UNIVERSITY OF CALIFORNIA, SAN DIEGO

Network Adaptation Techniques to Enhance Efficiency and Quality of Wireless Multimedia Transmissions

A dissertation submitted in partial satisfaction of the requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Computer Engineering)

by

Naomi Ann Suba Ramos

Committee in charge:

Professor Sujit Dey, Chair
Professor Rene Cruz
Professor Ramesh Rao
Professor Amin Vahdat
Professor Geoffrey M. Voelker

2007

The dissertation of Naomi Ann Suba Ramos is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____
Chair

University of California, San Diego

2007

DEDICATION

This dissertation is dedicated to my family; Mom, Dad and Marc, for all of their support and love. I thank them for teaching me the importance of a good education, for always believing in me, and for giving me the encouragement to overcome any challenge that came my way. I can never express my gratitude and love for all that they have done.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

There have been many people that have provided me with support and guidance through my doctoral program. I would first like to acknowledge the two people who have been instrumental to my research work. I am extremely grateful to my advisor, Professor Sujit Dey. I admire his enthusiasm for research and his willingness to venture and explore new research areas. He makes every effort to pass on his knowledge and experience to his students and is sincere in ensuring his students not only obtain a doctorate in the years that they work with him, but are able to have a valuable learning experience. I am thankful for having the opportunity to work with him these past years. I would next like to thank my collaborator and friend, Debashis Panigrahi. He has contributed significantly to this research work, and has always been there to discuss research ideas, answer questions, and provide feedback on my research. His ability to multi-task between different projects and the amount of energy he puts in all his projects are simply remarkable. In addition to being my colleague, he has been a close friend and has always been there to provide the needed support to get through all the challenges.

I would like to thank my thesis committee: Geoff Voelker, Ramesh Rao, Rene Cruz, and Amin Vahdat. They have given valuable feedback that has improved this work and have been extremely patient, flexible, and approachable throughout the doctoral process. Additionally, I would like to thank all the anonymous reviewers from the various conferences and journals mentioned in this thesis for your input and feedback about my work. I am grateful to the Center for Wireless Communications and UC Discovery for sponsoring my research work. Through their support, I was able to attend research conferences, participate in research reviews, and have discussions with representatives from industry.

I have had the opportunity to work alongside many bright and wonderful people at UCSD. I would like to thank the students, staff, and alumni of MESDAT lab: Kanishka Lahiri, Li Chen, Yi Zhao, Xiaoliang Bai, Clark Taylor, Dong-Gi Lee, Krishna

Sekar, Cathy MacHutchin, Chong Zhao, Shoubhik Mukhopadhay, Mayank Tiwari, and Saumya Chandra. I am grateful for the time that we have had together, whether it was spent in research discussions, coffee breaks, working on getting the lab together, and even the periodic venting sessions. You all made each work day more enjoyable and have been my colleagues and my friends throughout.

Additionally, I would like to acknowledge some friends who fall outside of the MESDAT lab. There are a number of them but I would like to specifically thank my roommates, Valeria Liverini and Dave Kauchak, and my best friend, Johann Ammerlahn, for being there to listen and give advice. I would not have been able to make it through this process without them. I would also like to thank my loving boyfriend, Jeff Nichols, who has been extremely patient and understanding with my work and time constraints. While it has been non-ideal to live apart these past few years, I am grateful that we have made it through and am excited at the prospect of finally ending up in the same place.

Finally and most importantly, I would like to thank my family; Mom, Dad, and Marc. I have dedicated this thesis to them in gratitude for all of their support and love.

The text of the following chapters, in part or in full, is based on material that has been published in conference proceedings or journals, or is currently in review. Chapter 2 is based on material published in the IASTED International Conference on Wireless and Optical Communication in 2003 (N. Ramos, D. Panigrahi, and S. Dey,"Energy-Efficient Link Adaptations in IEEE 802.11b Wireless LAN", *Proceedings of IASTED International Conference on Wireless and Optical Communications*, pp. 578-583, Banff, Alberta, July 2003). Chapter 3 is based on material published in the International Symposium on Wireless Personal Multimedia Communications in 2003, (N. Ramos, D. Panigrahi, and S. Dey, "ChaPLeT: Channel-dependent Packet Level Tuning for Service Differentiation in IEEE 802.11e", *Proceedings of International Symposium on Wireless Personal Multimedia Communications*, vol. 3, pp. 86-90, Yokosuka, Kanagawa, Japan, October 2003) and material published in the IEEE Networks Magazine in 2005 (N. Ramos, D. Panigrahi, and S. Dey, "Quality of Service Provisioning in 802.11e Networks: Challenges, Approaches, and Future Directions", *IEEE Networks Magazine*, vol. 19, no. 4, pp. 14-20, July/August 2005). Chapter 4 is based on material published in the ICST/IEEE Intl. Workshop on Broadband Wireless Multimedia in 2004 (N. Ramos, D. Panigrahi, and S. Dey, "Dynamic Adaptation Policies to Improve Quality of Service of Multimedia Applications in WLAN Networks", *Proceedings of ICST/IEEE Intl. Workshop on Broadband Wireless Multimedia*, San Jose, California, October 2004) and material published in the ACM Springer Wireless Networks Journal in 2007 (N. Ramos, D. Panigrahi, and S. Dey, "Dynamic Adaptation Policies to Improve Quality of Service of Real-Time Multimedia Applications in IEEE 802.11e WLAN Networks", *ACM Springer Wireless Networks Journal*, vol. 13, no. 4, pp. 511-535, August 2007). Chapter 5 is based on material published in IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications in 2007 (N. Ramos and S. Dey, "A Device and Network-Aware Scaling Framework for Efficient Delivery of Scalable Video over Wireless Networks", *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Athens, Greece, September 2007) and material submitted to the IEEE Transactions on Broadcasting (N. Ramos and S. Dey, "User

Device-Centric and Network-Cooperative Video Scaling Algorithms for Efficient High Quality Video Delivery", *IEEE Transactions on Broadcasting*). I was the primary researcher and author of each of the above publications, and the coauthors listed in these publications collaborated on, or supervised the research which forms the basis for these chapters.

VITA

2000	B.S., Computer Science and Engineering, University of Washington
2000–2007	Research Assistant, Electrical Engineering Department, University of California, San Diego
2002	Summer Research Intern, TRW / Northrop Grumman, Carson, California
2003	M.S., Electrical Engineering (Computer Engineering), University of California, San Diego
2007	Ph.D., Electrical Engineering (Computer Engineering), University of California, San Diego

PUBLICATIONS

N. Ramos and S. Dey, "User Device-Centric and Network-Cooperative Video Scaling Algorithms for Efficient High Quality Video Delivery", *IEEE Transactions on Broadcasting* (in review).

N. Ramos, D. Panigrahi, and S. Dey, "Dynamic Adaptation Policies to Improve Quality of Service of Real-Time Multimedia Applications in IEEE 802.11e WLAN Networks", *ACM Springer Wireless Networks Journal*, vol. 13, no. 4, pp. 511-535, August 2007.

N. Ramos, D. Panigrahi, and S. Dey, "Quality of Service Provisioning in 802.11e Networks: Challenges, Approaches, and Future Directions", *IEEE Networks Magazine*, vol. 19, no. 4, pp. 14-20, July/August 2005.

C.K. Toh, E.C. Lee, and N. Ramos, "Next-Generation Tactical Ad Hoc Mobile Wireless Networks", *Northop Grumman Technology Review Journal*, Volume 12-1, Spring Summer 2004.

N. Ramos and S. Dey, "A Device and Network-Aware Scaling Framework for Efficient Delivery of Scalable Video over Wireless Networks", *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Athens, Greece, September 2007.

N. Ramos, D. Panigrahi, and S. Dey, "Dynamic Adaptation Policies to Improve Quality of Service of Multimedia Applications in WLAN Networks", *Proceedings of ICST/IEEE Intl. Workshop on Broadband Wireless Multimedia*, San Jose, California, October 2004.

N. Ramos, D. Panigrahi, and S. Dey, "ChaPLeT: Channel-dependent Packet Level Tuning for Service Differentiation in IEEE 802.11e", *Proceedings of International Symposium on Wireless Personal Multimedia Communications*, vol. 3, pp. 86-90, Yokosuka, Kanagawa, Japan, October 2003.

N. Ramos, D. Panigrahi, and S. Dey,"Energy-Efficient Link Adaptations in IEEE 802.11b Wireless LAN", *Proceedings of IASTED International Conference on Wireless and Optical Communications*, pp. 578-583, Banff, Alberta, July 2003.

ABSTRACT OF THE DISSERTATION

Network Adaptation Techniques to Enhance Efficiency and Quality of Wireless Multimedia Transmissions

by

Naomi Ann Suba Ramos

Doctor of Philosophy in Electrical Engineering (Computer Engineering)

University of California, San Diego, 2007

Professor Sujit Dey, Chair

Driven by the improvements in wireless technology and proliferation of available content, wireless networks have become extremely popular. Having evolved significantly in recent years, wireless networks now serve as a gateway to a diverse set of rich applications and functionality. Applications range from standard Internet services to real-time services, such as multimedia video and voice over IP. Although wireless networks have become popular, there are many challenges that remain. In addition to the inherent problems of wireless connectivity, which include varying channel conditions, interference, and propagation loss, upcoming networks must support applications and devices that place a greater demand on network resources. Users will also have higher requirements for future wireless networks, expecting guaranteed problem-free wireless access and a better overall user experience.

This dissertation addresses several key problems in wireless networks: the need for improved resource efficiency, both in terms of energy and network resource allo-

cation, the lack of Quality of Service (QoS) support, and the inability of applications to scale based on channel or device limitations. This dissertation proposes network and application layer adaptation techniques to enhance resource efficiency and Quality of Service. It introduces the concept of dynamic adaptations at various layers of the network protocol stack and considers four protocol adaptation techniques. The first targets improving energy efficiency by dynamically changing link layer parameters. The second focuses on ensuring differentiated channel access by adapting related channel parameters. The third technique uses network scheduling to ensure appropriate network resource allocation and improved channel access. The final adaptation technique investigates scaling at the application layer to improve network resource efficiency and achieved application quality.

The presented experiments demonstrate that the developed network and application layer adaptation techniques can significantly improve the wireless user experience. By providing a more efficient utilization of energy and network resources, as well as providing QoS guarantees, the protocol adaptation schemes show remarkable improvement in several metrics compared to existing approaches. The adaptation techniques described in this dissertation will enable wireless networks to support higher quality applications for a greater and more diverse set of users.

Chapter 1

Introduction

The popularity of wireless networks has become quite apparent in recent years. Wireless users were once limited to a small subset of the population, with networks available only in limited areas and a lack of applications and content. However, driven by the improvements in wireless technology, lower costs and proliferation of available content, the number of wireless users has increased significantly. The projected number of wireless users will rise to 3.13 billion worldwide by 2010, with 270 million wireless users in the U.S. (80% of the U.S. population), according to TIA's 2007 Telecommunications Market Review and Forecast [1]. Additionally, wireless connectivity will become omni-present with the increase in the coverage and capacity of wireless wide area networks, local area networks, and the evolution of wireless broadband networks. For Wireless Local Area Networks (WLAN) alone, the number of wireless hot spots worldwide is projected to increase to 218,000 by 2010.

In addition to this increase in the actual number of wireless users and availability, there is a broadening in the functionality of wireless networks. There is a shift from using wireless networks for simple non-intensive applications to using these networks as a gateway to richer applications and functionality. This shift can be seen in the current diversity of devices connecting to these networks, as well as the type of applications in use. Wireless devices now range from cellular phones and Personal Data Assistants

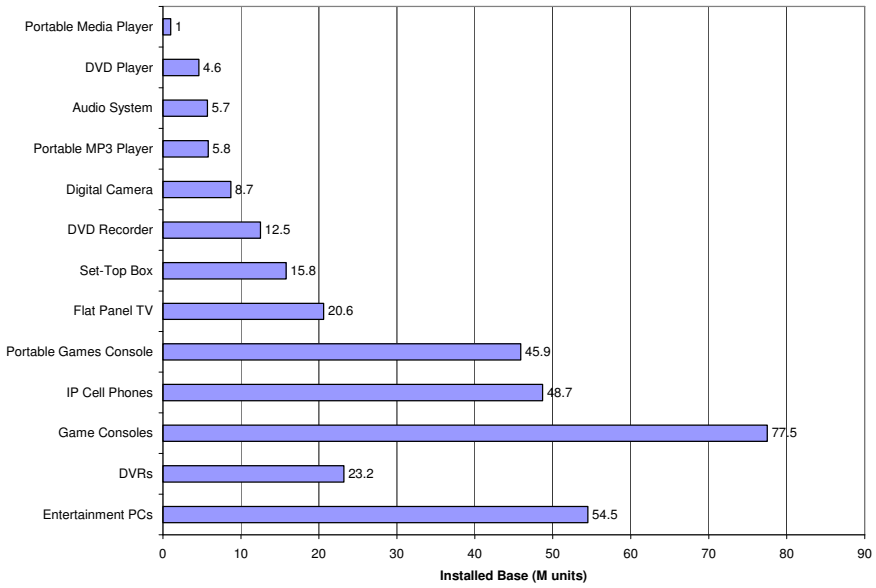


Figure 1.1: US Connected Home Forecast 2010

(PDAs) to televisions and appliances. In a typical home, the number of Internet Protocol (IP) entertainment devices, which include digital media servers and receivers, is projected to increase in sales to 324.5 million units per year [2]. Figure 1.1 shows the diversity of IP-enabled devices expected to be found in homes in 2010. It is expected that the majority of these devices will be wireless-enabled.

