genuine interest in minute details of models and results, and your time and effort to review my publications and this thesis. I have enjoyed our technical discussions towards intuition building on the white board in your office — your positive and open attitude is impressive. I have also cherished your concluding remarks in the SAN colloquiums — they are marvelous. Thank you for organizing the context of my research and stay.

I am grateful to my co-advisor, dr. T. Özçelebi — for his support, valuable suggestions and persuasive criticism. Tanir, I have lots of appreciation for your thoroughness in reviewing my publications and this thesis — it has helped greatly to sharpen up. Thanks for my progress meetings together with Johan and for your time and efforts during discussions in your office. Your comments have provided esteemed insights that are simply necessary. I also owe special thanks to dr. I. Radovanović (former co-advisor) for his guidance in research during the initial phase.

I would like to thank the members of my reading committee, prof.dr.ir. I.G.M.M. Niemegeers, dr. P. Corcoran, prof.dr. A. Liotta and prof.dr.ir. C.H. van Berkel, for assessing this thesis and insightful comments.

I wish to thank Dr. R. Verhoeven for his invaluable support, whether it be software or hardware issues, technical inquiries/discussions or reviews. He is indeed an asset for the group. I am also indebted to Dr. R.H. Mak, for welcoming me whenever I would knock on his door for a discussion. My thanks to our group secretaries, Cecile (former) and Anjolein, for their support during my stay — whether it be documentation, fresh coffee or arrangements for informal gatherings in SAN and annual social events.

I acknowledge the help from Wil Kortsmit for 'speaking' Mathematica to me that eventually I could understand enough. He is indeed a wonderful person with a very positive attitude. Wil, thanks for those cups of fresh coffee and introducing me to awesome hyacinths. I take this opportunity to thank Dr. J. Draisma for his brilliance and enthusiasm towards some of my open-ended food for thought.

To the SAN colleagues, you supported a commendable working atmosphere. My office-mates, especially René and his friend, Dirk, and Nataša — thanks for interesting discussions and impulsive talks. Sachin and Ionut, thanks for those random coffee/tea pauses and lunch walks that helped break the monotony.

There are two incidents that hampered my progress: theft of my laptop along with the backup devices and illness leading to hospitalization. Special thanks to the help desk in BCF at TU/e, and the doctors and the support staff in CZH Eindhoven, to help me recover. I hold high esteem for a number of people who visited me in the hospital — Johan, Tanir, Anjolein, Richard, Nataša, Vinh, Saskia, Imran, Zahid (my cousin from London), Oemar bhai, Shahid sahab, Wasim, Taimur, Ibrahim, Hussain and many more — some even daily. To Peter, Reinder, Pieter, Aravind, Martijn, thanks for your concerns about my health.

For offering some nice moments, my thanks to those 'extracurricular' friends — the mother nature and greenery, bike culture, the members of Tuesday Night Dinner Club and Dars group. I offer my heartfelt thanks to all my teachers at school, college, universities or otherwise — stay always blessed.

I gratefully acknowledge the funding sources that made it all possible: the Higher Education Commission (HEC) of Pakistan (primarily), OMECA and COMET projects. To Loes Minkman of NUFFIC, who has been the coordinator of the HEC programme in The Netherlands, thanks for your help and understanding.

Finally, to my support system back home that I could always count on, despite instability in the country due to recent events upon turn of the century. My buddies, Masood, Tahir, Moazzam and others, you guys have always been there for me — thanks. I owe thanks to my in-laws for their well wishes. With gratitude, I thank my brothers and sisters for their prayers for my success. To my wife, Samra, with whom I started journey during PhD — thank you for your patience, sacrifice and unwavering support. I am short of words to express my immense gratitude for the endless love, unconditional support, sacrifice and prayers that I have received throughout my life, from Ammi gee and Abba gee, my parents.

Waqar Aslam
Technische Universiteit Eindhoven
May 2014

Curriculum Vitae

Waqar Aslam was born in Bahawalpur, Pakistan, where he also studied, until B.Sc. from Government Sadiq Egerton College. He received M.Sc. in Computer Science from Quaid-i-Azam University Islamabad, Pakistan — the final thesis for which, titled 'Visual Differential File Comparator', focused on comparison of text files to highlight their differences visually. Later, he started to teach as a Lecturer at the Department of Computer Science, The Islamia University of Bahawalpur, Pakistan.

In 2006, he achieved a scholarship for the doctoral degree under the scheme of Higher Education Commission, Pakistan. In this regard, he was selected to conduct research within the expertise area, System Architecture and Networking (SAN), at the Department of Mathematics and Computer Science (WIN) of Eindhoven University of Technology (TU/e), The Netherlands under the supervision of prof.dr. J.J. Lukkien. The research has addressed challenges to obtain a predictive control of distributed wireless networks for provisioning of Quality of Service, the results of which are presented in this thesis. His research interests lie in performance modelling, prediction and optimization of networks and systems.