With the expected increase in the number of client devices, wireless applications are being developed for use in varied environments, including homes, offices, warehouses, hospitals, and other public-spaces. Hence, there is a diverse set of applications that are being used over wireless networks and these applications have very different requirements. Some of the applications can be considered background applications, such as email or file downloads, that have no strict requirements from the network. There are others, however, such as multimedia applications that are more challenging to support because of their associated real-time requirements. While wireless multimedia is more challenging, the expected growth of this set of applications can be seen in the projected revenue of wireless multimedia of $44 billion by 2010 [1].

The above trends are driving the research and development of new end-user devices, new applications, and new wireless technologies. With this type of explosive growth, it is necessary for wireless networks to be able to provide seamless support for this emerging market. However, in order to satisfy future wireless users, there are many challenges that need to be addressed. The first set of challenges is in terms of satisfying wireless user demands. Not only do wireless networks suffer from their inherent bottlenecks: varying channel conditions, interference, propagation loss, and multi-path effects, but they also must deal with a shift in the expectations of what wireless networks must provide. Prior to the popularity of wireless networks, users were generally patient and tolerant of connectivity loss, unreliability, and best-effort services of wireless networks, accepting these problems as unavoidable. However, users of today's wireless networks have little tolerance for errors and delays and place a greater demand on the available network resources. Mainly, users want guaranteed wireless connectivity and receive the highest quality content available given their device-type and current conditions. The second set of challenges is in ensuring efficient wireless systems, in both network and user resources, such as channel access time and energy resources. Network service providers, for example, want their users to not only be satisfied with their wireless access but they also want to support the maximum number of users in a given network. Additionally, ensuring efficiency at the user device is an important challenge. Users need to be able to utilize their device over the wireless network for extended periods and therefore, it is important to ensure battery and network efficiency over the wireless medium.

Having discussed the broader challenges, we next describe some of the specific difficulties faced by wireless networks and provide a brief overview of related research. Next, we introduce the approach taken by this thesis to address the above challenges. Mainly, we propose dynamic network adaptation at different network elements to improve the efficiency and quality of wireless multimedia transmissions. We follow with a section on the scope of the thesis. Finally, we summarize the contributions made by this thesis and provide an overview of the remaining chapters.

1.1 Challenges of Wireless Multimedia Networks

There are a number of key challenges in providing support for future wireless multimedia networks. These challenges include minimizing energy consumption, allocating the appropriate network resources and providing necessary quality of service support. Additionally, there is the challenge of achieving the best quality based on device and channel limitations. We next describe these challenges in greater detail and provide a brief overview of related research.

1.1.1 Energy Consumption

Energy consumption at the user device is one of the major challenges faced in wireless networks. The majority of wireless devices are mobile and must operate with limited energy resources. Hence, the device's lifetime is dependent on how efficiently resources are used, including minimizing the amount of energy consumed for communication. While improvements have been made in the past few years, communication costs still consume a significant portion of the energy budget for a device. For example, communication energy accounts for approximately 30% of the energy used in a typical laptop using Wireless LAN [3]. Energy consumption increases because of the radio or WLAN card itself and also by the power consumed in supporting the WLAN connection. The growth in wireless multimedia consumption will make energy a more critical concern. Previously, data transmission in wireless networks was mainly limited to sparse web viewing and emails. However, with the greater use of wireless devices to consume multimedia, applications will not only increase the amount of data transmitted over the wireless medium, but will also mean extended time periods of use. For example, streaming music, playing Internet games, watching online movies, and talking to colleagues using Voice over IP (VoIP) are all applications that call for extended periods of wireless use. The advantages of wireless connectivity, easy access and availability, will be greatly diminished if it becomes necessary to recharge devices or halt applications.

With user device energy consumption being recognized as a key problem in wireless access, research has focused on minimizing energy consumption. Areas of investigation include low-power circuit designs that optimize various components of the wireless radio to reduce power consumption. Notable examples of this area include [4], [5], [6], and [7]. In [5], the author provides an overview of the challenges of designing radio transceivers to minimize power. In [6], the authors describe an ultra-wideband transceiver architecture for low power and low rate systems. Another set of research is in terms of improvements in battery technology. This area includes profiling batteries and modeling their characteristics to improve system design. Notable examples of modeling are described in [8] and [9]. Dynamic power management and other battery management techniques have been incorporated by others such as [10] and [11].

In addition to these techniques, there have been approaches that target the two main components of communication energy: idle energy and transmission energy. Considerable research has been in identifying when the device is idle and putting the device in a sleep state where certain components of the radio are shut down to reduce energy consumption. Some examples of this research include [12], [13], [14], and [15]. In [16], a client-side prediction scheme of multimedia data is used to determine the appropriate sleep schedule for the user. Another research area investigates techniques for reducing the amount of energy used during data transmissions. One example is to develop methods of reducing the data sent over the wireless medium. In sensor networks, for example, research has been done to avoid transmitting data unnecessarily by performing data aggregation at certain points in the network. Examples of these techniques are [17], [18], [19], and [20]. Developing improved routing techniques to reduce energy in ad-hoc networks is also another research direction. These techniques have been discussed in [21], [22], [23], and [24], generating minimal energy routes through various wireless network topologies. Another method of reducing transmission energy is to dynamically adjust the transmission power. The research in [25], [26], [27], and [28] describe transmission power adaptation techniques for various wireless networks.

1.1.2 Network Resource Efficiency

In addition to the challenge of energy, another important concern is ensuring efficient network resource allocation. Since the wireless channel is a shared medium, we specifically look at network resources in terms of channel utilization. Although there have been improvements in wireless technology to increase the capacity of wireless networks, the number of wireless users and amount of data transmitted continue to increase, as described in the projections earlier in this chapter. With this in mind, it is essential that resources be allocated appropriately to ensure that the network can support the requirements of richer applications and a greater number of users.

There have been several techniques that have been introduced to increase the capacity of wireless networks. A number of the techniques have been focused at the physical layer and work towards improving spatial reuse and reducing retransmissions by minimizing the number of wireless errors. Spatial reuse research increases the number of users that are able to use the wireless medium at the same time. Some examples of spatial reuse research include [29], [30], [31], [30], and [32]. This area is generally focused on creating non-interfering channels through different means, such as dynamic carrier sensing, dynamic channel assignment, coding, frequency, diversity and even multiple wireless access. An example of recent research has been to improve spatial reuse by combining Multiple-Input Multiple-Output (MIMO) antennas with orthogonal frequency division multiplexing (OFDM) for improved capacity. This research area focuses on achieving spatial reuse through the use of multiple antennas and through the use of orthogonal (non-interfering) frequencies, as shown in [33], [34], and [35]. Another physical layer technique to improve network efficiency is to reduce the number of channel errors. By reducing the number of unsuccessful transmissions, the overall capacity and network goodput increases. One method of reducing errors is to choose an appropriate modulation scheme and error correction technique in real-time based on the monitored channel. Examples can be found in [36] and [37], where they improve the chance that a packet will be correctly received by the intended user. In addition

to these physical layer techniques, network resource efficiency can also be improved through other means. Other techniques include determining optimal routing in the case of ad-hoc networks, as described in [38], [39], and [40] and admission control policies for various networks, as described in [41], [42], and [43].

1.1.3 Quality of Service

Beyond the need for developing techniques to extend a device's lifetime and supporting a greater number of devices in wireless networks, another important challenge that needs to be addressed is quality of service. Wireless users not only want access to the network, but also want to have their applications work seamlessly over the wireless medium. In order to do this, application requirements, such as delay tolerance, jitter, and throughput, have to be met by the network. Application requirements can vary significantly, ranging from real-time applications, such as VoIP and video, to background traffic, such as email or downloads. Even the same type of applications can have different requirements. For example, two video streams that have been encoded with different parameters will have significantly different properties. Additionally, application requirements can fluctuate dramatically over time if the application has been Variable Bit Rate (VBR) encoded.

A major component of satisfying different quality requirements is understanding the needs of the user applications. Research has been conducted on profiling various applications and modeling their time-varying requirements. Examples of this type of research investigation include [44], [45], [46], and [47]. One method of abstracting these requirements is to categorize applications into various groups and provide a means of prioritizing one group over another. The grouping can be based strictly on application type or some other attribute, including pay structure. This means of prioritization can be found in different levels of the protocol stack. For example, with Differentiated Services (DiffServ) [48], priority levels are used in routing packets through the Internet with the goal of reducing end-to-end delay. This concept of grouping into different priority levels

can also be found in lower layers, such as the medium access layer. The recently adopted IEEE 802.11e standard for Quality of Service in Wireless LAN [49] is one example of a medium access layer that defines different access categories and gives prioritized access to the wireless shared medium. Similar techniques can be seen in cellular networks, such as [50] and [51].

Another means of supporting quality of service is to take the requirements of the application and use this information directly to reserve some of the network resources. These systems use a defined traffic specification that often contains information about the minimum, average, and maximum requirements of the applications. In comparison to Diffserv, Integrated Services (Intserv) [52] is the fine-grained approach for routing. Intserv uses traffic specifications to achieve the appropriate channel access in routing based on their requirements. The reservation-based approach can also be seen in the lower layers. The IEEE 802.11e standard [49] has made provisions for reservation-based quality of service, which uses traffic specifications to allocate a fixed period of channel access in a given period. There has also been research, such [53] and [54], that describes bandwidth reservation policies in the context of cellular networks.

In addition to these above schemes, quality of service using reservations and categorized prioritization has been evaluated in the context of admission control schemes, such as [55], [56], and [57], and routing protocols as surveyed in [58] and discussed in [59] and [60] for various wireless networks.

1.1.4 Serving Diverse Platforms

The final challenge we discuss is how to serve multimedia content to a growing set of wireless devices with significantly diverse capabilities. Not only is the range of traditional wireless devices, such as cellular phones and PDAs, widening, but also appliances, such as televisions and game consoles, are being designed to support wireless connectivity. Given the increasing diversity in wireless devices, the same application must be scaled based on the limitations of the devices, as well as the network or channel

limitations. For example, if we consider a single video stream, a user will receive different experiences depending on whether it is being viewed on a laptop in comparison to a small cellphone. Another factor for content adaptation can be the ability for the network to handle the needed application requirements. If channel conditions are varying or if the network does not have additional resources to allocate, then content adaptation may be the only means of providing satisfactory application quality.

There are a few existing techniques that are used to adapt content based on user device capabilities. One method is to prepare and store various versions of the content at the server. In this method, user devices are served the most suitable version of video stream at the time of their request [61]. The alternate approach is to use transcoding techniques, such as those discussed in [62], [63], and [64], to decode and re-encode data based on the user request. Although these techniques are used today, the growth in wireless multimedia will lead to higher storage and processing costs. The transcoding and encode and store methods will become unmanageable with the greater use of multimedia. In recent years, there has been research on developing encoding schemes that provide a means of bit-stream scalability. This reduces the load on the content server by providing a means of providing different quality levels of the same content. The extent of scalability of the different video standards vary but some scalability can be found in H.264 [65], MPEG4 [66], and the upcoming Scalable Video Codec standard [67].

1.2 Thesis Approach

In this thesis, we address the key challenges of energy consumption, network resource efficiency, quality of service, and diverse user devices. We approach these challenges by looking at a typical wireless network and evaluating what can be done at different elements of the network. Figure 1.2 illustrates a typical wireless network consisting of a content source, an Internet connection, a network access point, and a set of user devices with different requirements and capabilities. While previous research has mainly focused on making changes at a single network element, we propose dynamic

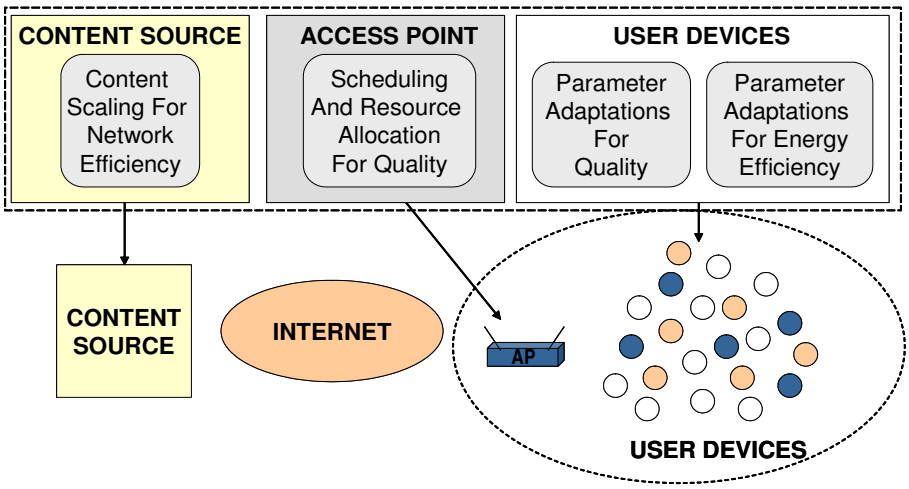


Figure 1.2: Thesis Summary

adaptations within multiple network elements to more effectively address the mentioned challenges. Figure 1.2 provides a summary of the developed dynamic adaptations developed in this thesis.

This research recognizes that each network element, whether it is a user device, access point, or the content source, has a certain scope of control and insight into the entire system. In this thesis, we develop adaptation schemes based on the role of the network element and the potential impact of the adaptation scheme. Our proposed schemes take advantage of information available to these elements and directly use monitored information from the network, devices, and applications to determine the appropriate dynamic adaptation.

As seen in Figure 1.2, we first investigate what can be achieved at the user device. User devices in wireless networks have direct control of their own channel access and transmissions. They are also aware of their local channel conditions and internal energy resources. Understanding these characteristics, we propose two dynamic adaptations schemes for user devices that tackle the challenges of energy and quality of service. The developed adaptation techniques target the ability of the user device to change channel access and transmission based on monitored information about current channel conditions. To optimize energy usage, we identify and develop adaptation schemes that focus

on reducing energy inefficiencies caused by wasteful and unsuccessful transmissions. In our quality of service adaptation techniques, user devices recognize when they are suffering a degradation in quality and make changes in their channel access and transmission parameters to achieve their expected level of service.

Next, we look at the network coordinator or, more specifically, at the access point in the case of our example network. The access point can interact with all the devices in the network and is in control of channel allocation when a reservation-based system is used. Additionally, an access point is in charge of any network management functionality in the wireless network. Given these characteristics of a typical access point, we use the access point's ability to monitor all the users in the network, as well as its ability to schedule and allocate resources to address the challenge of quality of service and bandwidth efficiency. The adaptation scheme improves quality of service and efficiency by evaluating the current user needs based on their monitored application traffic load and determining the appropriate type and duration of channel access given network limitations. The adaptation makes the necessary changes in terms of scheduling to provide the suitable resources for each device.

The final network element that this thesis evaluates is the content source. This network element usually stores and serves suitable content for a user with a download or streaming request. The content source may have some visibility into the intended user devices and their networks. With control of the served content and abstracted network and device information, this network element can be used to address the challenges of network resource efficiency and serving diverse platforms. In this thesis, we evaluate scaling algorithms based on user device and network information in order to reduce wasteful transmissions, improve overall quality of the network, and ensure that each user device receives a suitable video stream.

In summary, this thesis proposes dynamic adaptations at different network elements to address the challenges described in the previous section. By targeting different network elements and taking advantage of the element's visibility and role in the network, we can better improve the quality and efficiency of wireless networks.

1.3 Scope of Developed Network Adaptation

Having briefly described our approach, we now describe the scope of our developed network adaptations before providing the thesis overview. The majority of the proposed research in this thesis can be adapted to work in other similar types of wireless networks. These include different physical layer technologies and wireless channel access modes.

Although the concepts can be applied to diverse wireless networks, we have investigated the proposed techniques in the context of Wireless LAN IEEE 802.11-based networks. We focus on these networks because of their popularity and their wide use in accessing wireless multimedia. The simulation experiments are conducted in the context of infrastructure mode networks. However, all the user device adaptations can be applied in ad-hoc IEEE 802.11 networks. The techniques developed in this thesis for energy and quality at the user devices and access point are adaptations at the data link layer because of this layer's ability to control access to the wireless medium and awareness of channel variations.

In terms of quality of service, the techniques we developed can be extended to other wireless networks with prioritized channel access and reservation based channel access. For the thesis, we showed the impact of our techniques in the context of IEEE 802.11 networks. Mainly, we looked into the developments made in the IEEE 802.11 standard for Quality of Service support. The IEEE 802.11e improvements were standardized and finalized in the same time frame of this dissertation. The techniques developed in this thesis identify limitations in both the distributed and point coordination channel access schemes described in the standard and determine the suitable scheduling and parameter adaptation to improve quality of service.

Finally, in terms of multimedia applications, this thesis limits the scope of content adaptation to multimedia video. Due to the real-time requirements and growing use of multimedia video over wireless networks, providing support for multimedia video can be extremely challenging. We look at Variable Bit Rate (VBR)-encoded video as

the targeted application for our quality of service scheduling techniques. For our content adaptation, we focus our attention on Scalable Video Coding (SVC), which is currently being standardized as an H.264 Advanced Video Codec (AVC) Extension, and we propose a scheme to scale video prior to delivery over wireless networks.

1.4 Thesis Contributions and Overview

We now describe the thesis contributions and provide an overview of the thesis. This thesis proposes network adaptation techniques to enhance resource efficiency and quality of service in wireless networks. It introduces the concept of dynamic adaptations at various network elements to provide higher quality content for wireless network users. This thesis addresses several key problems of wireless networks discussed above: the need for improved resource efficiency, both in terms of energy and resource allocation, the lack of quality of service support, and the inability of applications to scale based on channel or device limitations.

As described earlier, Figure 1.2 maps the developed techniques to a typical wireless system. We have investigated network adaptations at the user device, access point, and content source to improve the efficiency and quality of the wireless network. In each of these schemes, monitored information from the network, devices, and applications is used to determine the appropriate dynamic adaptation. The network adaptations were designed with the intention of working with the current standards and to avoid making changes to the current protocols being used for wireless access or video encoding.

The thesis first investigates what can be achieved at the user device with two adaptation techniques. The first is a network adaptation technique that dynamically changes link layer parameters to reduce energy usage in transmissions over wireless networks. The second looks at improving differentiated service and quality of service by changing channel access parameters. After looking at device adaptation, we next consider adaptation at the access point. With this third technique, we propose network scheduling and resource allocation based on application demands to provide a means

of satisfying a greater set of users, as well as provide higher quality content to users. Having focused on network adaptation and scheduling in the previous three techniques, the final adaptation technique looks at the potential of scaling of the content. The developed adaptation focuses on shaping multimedia video streams to efficiently use network resources, provide high quality video streams to users, and meet network service objectives despite device and network limitations.

The experiments and simulation studies demonstrate that the developed network and application layer adaptation techniques can significantly improve the user experience. By providing a more efficient utilization of energy and network resources, as well as improved quality of service, the protocol adaptation schemes show remarkable improvement in several metrics compared to existing approaches. The adaptation techniques described in this dissertation enable wireless networks to support high quality applications for a greater and more diverse set of wireless users.

The remainder of this thesis is organized as described below. Note that a more detailed description of research related to each of the proposed network adaptation techniques is provided in each of the chapters.

- In Chapter 2, we present a user device parameter adaptation technique that dynamically tunes three important link layer parameters based on experienced channel conditions to improve energy-efficiency.

- In Chapter 3, we look into improving quality of service and present a user device parameter adaptation technique to avoid starvation and priority inversion. The adaptation algorithm dynamically changes three link layer parameters based on application requirements and channel experience at the end-user device.

- In Chapter 4, we present a scheduling algorithm that allocates additional resources and determines the type of channel access based on the time-varying needs of the application to improve quality metrics and increase the number of users supported in the network.

- In Chapter 5, we describe a content adaptation technique to improve network efficiency and application quality. In particular, we describe a video scaling framework that uses information about the device, network limitations, and potential quality improvement to determine the characteristics of the video to be delivered to the end-user.

- Finally, in Chapter 6, we discuss future research directions that can be pursued based on the research presented in this thesis.

Chapter 2

User Device Parameter Adaptation to Improve Energy Efficiency

2.1 Introduction

Energy efficiency is one of the primary bottlenecks of accessing data over wireless networks from portable devices. The majority of mobile devices have limited available energy resources and the energy used for communication accounts for a significant portion of a wireless device's energy consumption. Despite improvements in battery capacity, the energy requirements of functionality-rich wireless access to the Internet and wireless multimedia will make communication energy an even bigger challenge in the future. To enable real mobile access to rich data, energy consumption should be reduced in order to extend the device lifetime effectively.

In this chapter, we specifically look at energy-efficient techniques in a Wireless LAN IEEE 802.11b network. With a forecast of increased concentration of Wireless LAN (WLAN) hotspots and WLAN-enabled appliances, the proposed adaptation impacts a large sector of wireless networking. Additionally, some of the proposed adaptation can be directly applied in other wireless networks without any changes.

In these networks, high application demand can impact the energy consumption

by increasing the communication load. In addition to application demand, another factor that significantly affects energy consumption with WLAN is the varying channel conditions. Channel interference caused by other devices operating in the same public radio spectrum (2.4 GHz) of the IEEE 802.11b network is a primary contributing factor to the channel variation. In addition, due to the typical indoor usage of WLAN, propagation loss and multi-path effects are also heavily present in these networks.

With these challenges in mind, there has been a wide range of work that has focused on minimizing energy in WLAN networks. Previous work varies from power-saving methods [68], which controls the device's sleeping routine, to routing/scheduling techniques in ad-hoc networks [69] [70]. While the other schemes target minimizing energy consumption during idle periods, in this chapter, we focus on minimizing the energy consumed during data transmission periods, and hence can be complimentary to other power-saving methods. In our approach, we focus on adapting parameters related to data transmission that have impact on energy consumption.

We target the data link layer and identify parameters that can be adapted to achieve energy minimization. We focus on the data link layer because of its ability to control access to the wireless medium and obtain information about current channel variations from the physical layer. As a first step, we identify and study the effects of dynamically adapting parameters available in the IEEE 802.11b protocol standard. In this chapter, we study the following parameters: *fragmentation threshold*, *transmission power*, and *retry limit*. In our choice of these parameters, we looked for parameters that can be adapted without affecting the backward compatibility with the existing standard.

With the goal of reducing energy consumption, we design run-time adaptation policies that appropriately set the values of these parameters for energy minimization. For each of these policies, we select a technique to monitor channel conditions, determine when to invoke the adaptation, and use feedback mechanisms to further improve the policies. Additionally, we investigate the issues in combining multiple policies and develop multi-parameter adaptation policies. Using an Opnet-based simulation framework, we demonstrate that significant savings can be achieved by enabling energy-

efficient adaptations at the IEEE 802.11b Data Link Layer. Note that the proposed techniques can be applied to other IEEE 802.11-based standards, including IEEE 802.11a and 802.11g.

While previous work has been done in adapting some of these parameters, our work improves upon these techniques by enabling dynamic adaptations using alternative channel estimation techniques as well as passive channel feedback. For example, although the fragmentation threshold parameter was investigated in [71], [72], and [36], the older work focuses on only a static setting of this parameter and the more recent work is limited to using downlink measurements to estimate the uplink channel conditions. In addition to proposing a closed-loop formula to find the optimal fragmentation threshold, we design a policy that uses the experienced retransmission in the uplink direction as a feedback mechanism to optimally set the fragmentation threshold.

Similarly, with respects to transmission power, there has been recent work done on determining the appropriate setting power level for wireless networks [73], [74], [75], [76]. The above efforts range from power control loops using a signaling system in ad-hoc networks to the use of inter-packet power level fluctuations. In this chapter, we present a simple adaptation policy that sets the transmission power level per packet to be effective in very extreme channel conditions.

Finally, in terms of the retry limit, work has been done to evaluate the effects of different link level error control mechanisms such as ARQ in terms of energy and throughput [77]. However, to the best of our knowledge, there are no adaptation policies that use retry limit to minimize energy consumption in IEEE 802.11b networks.

Having briefly introduced our work, the rest of the chapter presents a more detailed description of our adaptation techniques and is organized as follows. We present our proposed adaptation policies in Section 2.2. For each of the parameters, we first present an analysis of the effects on energy consumption, and follow with a description of our adaptation policies. Experimental results follow next in Section 2.3, demonstrating the effectiveness of our adaptation methodology. Finally, in Section 2.4, we conclude and summarize the chapter.

2.2 Adaptable Parameters and Effects

In this subsection, we present different adaptable parameters and describe their effects on energy constraints. The parameters we target are *fragmentation threshold*, *transmission power*, and *retry limit*. Our goal is to reduce energy consumption by finding the appropriate value of these parameters considering the current channel conditions.

2.2.1 Fragmentation Threshold

IEEE 802.11b defines a mechanism to partition network level data packets into smaller units. Using packet size as a criterion, packets larger than a specified *fragmentation threshold* are partitioned in order to respond to dynamic erroneous channel conditions. Fragmentation improves the packet transmission reliability as the probability of a successful transmission increases with the decrease in packet size. However, partitioning of large data packets into smaller fragments can lead to additional data transmission overhead. Once a packet is partitioned, each fragment is augmented with individual headers, sent as an independent transmission, and acknowledged individually. Hence, under good channel conditions, fragmentation can lead to suboptimal performance. Having described the fragmentation threshold, we next analyze the effect on energy consumption.

Analysis

In order to understand the effects of fragmentation, we present an analysis of the energy consumed given a specified fragmentation threshold under current channel conditions. Given that n_{frag} is the number of fragments after a packet has been partitioned, a_{ret} is the average number of attempts before a successful fragment transmission, and e_{frag} is the energy consumed for each fragment, the total energy (E) consumed by transmitting a packet can be written as

$$E = n_{frag} * a_{ret} * e_{frag} \tag{2.1}$$

Given a fragmentation threshold of *F* (bits), the probability of a fragment not in error is given by $p = (1 - BER)^F$, where *BER* is the bit error probability. Note that, a packet is considered erroneous in this analysis when one or more bits are in error. With this probability, we calculate the expected number of attempts required a_{ret} to be $1/(1 - BER)^F$.

Given that *X* represents the upper layer packet size in bits, we can replace the number of fragments with n_{frag} = *X/F*. Furthermore, the energy consumed by each fragment can be expressed as e_{frag} = $(P_T * (F + H))/R$, with P_T as transmission power (W), R as data rate (bits/sec), and H as header size (bits). Hence, total energy consumption to send a packet of size X can be expressed as

$$E = \frac{X}{F} * \frac{1}{(1 - BER)^F} * \frac{P_T}{R} * (F + H) \tag{2.2}$$

In order to find the optimum fragmentation threshold, we compute the gradient (dE/dF) and equate it to zero. For very small values of BER, the optimum fragmentation threshold can be calculated as follows.

$$F_{opt\_energy} = -\frac{H}{2} + \sqrt{\frac{H}{BER}} \tag{2.3}$$

Adaptation Policies

Having presented the optimum fragmentation threshold value given the current BER, we present two fragmentation threshold adaptation policies: *Dynamically Varied Open Loop (DVOL)* and *Dynamically Varied Closed Loop (DVCL)*.

DVOL uses downlink channel measurements to estimate uplink channel conditions. It measures the signal strength, and estimates the BER of the received traffic from the access point. It uses the BER information to set the fragmentation threshold for the uplink traffic. We call this policy *Open Loop* since the decisions are made based on the

observed signal strength, and do not use any form of active feedback on the uplink. Note that, the measured BER information is used to estimate the channel conditions in the uplink assuming that the communication channels are symmetric in IEEE 802.11b based networks. It can be noted that the success of the policy is dependent on the availability of downlink traffic, which may not be available in particular applications. Additionally, it assumes that receiver and transmitter experience the same interference.

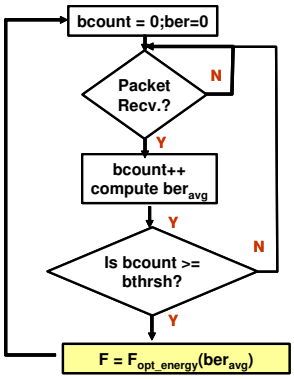


Figure 2.1: Fragmentation Adaptation Policies : Dynamically Varied Open Loop (DVOL)

To counteract the heavy dependence on downlink traffic, as well as consider a possible difference in experienced interference at the receiver and the access point, *DVCL* uses packet retransmission information to estimate the uplink channel condition. Using the relationship between the packet error probability and the BER value, we estimate the current BER with the following Equation 2.4. We use the measured retransmission ratio as an indication of packet error probability. In this equation, fragment and header sizes are represented by F and H respectively. We call this policy *Closed Loop* as it uses both measured information and retransmission information on the uplink channel as an active feedback.

$$BER_{est} = \frac{Retransmission\ Ratio}{(F+H)} \tag{2.4}$$

The detailed flowcharts of DVOL and DVCL adaptation policies are shown in Figure 2.1 and Figure 2.2 respectively. In the DVOL policy, with every received packet,

the average BER of the channel is updated. A basic counter is used to control the frequency of the fragmentation adaptation. Given that the BER is averaged for a sufficient number of packets, the fragmentation threshold is set to a final value, as given by Equation 2.3.

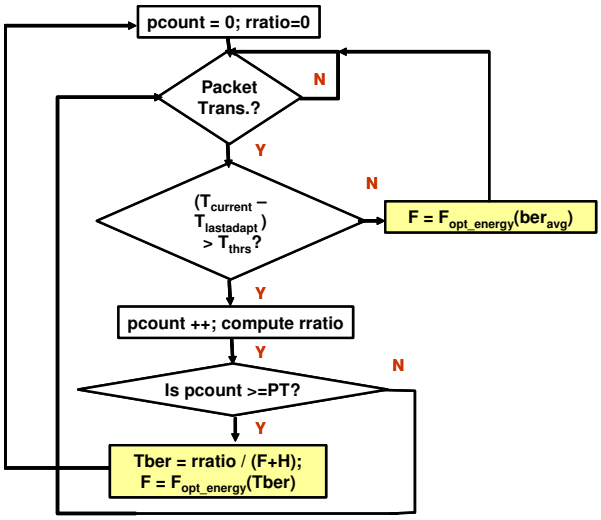


Figure 2.2: Fragmentation Adaptation Policies : Dynamically Varied Closed Loop (DVCL)

For the DVCL adaptation, the main goal is to use retransmission information to determine the current channel condition. If the downlink measurements of the retransmission ratio are insufficient, the DVCL policy defaults to DVOL and uses the averaged BER value to determine the fragmentation threshold. However, if the uplink information is sufficient, DVCL estimates the current BER of the channel using the retransmission ratio with Equation 2.4, and then uses this value of BER in Equation 2.3 to determine the fragmentation threshold. Next, we present the effect of another parameter that can be controlled by the link layer *i.e. transmission power*

2.2.2 Transmission Power

In addition to selecting the appropriate packet size, another parameter that affects the probability of successful transmission of packets is the *transmission power*. We next

present a brief description on the effect of transmission power on energy consumption.

Analysis

The experienced BER value at the receiver end provides information about the channel conditions and can be used in energy saving techniques at the MAC level. The BER heavily depends on the signal strength at the receiver, which can be influenced by many factors including the transmission power level, propagation loss, antenna gain, and processing gain. Since the received signal strength is directly proportional to the current transmission power level, the transmission power level can be adjusted to change the signal strength and hence the BER at the receiver.

In terms of energy consumption, the effect of transmission power on energy consumption has been measured and modeled in previous work. In [78], the results indicate a strong dependence between the energy consumption and transmission power and this relationship can be modeled using a linear model.

With the above-mentioned effects, a tradeoff can be seen between the BER experienced and the energy consumption based on the transmission power level selection. Under good channel conditions, the transmission power level can be reduced to save energy consumption. However, under bad channel conditions, in addition to fragmentation, an increase in transmission power will reduce the percentage of retransmission and dropped packets, thus improve energy per goodput byte sent.

Adaptation Policies

With an understanding of the transmission power level parameter, we design an adaptation policy that responds to very extreme channel conditions. The transmission power adaptation algorithm increases the transmission power when the channel conditions are poor, as indicated by a significant amount of dropped packets or high BER. On the other hand, when the channel conditions are good, the transmission power is decreased to conserve energy.

One possible concern in adapting transmission power level is that it may lead to an increase in hidden node scenarios when nodes using different power levels may be unable to hear each other properly. In our adaptation policy, we address the concern by using high transmission power levels for control packets, such as RTS, CTS, and ACK, to minimize the chances of hidden nodes.

2.2.3 Retry Limit

In addition to fragmentation threshold and transmission power, the IEEE 802.11 Data Link Layer uses an automatic repeat-request (ARQ) based error recovery mechanism in order to tolerate interference and collisions. This generally involves retransmission of frames after a timeout period if no acknowledgment is received from the destination station. To regulate the number of retransmissions, the Data Link Layer defines a parameter named *retry limit*, the maximum number of retransmissions allowed. The station discards the frame after transmitting the packet for the maximum number of times allowed.

It is important to note that the number of retransmissions needed to successfully send a packet depends on current channel conditions and application requirements. Although an increase in the allowable number of retransmissions can increase the probability of a successful transmission in bad channel conditions, such an increase leads to greater energy consumption. Additionally, the retry limit can be used to control the average number of packets dropped. Under bad channel conditions, this parameter can be useful when applications are able to tolerate a certain percentage of dropped packets.

Analysis

Next, we present the effect of the *retry limit* on energy consumption and packet drop rate given current channel conditions. The probability of packet of size P received correctly with bit error rate of BER is $(1 - BER)^P$. Let us denote the probability of successful packet transmission as p and the probability of erroneous transmission as

$q = 1 - p$. Assuming the maximum retransmissions allowed is RL, the drop packet probability (dp_{prob}) after RL attempts is given by

$$dp_{prob} = q^{RL} \tag{2.5}$$

If the target packet drop rate is P_{drop}, we would want to insure that $q^{RL} \leq P_{drop}$. Solving for RL, we obtain the following relationship

$$RL \quad \geq \quad \frac{log(P_{drop})}{log(1-(1-BER)^P)} \tag{2.6}$$

Let us call the right side of Equation 2.6 to be R_{exp}, the expected number of retransmissions given the BER and tolerable drop rate. In terms of energy consumption, if the expected number of retransmissions, R_{exp} is greater than the current retry limit for a specific packet, we can expect the packet to be dropped. The retransmissions of this packet contribute to wasteful energy consumption without increasing the goodput of the system. Hence, under drastic channel conditions, which require a large amount of retransmissions, packets can be dropped early to conserve energy. In addition to dealing with drastic conditions, retry limits can be related to the application requirements. For example, with real-time applications and strict latency requirements, packet retransmissions may be unnecessary and hence, the retry limit can be set to lower values.

Adaptation Policy

Based on the above observations, we develop an adaptation policy to select the appropriate *retry limit* (RL) value depending on current conditions and requirements. In this policy, the expected number of retransmissions, R_{exp} is calculated using Equation 2.6 and compared to the current retry limit. If it is clear that the packet will be dropped by this estimate, we set the retry limit to a small value in order to avoid excess retransmissions. If this is not the case, we set the retry limit to either the default values or the expected number of retransmissions calculated at the earlier step, depending on which has the smaller value.

2.3 Experimental Results

In this subsection, we present the experimental evaluation of our adaptation policies. We first briefly describe the experimental setup. Next, we describe the effects of our adaptation polices on the link layer as described in the earlier sections. Finally, we present the performance of combining multiple policies on energy consumption.

2.3.1 Setup and System Configuration

For our experiments, we use the *Opnet* simulation environment, which provides wireless models of the DCF Data Link Layer and Direct Sequence Spread Spectrum (DSSS) physical layer of the IEEE 802.11b standard. The *Opnet* tool uses a pipeline model to determine channel variation and evaluates characteristics such as propagation loss and interference. For our experiments, we consider an IEEE 802.11b infrastructure-based network and evaluate a number of test cases with varying number of nodes and mobility patterns. We constructed a network scenario where the wireless clients connect to the remote data server through an IP backbone infrastructure. For estimating energy consumption, we used the energy models proposed in [79] and [80] that were developed through measurements of IEEE 802.11b interfaces with different transmission power.

2.3.2 Adaptation Policies

We now discuss the results of running the adaptation policies described in earlier sections and describe the effects of the following parameters: *fragmentation threshold*, *transmission power*, and *retry limit*. For these sets of experiments, we created and simulated a network with nodes running the *FTP* application.

Fragmentation Threshold

To determine the effects of the different fragmentation adaptation policies on

energy and throughput, we implemented both the DVOL and DVCL policies presented in Section 2.2.1 and integrated these mechanisms in the *Opnet* model. Due to sensitivity of the fragmentation threshold policy to channel variations, the simulation considered three different channel models: *fixed*, *Gilbert-Elliot Model*, and the *Opnet Generated*.

For the *fixed* cases (FIX1 & FIX2), the channel models were set to the values representative of bad and good conditions, i.e. BER values set to 5*10^{-4} and 10^{-6} respectively. Using the *Gilbert-Elliot* model, we simulated four test cases (GE1 to GE4). The Gilbert-Elliot model, as described in [81] and [82], generates BER based on a two state Markov chain model. We varied the transition probabilities of the model, similar to the values presented in [81], using combinations of 0.99 and 0.95. These probabilities alter the percentage of time spent at each state and the frequency of transition. The probabilities values selected for the models are as follows: GE1 (Pgg=0.99, Pbb=0.99), GE2 (0.95, 0.99), GE3 (0.99, 0.95), and GE4 (0.95, 0.95). Finally, we also evaluated a single mobile node experiencing the channel variation as given by *Opnet Generated* pipeline model (MOB) and a case with multiple nodes (MUL).

The criteria used for comparing these scenarios were *Energy per Goodput* and *Link Level Throughput*. For each of the above test cases, we compared the fragmentation adaptation policies, DVOL and DVCL, with a test case using a static fragmentation threshold of 800 bytes, as suggested by [71]. Table 2.1 and Table 2.2 present the results of the evaluation comparison.

The percentage savings recorded in the table indicate the percentage improvement between the adaptive policies and the static case. The results indicate that the adaptation policies of DVOL and DVCL can improve energy and throughput. On average, we see an average savings of 17% for DVOL and 19% for DVCL in terms of Energy per Goodput and see an average increase in DVOL of 13% and 19% in DVCL in terms of throughput. We note that the improvement is more significant in the situations when the channel is experiencing high BER, such as in FIX2, GE2, and MOB. In addition, the results indicate the DVCL policy fairs better than DVOL when there are frequent transitions of the channel or downlink prediction is inaccurate, such as in GE4

Table 2.1: Effect of Fragmentation Adaptation Policies on Energy per Goodput

Energy per Goodput (mAs/bit)					
	Static	DVOL	% Savings	DVCL	% Savings
FIX1	3.65E-05	3.42E-05	6.30%	3.41E-05	6.58%
FIX2	1.20E-03	1.10E-04	90.83%	1.40E-04	88.33%
GE1	5.90E-05	5.76E-05	2.37%	5.75E-0	52.54%
GE2	8.80E-05	7.81E-05	11.25%	7.90E-05	10.23%
GE3	4.11E-05	4.05E-05	1.46%	4.04E-05	1.70%
GE4	5.38E-05	5.0E-05	3.35%	5.08E-05	5.58%
MOB	4.83E-05	4.06E-05	15.94%	4.05E-05	16.77%
MUL1	1.13E-05	1.06E-05	5.60%	9.10E-06	19.11%

and MUL.

Table 2.2: Effect of Fragmentation Adaptation Policies on Throughput

Link Layer Throughput (Mbit/sec)					
	Static	DVOL	% Savings	DVCL	% Savings
FIX1	6.31	6.89	9.19%	6.90	9.35%
FIX2	0.44	0.62	40.68%	0.58	32.27%
GE1	5.30	5.90	11.32%	6.10	15.09%
GE2	3.69	4.26	15.45%	4.18	13.28%
GE3	6.09	6.57	7.99%	6.50	6.85%
GE4	5.50	5.60	1.82%	6.10	10.91%
MOB	5.66	6.87	21.41%	7.00	23.67%
MUL1	1.10	1.12	1.52%	1.57	42.29%

Transmission Power

In addition to the fragmentation policy, we also evaluated the effects of our transmission power adaptation. Using the Opnet pipeline model, we considered multiple test cases with varying fragment sizes and number of nodes in the network. In Table 2.3, we summarize our simulation results. We see that we can improve Energy per Goodput by 5-13% with a slight decrease in throughput.

Retry Limit

Table 2.3: Effect of Transmission Power Adaptation

Description		Energy per Goodput (mAs/bit)		
Number of Nodes	FragThresh	Without adapt	With adapt	% Savings
4	512	8.488-05	6.64E-05	21.82%
	None	9.54E-05	5.71E-05	40.16%
2	512	7.04E-05	5.49E-05	22.02%
	None	8.93E-05	4.55E-05	49.10%
1	512	4.75E-05	3.90E-05	17.89%
	None	5.10E-05	3.80E-05	25.49%

For the retry limit, we implemented our adaptation algorithm in order to determine the effect in Energy per Goodput and throughput. Our initial experiment was to determine whether the expected number of retransmissions, as given by Equation 2.6, was accurate in comparison to the actual retransmissions required. We considered a case without fragmentation and found that the expected number of retransmissions given by our formula is 80-90% accurate. Next, we created a scenario using FTP traffic with UDP as the transport layer (so as to allow for packet drops at the transport layer) and consisting of three different nodes experiencing varying BER. The target P_{drop} for this experiment was assumed to be 10%. We simulated with and without adaptation and obtained the results summarized in Table 2.4 and Table 2.5.

Table 2.4: Effect of Retry Limit Adaptation on Energy

Node	Average BER	Energy per Goodput (mA/s)		
		Without Adapt	With Adapt	% Savings
1	3.00E-04	2.32E-04	1.59E-04	31.47%
2	1.00E-05	4.79E-04	3.68E-04	23.17%
3	7.50E-06	1.16E-05	1.10E-05	5.17%

As seen in these tables, the adaptation leads to a 31% improvement in Energy per Goodput for Node 1 with only an increase of 1% dropped packets. With Node 2 and 3, we obtain an improvement of 23% and 5%, respectively, but suffer from a slight increase

Table 2.5: Effect of Retry Limit Adaptation on Drop and Retransmission Percentage

Node	Drop Percentage			Retransmission Percentage		
	Without Adapt	Witht Adapt	% Inc	Without Adapt	With Adapt	% Dec
1	20	21.00	1%	33	25	8%
2	7.5	11	3.5%	25	18	7%
3	1	4	3%	7	9	-2%

in dropped packets. This increase may be tolerable depending on the application.

2.3.3 Multiple Policies

Table 2.6: Effect of Multiple Policies

Policies	Energy per Goodput (mAs/bit)	% Savings
None	1.09E-04	0%
Frag	6.91E-05	36.75%
Retry	9.08E-05	16.93%
Tx	7.80E-05	28.60%
Frag-Tx	6.17E-05	43.52%
Frag-Retry	6.58E-05	39.77%
Tx-Retry	7.79E-05	28.72%
All	6.13E-05	43.89%

Having looked at the individual parameters and their adaptations, we evaluated the effects of using multiple policies. In addition to activating multiple policies at once, we also considered the dependencies and coordination needed between the policies. For example, if both the fragmentation and retransmission policy were activated, the fragmentation policy was scheduled to run first due to the retransmission policy's dependence on fragment length. In addition, when the transmission power and fragmentation policy were activated, the use of the transmission power policy was only activated when the fragmentation policy was exhausted and could not improve the situation.

In order to understand the effects of these policies, we evaluated the effect of different combination of adaptation policies on a network of four nodes in terms of Energy

per Goodput. Table 5 summarizes the results averaged over the nodes. It can be seen from the table that transmission power adaptation policy is more effective compared to any other individual policy. Also, it is clear that energy savings can be improved by combining policies.

2.4 Conclusion

In this chapter, we presented the effects of different adaptable parameters in the IEEE 802.11b Data Link Layer. Additionally, we introduced our runtime adaptation policies and showed that it can be beneficial in terms of energy savings to use these parameter adaptations. We discussed the challenges of combining the policies and showed that with the appropriate selection of policies, we were able to further improve our energy savings. Though our experimental results are based on an IEEE 802.11b model, we believe the presented techniques would be valid for other IEEE 802.11-based standards, including IEEE 802.11a and 802.11g, as well.

The text of this chapter, in part, is based on material that has been published in the IASTED International Conference on Wireless and Optical Communication in 2003 (N. Ramos, D. Panigrahi, and S. Dey,"Energy-Efficient Link Adaptations in IEEE 802.11b Wireless LAN", *Proceedings of IASTED International Conference on Wireless and Optical Communications*, pp. 578-583, Banff, Alberta, July 2003). The dissertation author was the primary researcher and author, and the coauthors listed in these publications collaborated on, or supervised the research that forms the basis for this chapter.

Chapter 3

Dynamic User Device Parameter Adaptation for Enhanced Quality of Service

3.1 Introduction

In our previous chapter, we presented network adaptation techniques at the user data device to improve energy efficiency. We focus in this chapter on the challenge of ensuring quality of service and investigate the potential of dynamic parameter adaptation at the user device to address this challenge. With recent improvements made in IEEE physical layer to support high data rates, applications with real-time requirements, such as video and voice applications, will be more commonly used in these networks. However, the current IEEE 802.11's medium access control (MAC) protocol [83] can only provide best-effort service to all applications. Without the ability to provide service differentiation, IEEE 802.11 networks will be inadequate and unable to satisfy the requirements of these applications.

Recognizing this challenge, the IEEE 802.11 Working Group has pursued standardization efforts to enhance the channel access mechanism of the IEEE 802.11 proto-

col, which has resulted in a new standard called IEEE 802.11e [49]. One of the enhancement mechanisms is the Enhanced Distributed Channel Access (EDCA), where service differentiation is achieved with the introduction of different access categories and the association of different contention parameters to prioritize access to the medium.

Although EDCA is able to provide a level of service differentiation, the mechanism does not consider the varying channel conditions present in WLAN networks. Channel conditions can be affected by problems such as propagation loss and multipath effects that are prominent in a typical indoor WLAN environment. In addition, channel interference caused by other devices operating in the same public radio spectrum can contribute to channel variation. Poor channel conditions can cause errors in transmitted packets and lead to retransmissions, causing delays, dropped packets, and throughput degradation. Another factor in WLAN access is the variation in network load. Since distributed contention-based medium access protocols use a shared channel access mechanism, the number of nodes and their respective traffic loads will affect the experienced channel access delay. A network experiencing high traffic load can make it difficult for a node to obtain access to the network, leading to high delays and low throughput.

Due to channel and network load variations, nodes in a WLAN network can experience significant throughput degradation. In addition, these effects may be amplified in quality-enhanced networks, such as the one proposed in IEEE 802.11e. In these networks, flows are differentiated by traffic classes, and flows with higher priority should normally experience higher throughput. However, the throughput degradation caused by channel and load variations can make it difficult to provide the desired service differentiation and may cause problems such as *priority inversion* and *starvation*, which will be later described in the chapter. Hence, depending on channel variability, the traffic class-based service differentiation mechanism alone may be insufficient to provide the desired QoS.

Since the potential negative impact of varying channel conditions and high network load can be significant, there is a need to recognize when such conditions are

present and provide techniques to improve the achieved throughput. The previous work in this area of research, which is described in detail in Section 3.5, has mainly looked at changing individual parameters in either the transmission period or medium access period to focus on separate problems of channel conditions, network load, or service differentiation.

In order to combat the effects of poor channel conditions, past techniques have mostly concentrated on increasing the probability of a successful packet transmission by adapting parameters such as packet size, transmission power, coding, etc. However, these techniques do not consider the effects of poor channel conditions on medium access delay. It is shown in this chapter that poor channel conditions can significantly affect the medium access delay, and thus cause throughput degradation. In the area of medium access parameter adaptation, techniques have been proposed to change medium access parameters to provide service differentiation, however they do not consider the effects of poor channel conditions.

To address the above limitations, we present an adaptation framework called Channel-dependent Packet Level Tuning (ChaPLeT), which dynamically changes link layer parameters based on channel information and experienced throughput. As opposed to past techniques, the ChaPLeT framework adapts multiple parameters, both related to packet transmission and medium access, to dynamically adapt to channel conditions and network load, and improve throughput and service differentiation.

Specifically, we concentrate on dynamically adapting three link layer parameters: *fragmentation threshold*, *persistence factor*, and *defer countdown*. We study the effect of these parameters on throughput under different network condition and present algorithms to dynamically adapt these parameters. To implement our adaptation techniques, we present an adaptation framework to set the parameter values based on current channel conditions, such as experienced bit-error-rate (BER), and application requirements, such as expected throughput. In these adaptations, we use the *fragmentation threshold* and *persistence factor* parameters to reduce the number of retransmissions and the medium access delay due to retransmissions. Additionally, we use the *defer*

countdown parameter to decrease the medium access delay caused by high network load. Finally, we discuss how the individual parameter adaptations can be applied in a quality-enhanced network to not only improve throughput, but also maintain desired service differentiation in the network.

To show the effectiveness of our approach, we implement our adaptation policy on top of the IEEE 802.11b medium access layer using the *Opnet* simulation framework [84]. We demonstrate that significant improvement can be achieved by enabling dynamic adaptation of channel access parameters. Additionally, we present the results of our adaptation policy when applied in a quality-enhanced network, based on IEEE 802.11e. We show that by improving the throughput of nodes experiencing high load and poor channel conditions, we are also able to improve the service differentiation experienced.

The rest of the chapter is organized as follows. We present background of Wireless LAN networks and describe the distributed medium access mechanism used in IEEE 802.11 and IEEE 802.11e in Section 3.2. In Section 3.3, we motivate the need for addressing channel conditions and network load variations to improve throughput. Next, we provide a description of the selected parameters, and present the ChaPLeT adaptation framework. Experimental results follow next in Section 3.4, demonstrating the effectiveness of our adaptation methodology. We next summarize related work and contrast our technique with these methods. Finally, we conclude in Section 3.6.

3.2 IEEE 802.11e Background

In this section, we provide a description of the medium access mechanism used in the IEEE 802.11 Wireless LAN protocol. Additionally, we briefly describe the improvements made in the IEEE 802.11e to support Quality of Service (QoS), as a reference to our adaptations in a quality-enhanced wireless network.

3.2.1 IEEE 802.11 Distributed Coordination Function

IEEE 802.11 defines two modes of accessing the channel: a distributed approach, called Distributed Coordination Function (DCF) and a centralized approach, called Point Coordination Function (PCF). However, since the latter is not supported by most commercially available wireless cards, we focus only on the distributed contention-based medium access component.

The IEEE 802.11 Distributed Coordination Function is based on the Carrier-Sense Multiple Access/ Collision Avoidance (CSMA/CA) mechanism. In order to gain access to the channel, each node must follow a number of steps before transmitting their packets. Figure 3.1 illustrates the IEEE 802.11 DCF protocol. First, the node senses the channel and waits to see if the channel remains idle for a minimum duration defined by the standard, called DCF Interframe Spacing (DIFS). When this criteria is satisfied, the node moves to the next phase of channel access, which is called the backoff. In the backoff, the node chooses a random number of time slots to wait before accessing the medium, where the backoff duration is chosen to fall in a range predefined by each node, called the Contention Window (CW).

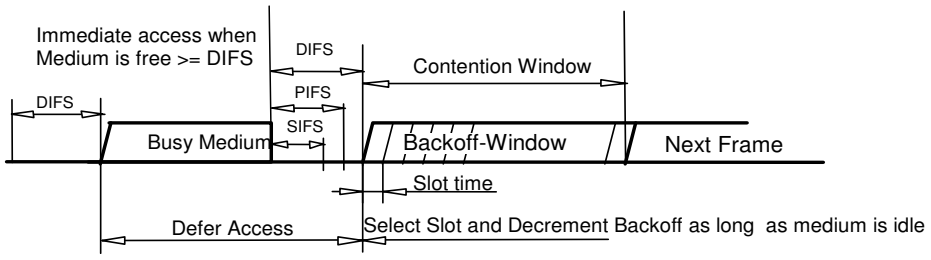


Figure 3.1: IEEE 802.11: Distributed Coordination Function

After choosing a backoff duration, the node then proceeds to countdown. However, if another node accesses the channel during the backoff duration, the node performs a deferral, where it will stop its counter and record the remaining number of slots. The node only resumes the backoff countdown when the channel remains idle for a minimum DIFS period. However, once the backoff duration is complete, the node will proceed to

transmit the packet and will wait for an acknowledgment frame (ACK). The ACK frame is only sent by the receiving node after a short IFS (SIFS) interval, which is a shorter duration than DIFS. If an ACK is not received within a timeout period, the packet will be retransmitted. Note that a retransmission can occur because of various reasons, including poor channel conditions or packet collisions. However, IEEE 802.11 takes precautionary measures and assumes all retransmissions are caused by collisions. In order to decrease the probability of collisions between contending nodes, the maximum backoff period is increased by a magnification factor of two. The following formula determines the CW, where i is the number of retransmissions made for the particular packet and $CW_i = 2^i * CW_{min}[TC]$. Note that the CW size initially begins with a value of CW_{min} and has an upper bound of CW_{max}.

$$Backoff = DIFS + rand(0, CW_i) * SlotTime \tag{3.1}$$

In summary, a packet transmission cycle can be described with three distinct periods: an *access period*, which includes the DIFS and backoff duration, a *deferral period*, which is the time elapsed during deferrals, and finally, a *transmission period*, which is the time for the packet to be transmitted and acknowledged.

3.2.2 IEEE 802.11e Enhanced Distributed Channel Access

One of the shortcomings of the IEEE 802.11 DCF protocol is its inability to support differentiated services. Having recognized this need, the IEEE 802.11e standard makes improvements to support QoS [49]. Although the standard describes improvements in both centralized and distributed channel access, we consider only the distributed channel-access mechanism of IEEE 802.11e, called Enhanced Distributed Channel Access (EDCA), in this chapter.

In a quality-enhanced network supporting EDCA, as illustrated in Figure 3.2, traffic flows are categorized into different classes called *access categories* (AC) and are used to decide channel access parameters of traffic flows. Specifically, the access

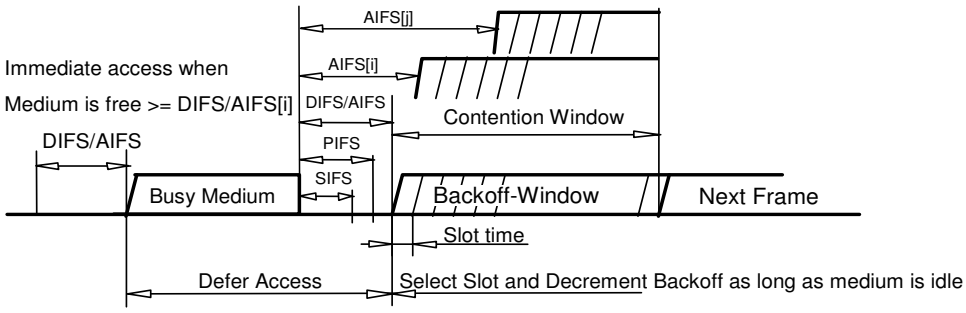


Figure 3.2: IEEE 802.11e: Enhanced Distributed Channel Access (EDCA)

point (AP) decides values of channel access parameters, such as Arbitration Interframe Spacing (*AIFS*), Transmission Opportunity (*TXOP*), and Contention Window parameters (CW_{min} and CW_{max}), for each access category. Next, the access parameters are beaconed by the AP to the nodes in the network. Having received the appropriate values for these channel access parameters, each node uses these values in contending for the medium. The end result is that channel access is dependent on the priority of the access category. For example, traffic classes with higher priority use lower values for $AIFS$ and $CW_{min/max}$. Hence, prioritized nodes will experience a lower waiting period on average than other classes.

3.3 Channel-dependent Packet Level Tuning

Having provided some background of IEEE 802.11 WLAN networks, in this section, we describe our proposed Channel-dependent Packet Level Tuning (ChaPLeT) framework. First, we describe the problems that can occur due to channel variance and network load, and present an overview of our adaptation policy. We then describe the effects of selected parameters, which include *fragmentation threshold*, *persistence factor*, and *defer countdown*. Next, we present runtime adaptation policies using the presented parameters to improve achieved throughput. In addition to describing these adaptations in the context of IEEE 802.11, we describe how these adaptation policies can be used in quality-enhanced networks for service differentiation.

3.3.1 Problem Description and Adaptation Overview

As described earlier, the packet transmission cycle consists of three phases: (1) *access period*, which is the minimum time (number of backoff slots) a node has to wait before attempting to transmit; (2) *defer period*, which is the time consumed in waiting for other nodes' transmissions to finish prior to accessing the medium; and (3) *transmission period*, which is the time consumed in sending the packet/fragments successfully. Channel variance and network load can have a significant impact on the packet transmission cycle and can lead to throughput degradation. When poor channel conditions are present, a large number of retransmissions can occur. With each retransmission, the node must re-access the channel using a magnified contention window, defer for other nodes, and retransmit the current packet. Hence, the medium access delay increases significantly with poor channel conditions. Similarly, when the network load is high, the defer period of a node will increase and can cause delayed transmissions. The throughput degradation in IEEE 802.11 networks caused by channel variance and network load is amplified in quality-enhanced networks, such as IEEE 802.11e. In these networks, throughput degradation can affect service differentiation and lead to further problems, such as *priority inversion*, where higher priority flows experience lower throughput than other flows, or *starvation*, where low priority flows are unable to access the medium.

In order to improve achieved throughput and counter channel and network load variations, we consider dynamically adapting data link parameters based on monitored information. Specifically, we target adapting parameters related to different phases of the packet transmission cycle. The three parameters we consider are *fragmentation threshold* to reduce experienced retransmissions and dropped packets, *persistence factor* to avoid unnecessary channel access delay during retransmissions, and *defer countdown* to decrease channel access delay when a high network load is experienced. For each of these parameters, the *ChaPLeT* tuner determines when the adaptation should be run, which parameters are appropriate to adapt, and also what parameter settings are appropriate for the current channel conditions.

3.3.2 Tunable Parameters

In this subsection, we describe three tunable parameters that can be used to alleviate the problems caused by varying channel conditions in both the IEEE 802.11 and 802.11e networks. We next motivate the use of each of the parameters and describe their effects.

Fragmentation Threshold

In order to address poor channel conditions, most wireless networks provide the ability to partition higher level packets into smaller units. In IEEE 802.11, this is made possible through a parameter, called the *fragmentation threshold*, which indicates the maximum packet size before fragmentation is performed. Since the probability of a successful transmission increases with smaller packet size, fragmentation can be used to respond to dynamic erroneous channel conditions and improve the packet transmission reliability.

Although fragmentation can improve the probability of a successful transmission, partitioning large data packets into smaller fragments can lead to higher data transmission overhead. When a packet is partitioned, each fragment is augmented with individual headers, sent as an independent transmission, and acknowledged individually. Hence, under good channel conditions, fragmentation can lead to suboptimal performance.

The fragmentation threshold is a parameter that can be used to counter the problems that can occur in the presence of channel variations. Nodes that are unable to meet their traffic requirement because of poor channel conditions can choose their fragmentation threshold appropriately to improve their experienced throughput. We next present an analysis to determine the appropriate value of the fragmentation threshold.

The throughput per packet, T, can be calculated as the ratio of the packet size, X, to the expected total transmit time, t_{pkt}. The total transmit time, t_{pkt}, can be written as $t_{pkt} = (n_{frag} * a_{rtx} * t_{frag})$, where n_{frag} is the number of fragments after a packet has

been partitioned, a_{rtx} is the average number of attempts before a successful fragment transmission, and t_{frag} is the transmission time for each fragment.

Given a fragmentation threshold of *F* (bits), the probability of a fragment not in error is given by $p = (1 - BER)^F$, where ***BER*** is the bit error probability. Note that a packet is considered erroneous in this analysis when one or more bits are in error, as current IEEE 802.11 protocol does not have any form of forward error correction. With this probability, we calculate the expected number of attempts required a_{rtx} to be $1/(1-BER)^F$. With *X* representing the upper layer packet size in bits, we can replace the number of fragments, n_{frag}, with the value X/F. Furthermore, the time consumed by each fragment, t_{frag}, can be expressed as a combination of the transmission time and the channel access time of the packet. We use the expression F/R for the transmission time, where R represents the data rate, and for the channel access time, we assume a value C representing average overhead time per fragment, which includes inter-frame spacing, medium access delay, and acknowledgment time. The throughput, T, experienced by a packet can thus be expressed with the following equation.

$$T = \frac{1}{\frac{1}{F} * \frac{1}{(1-BER)^F} * (\frac{F}{R} + C)} \tag{3.2}$$

In order to find the optimum fragmentation threshold, we compute the gradient (dT/dF) and equate it to zero. The optimum fragmentation threshold can be expressed with the following formula:

$$F_{opt\_thrpt} = -\frac{C * R}{2} + \sqrt{\frac{C * R}{BER}} \tag{3.3}$$

Persistence Factor

The next parameter we consider is the factor used to expand the CW for medium access after a retransmission. As described in the background section, after a failed transmission attempt, IEEE 802.11 assumes a collision has occurred and takes a precautionary measure to magnify the CW for the next attempt. In the current IEEE 802.11

protocol, the magnification factor is statically fixed for all nodes to the value of two. However, previous work, including earlier versions of IEEE 802.11e, has considered varying the magnification factor and has named this parameter the persistence factor, PF. The focus of this previous work is to provide service differentiation by associating lower priority nodes with larger PFs and the contention window is calculated with the following equation: $CW_i = PF^i * CW_{min}$. With this proposed improvement, higher priority nodes obtain earlier access to the channel while retransmitting.

In this chapter, rather than use the PF for service differentiation, we consider varying the parameter to reduce channel access delay during retransmission. We note that the magnification of the contention window for every retransmission is only effective when retransmissions are caused by collisions. However, when the retransmissions are caused by poor channel conditions, the magnification and use of PF unnecessarily increases the channel access period of a node. Hence, in addition to the delays from the retransmission of the packet, a node suffering from poor channel conditions will experience high channel access time unnecessarily.

In order to address this problem, we consider changing the PF to improve a node's achieved throughput. By recognizing the case when channel conditions are the cause of retransmissions other than collisions, we can avoid magnifying the contention window used by the node and reduce the channel access delay in subsequent retransmissions.

Defer Countdown

In addition to the above parameters, another mechanism we target is the deferral process. As described in the background section, a node defers access to the medium if another node transmits before the backoff duration is completed. In this case, the node freezes the backoff countdown, and waits for the channel to be idle to resume the residual backoff duration. The waiting time introduced by other nodes accessing the medium can lead to significant delay in the packet transmission cycle. Due to high

network load and poor channel conditions, the effect of a large delay in the defer period will lead to a decrease in the average throughput and in the extreme case, can lead to starvation.

In quality-enhanced networks, such as IEEE 802.11e networks, deferrals will be amplified for lower priority nodes, since the defer periods of lower priority nodes increase significantly with EDCA's provisions for service differentiation. Although a decrease in lower priority nodes throughput is necessary to provide service differentiation, the degradation can be extreme and lead to cases of starvation.

In order to prioritize nodes that have experienced a large number of deferrals, we consider the effect of scaling the *defer countdown*, which is the countdown rate used for the residual backoff period. By speeding up the backoff residual duration, the node has a higher probability of accessing the channel and can be able to avoid continual deferrals. The speedup of the node's *defer countdown* will be effective in improving throughput when the throughput degradation is due to the extra wait time due to deferrals. However, the scaling should not be used excessively by a node since the abuse of the scaling can lead to decreased throughput by other nodes in the network. In order to avoid this, changing the *defer countdown* is only allowed based on the wait ratio experienced by the node, which is the ratio of experienced to expected wait time, and the number of deferrals experienced. Appropriate scaling can aid in preventing starvation, by providing higher throughput to nodes experiencing excessive deferrals and obtaining less than their required throughput.

3.3.3 Adaptation Policy

Having described the individual parameters and their effects, we now describe our proposed framework to adapt these parameters, which we call Channel-dependent Packet Level Tuning (ChaPLeT). ChaPLeT takes a distributed approach of improving throughput with each node monitoring and tuning its parameters based on current channel conditions. The ChaPLeT framework consists of two major components, a *monitor*

and a *tuner*, as seen in Figure 3.3.

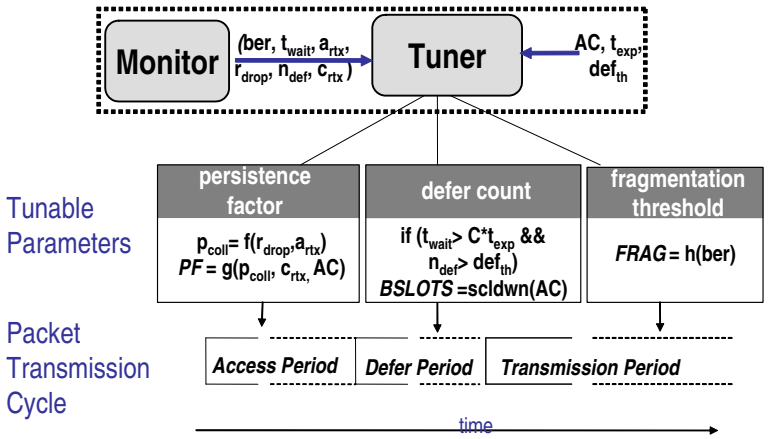


Figure 3.3: ChaPLeT Adaptation Framework

The first component, the *monitor*, records and estimates the experienced channel conditions, such as BER, dropped packet ratio, r_{drop}, and average number of retransmissions per packet, a_{rtx}. The tuning functionality is designed to completely reside in the link layer. For instance, the monitor estimates BER based on the current average number of retransmissions and packet sizes. Additionally, the *monitor* tracks packet-specific information, such as the time spent on the current packet, t_{wait}, number of deferrals n_{def}, and current retransmission count, c_{rtx}.

The ChaPLeT *tuner* keeps track of node specific information, such as the expected waiting time for the current packet, t_{exp}, tolerable number of deferrals, def_{th}, and the access category, AC, if the node supports EDCA. The *tuner*'s choice of parameters affects the three phases of the packet transmission cycle. For each of the parameters, the tuner must decide when the adaptation should be run, which parameters to adapt, and also what parameter setting is appropriate for the current channel conditions. The individual parameter adaptations used by the tuner are described next. Pseudo-code for important parts of ChaPLeT functionality is shown next.

```
/* ChaPLeT algorithm */
ChaPLeT () {
 /* Initialize the adaptations */
 pf_adpt = 0;
 def_adpt = 0;

 /* Monitor is called when a packet is successfully sent, dropped,
    ready to transmit/retransmit, and when a deferral occurs. */
 {ber, t_wait, a_rtx, r_drop, n_def, c_rtx} = monitor ((Event) e);

 /* Tuner is called before a packet is sent*/
 {frag, pf, bslots} = tuner (ber, t_wait, a_rtx, r_drop, n_def, c_rtx);
}

/* Tuning algorithm executed by each node before packet is sent*/
tuner (ber, t_wait, a_rtx, r_drop, n_def, c_rtx) {
  /* Fragmentation adaptation, after every frag_th packets have been sent*/
  if ((n_pkt > frag_th) || (r_drop > 0.02)) {
    frag = fragmentation(ber);
    n_pkt = 0;
  }
  else n_pkt++;

  /* Persistence factor adaptation performed for each packet
     and scaledown occurs only once per packet*/
  if ((pf_adpt == 0) && (c_rtx >= 2) && (a_rtx > r_th)) {
    pf = pf / pf_scaledown[AC];
    pf_adpt = 1;
  }

  /* Defer countdown adaptation performed for each packet
     and scaledown occurs once per packet*/
  if ((n_def > def_th) && (def_adpt == 0) &&
```

```
        (t_wait/t_exp) > C) && (bslots > CWmin[AC]/2)) {
        bslots = ceil(bslots /df_scaledown[AC]);
        def_adpt = 1;
    }
}
```

Fragmentation Threshold

The first parameter we focus on is the fragmentation threshold. Using our analysis from the previous subsection, the ChaPLeT tuner uses the BER estimate provided by the *monitor* and decides the optimum fragmentation threshold to maximize throughput under current channel conditions. This adaptation can benefit nodes that are suffering from poor channel conditions and are unable to meet their desired throughput. In order to control the frequency of the adaptation, the adaptation is run only after $frag_{th}$ packets have been sent or when the packet drop rate (r_{drop}) is high. Running the fragmentation adaptation too frequently can lead to oscillations in fragmentation threshold value. A less frequent adaptation, on the other hand, cannot respond to channel variations in time and may degrade the performance. Hence, the choice of $frag_{th}$ can be scaled according to current channel conditions. For example, if the node is experiencing large variations in the drop rate or recorded BER, then the $frag_{th}$ is set to a smaller value dynamically. Additionally, after deciding the optimum fragmentation threshold, we attempt to make the size of the fragments similar by allocating evenly over the total number of the fragments. The smoothening process increases the chance of each fragment being received correctly without increasing the overhead associated with each fragment.

Persistence Factor

Another parameter that ChaPLeT adapts is the persistence factor. The first aspect of the ChaPLeT *tuner* is to decide when the PF parameter should be adapted. As noted earlier, the PF should only be adjusted if the experienced retransmissions are not caused by collisions. Hence, in order to set the PF, the *tuner* first evaluates the current channel

conditions and estimates whether the experienced retransmissions are mainly caused by collisions or by poor channel conditions.

We use a threshold-based algorithm to estimate the chances of collision using the average number of retransmissions as the input metric. Since the probability of a collision decreases with each retransmission attempt, high observed values of average retransmissions, a_{rtx}, indicate that retransmissions are likely to be caused by poor channel conditions, rather than collisions. From our experiments, described in greater detail in Section 3.4, we observed that the a_{rtx} is very high for scenarios suffering from poor channel conditions and varies based on the current BER experienced. On the other hand, the value of a_{rtx} does not increase significantly when the retransmissions are caused by collisions or high network load. Hence, ChaPLeT scales the PF if the experienced a_{rtx} exceeds a set threshold r_{th} and the scaling of the PF is based on the estimated probability of collision, p_{coll}. The adaptation of PF also depends on the node's access category, AC, if it is in a quality-enhanced network. This allows nodes with higher priority to reduce their PF by a larger value than other nodes and be able to increase their throughput under poor channel conditions. With this adaptation, a higher priority node may be able to avoid priority inversion.

Defer Countdown

The final parameter determined by the ChaPLeT *tuner* is the scaling of the defer countdown. As noted earlier, the defer countdown can be used to improve throughput when deferrals lead to large delays and throughput degradation. In order to determine when the adaptation should be run, the ChaPLeT *tuner* factors in the waiting time introduced by others accessing the medium. For each packet, using the time-stamp of the arrival of the packet, the *monitor* calculates the current wait time, t_{wait}, and compares this value with the expected wait time, t_{exp}. The t_{exp} is determined using the packet size, expected throughput, and data rate currently being used.

If the node has experienced a large number of deferrals, n_{def}, *i.e.* greater than def_{th}, and has experienced a wait ratio, t_{wait}/t_{exp}, greater than a set threshold, C, the

packet is flagged and the adaptation algorithm speeds up the countdown process by changing the rate at which the countdown continues. This will expedite the channel access when the waiting period caused by deferrals exceeds tolerable limit for the node. The value of C can be used to indicate the minimum tolerable throughput for the flow and when the adaptation should occur.

In quality-enhanced networks, we consider using the defer countdown adaptation for low priority nodes to avoid the possibility of starvation, which can be amplified with service differentiation. Although a throughput decrease in lower priority nodes is necessary to provide service differentiation, the defer countdown adaptation gives these nodes a means of avoiding the extreme case of starvation.

The ChaPLeT adaptation policies described above affect different periods in the packet transmission cycle. As can be seen in Figure 3.3, the fragmentation threshold primarily affects the transmission period, the persistence factor affects the channel access period, and the defer countdown aids with a decrease in the defer period. Hence, the parameter adaptations of ChaPLeT can be run in parallel.

3.4 Experimental Results

In this section, we present the experimental evaluation of our proposed adaptation policies and framework. We begin by briefly describing the experimental setup and system configuration. We then evaluate the effect of individual parameter adaptation. Next, we present the results demonstrating the effectiveness of the overall ChaPLeT framework. Finally, we discuss the performance of the ChaPLeT framework in a quality-enhanced network, similar to IEEE 802.11e.

3.4.1 Setup and System Configuration

For our experiments, we use the *Opnet* simulation environment [84]. In order to evaluate the performance of ChaPLeT, we modified the IEEE 802.11b MAC layer mod-

els to incorporate our adaptation scheme. We used the *Opnet* Direct Sequence Spread Spectrum (DSSS) physical layer with a data rate of 11 Mb/s. The *Opnet* tool estimates channel error and channel variations using a pipeline model, which evaluates characteristics such as propagation loss and interference. For generating different types of application traffic (video, chat, ftp), we used the *Opnet* application and profile configuration to set the packet size, as well as the inter-packet gap for the traffic generation process.

For the quality-enhanced network experiments, we modified the *Opnet* model to incorporate the EDCA service differentiation. We consider four traffic classes and the parameter values for different traffic classes (priorities) are summarized in Table 3.1. We assume the same value for CW_{max} and TXOP in these experiments. The default characteristics of the simulated wireless networks and parameters have been chosen similar to those that presented in [85].

Table 3.1: EDCA Simulation Parameter Settings

Parameters	EDCA Classes			
	H	M1	M2	L
AIFS	2	2	2	4
CWmin (slots)	7	15	15	31
PF	2	2.5	3	4
Traffic (Kb/s)	512	256	128	64

3.4.2 Individual Parameter Adaptation

We first begin with evaluating the effects of the individual parameter adaptations. For our comparative evaluation, we use the link layer throughput as the evaluation metric, which is computed as the average of the experienced packet-level throughput. Note that the packet-level throughput is calculated using the size of the packet and the time spent per packet *i.e.* the time period between packet arrival from the upper layer and the acknowledgment by the receiver. The throughput computation includes the delay due to retransmission and reports zero throughput when a packet drop occurs. Although there

are other quality of service metrics, such as delay and jitter, we only consider the link layer throughput metric with the simulations in the quality-enhanced networks.

Fragmentation Threshold

We first evaluate the effectiveness of the fragmentation threshold. For this parameter, we first investigate the effect of setting different values of the fragmentation threshold on throughput under varying channel conditions. Figure 3.4 presents the results of the evaluation. As can be seen in the figure, for good channel conditions ($BER = 0.00001$), the larger the packet size, the higher the throughput experienced. On the other hand, for poor channel conditions smaller packets, it can be seen that throughput can be improved by setting the fragmentation threshold appropriately. For instance, for BER = 0.0001, by setting the fragmentation threshold very high, the node does not fragment any of its packets but achieves very low throughput. However, by fragmenting packets with a threshold of approximately 512 bytes, the node can improve its achieved throughput to 3 Mb/s. The circled points indicate the optimum fragmentation threshold values determined using our algorithm based on Equation 3.3 for the specified BER. Note that our choice of fragmentation threshold corresponds with the optimal value for maximum throughput for most of the test cases.

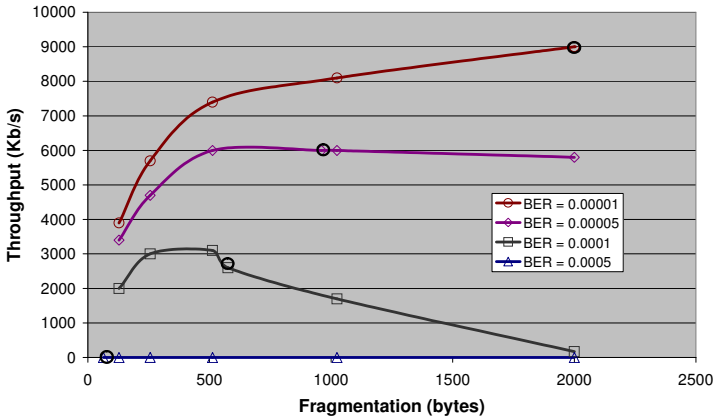


Figure 3.4: Throughput Variations using Different Fragmentation Thresholds

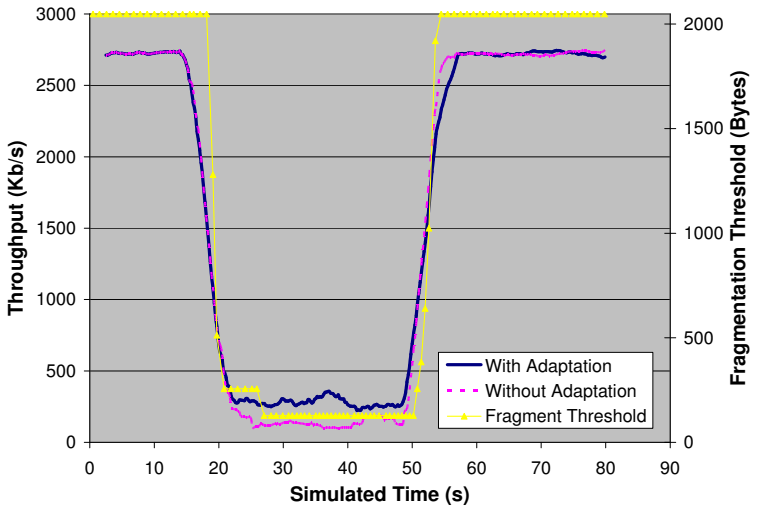


Figure 3.5: Throughput Improvement with Dynamic Fragmentation Threshold Adaptation

Next, we evaluate the responsiveness of the fragmentation threshold adaptation and its ability to track dynamic variations in channel conditions. For this evaluation, we observe the throughput of a node that moves away from the access point, reaches a point where it is experiencing poor channel conditions, and then moves back toward the access point. Figure 3.5 shows the moving average of the node's experienced throughput with and without adaptation during the simulation period. The figure also shows the fragmentation threshold values (right y-axis) used by the adaptation policy during the simulation period. Note that when the node transitions from experiencing good channel conditions to poor channel conditions (after 20 seconds of simulation), the adaptation policy changes the value of the fragmentation threshold and provides higher throughput (up to 3x) than without adaptation. Similarly, when the channel conditions improve as the node returns (after 50 seconds of simulation), the adaptation algorithm recognizes that the packets no longer need to be fragmented. In summary, this figure illustrates that the fragmentation threshold can lead to an improvement in experienced throughput.

Persistence Factor

Next, we focus on evaluating the effect of the persistence factor (PF) adaptation. One important characteristic of this adaptation is to be able to recognize when a retransmission is caused by poor channel conditions or by collision. As described in the parameter description, we use the average number of retransmissions per packet, a_{rtx}, as an indicator of the collision rate. In Figures 3.6 and 3.7, we show two curves from our simulations, the average number of retransmissions as a function of BER, and the average number of retransmissions as a function of nodes in the network. Note that from these results, the value of a_{rtx} increases drastically with higher BER but does not vary by much with the number of nodes.

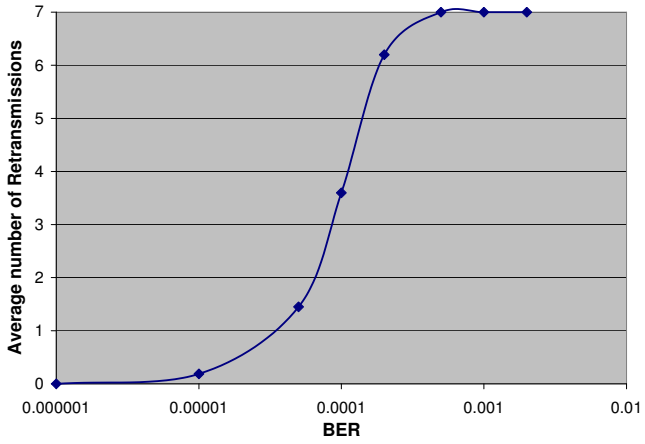


Figure 3.6: Experienced Average Retransmission for Different Channel Conditions

The next aspect we evaluate is whether the adaptation policy is able to adapt to varying conditions. We look at the throughput experienced by a node as the channel conditions change. The node faces three distinct periods in the simulation: (1) good channel conditions; (2) poor channel conditions; and (3) high network load. Figure 3.8 presents the throughput achieved during these variations with and without adaptation. The figure also reports the average number of retransmissions used in the adaptation. From the figure, it can be seen that with adaptation the throughput can be improved by a factor of 2 under poor channel conditions (between 20 to 50 seconds of simulation).

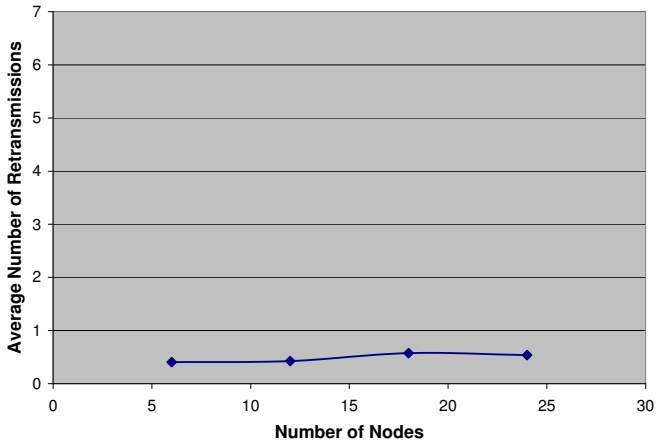


Figure 3.7: Experienced Average Retransmission with Different Number of Nodes

It can also be seen that the algorithm makes changes in the persistence factor appropriately. For example, when the node experiences poor channel conditions, it experiences a significant number of retransmissions and the persistence factor is adapted. However, the PF adaptation is disabled when the node moves back to good channel conditions. It also remains disabled in the case where there is a higher load in the network (between 80 to 100 seconds of simulation) since the a_{rtx} is less than our chosen r_{th} of 2. From these simulations, it is clear that a node suffering from poor channel conditions can improve its throughput by recognizing that its retransmissions are not being caused by collisions and adapting PF accordingly. Additionally, the throughput of other nodes in the network was not negatively affected by the adaptation.

Defer Countdown

The final parameter that we evaluate is the defer countdown. With this parameter, we wanted to evaluate how the adaptation performs as the number of nodes increases. Table 3.2 summarizes the throughput achieved with and without adaptation in scenarios with varying number of nodes. As the number of nodes increase, the number of deferrals and the experienced wait ratio increases. For each scenario, we evaluate the effect of two

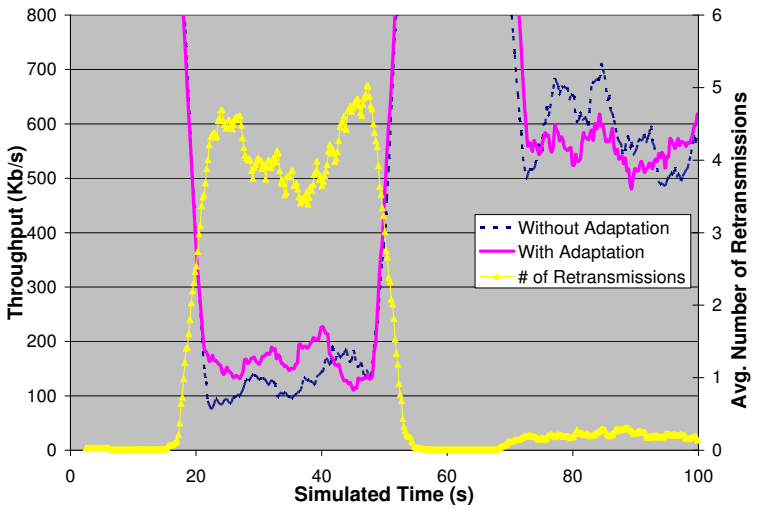


Figure 3.8: Throughput Improvement with Dynamic Persistence Factor Adaptation

wait ratio threshold values, $C = 10$ and $C = 5$. With the wait ratio threshold set to five, the adaptation is more aggressive and attempts to start speeding up the defer countdown too often. This can lead to lower throughput and high numbers of collisions. For example, in the case of 12 and 14 nodes, $C = 5$, leads to a slight decrease in the throughput achieved. However, since this adaptation is designed specifically for cases of starvation or high wait ratios due to deferrals, we evaluate our adaptation using a higher threshold of $C = 10$. With our adaptation, we are able to improve throughput (up to 6x) in all the cases where adaptation was necessary. We can conclude from these evaluations that the defer adaptation is more effective in situations where the network is highly loaded but not completely saturated. In addition, for these scenarios, we computed the Jain Fairness Index [86] with and without adaptation and observed that the adaptation does not have a negative impact.

3.4.3 Overall ChaPLeT Adaptation

In this section, we evaluate the effectiveness of the ChaPLeT framework to dynamically adapt the three selected parameters automatically in four different scenarios, S1-S4. Briefly, the scenarios can be described as nodes experiencing poor channel con-

Table 3.2: Defer Countdown Adaptation

Adaptation \ Number of Nodes	Throughput (Kb/s) 6	12	14	16	18
None	1201	738	600	40	6
C=10	1201	738	619	277	20
C=5	1201	722	597	316	24

ditions, good channel conditions, mixed static channel conditions, and mixed dynamic channel conditions caused by mobility. For each scenario, we evaluate the performance of our adaptation in terms of average throughput over each simulation run. We also compute the mean and maximum percentage improvement achieved over the simulation by a node. Table 3.3 shows the results of our comparison. To better understand the results, we next describe each scenario and discuss the effects of the adaptation.

Table 3.3: Effect of ChaPLeT Adaptation

	Throughput (Kb/s)							
	Scenario 1 [1]		Scenario 2		Scenario 3		Scenario 4	
	Without	With	Without	With	Without	With	Without	With
Mean	71	1164	1789	1789	642	615	1109	1145
Max	39	1128	1169	1240	150	330	151	278

[1] *Scenario 1*: Nodes experiencing poor channel conditions; *Scenario 2*: Nodes experiencing good channel conditions; *Scenario 3*: Half of the nodes experiencing good channel conditions and other half experiencing poor channel conditions; *Scenario 4*: One node moving away from AP, other nodes experiencing good channel conditions

In our first scenario, S1, we consider eight nodes that are all suffering from poor channel conditions, *i.e.* a BER of 0.0001. Note that the throughput achieved is extremely low when poor channel conditions are present and is on average around 71 Kb/s. However, as can be seen in Table 3.3, with the ChaPLeT adaptation, we can significantly improve the throughput for all the nodes, with an average improvement of 15x and a peak improvement of 28x. The majority of the performance gain in this case is from the use of fragmentation threshold and persistence factor adaptation.

We next investigate the effect of the ChaPLeT adaptation in the case where all nodes are experiencing good channel conditions. In this scenario, S2, the objective is to evaluate whether the ChaPLeT framework has a negative impact on the system because of unnecessary adaptations. It can be seen in Table 3.3 that the change in the throughput achieved is very minimal. This indicates that the ChaPLeT framework is activated infrequently in this scenario and does not negatively impact nodes when they are in good channel conditions.

In our third scenario, S3, we consider a static topology where the nodes are experiencing a varying set of channel conditions. With the scenario's topology, half of the nodes are experiencing good channel conditions and the rest are experiencing poor channel conditions (BER = 0.0001). In the non-adaptive case, nodes under good channel conditions experience extremely high packet level throughput. However, the other nodes suffering from poor channel conditions only achieve throughput around 150 Kb/s. With adaptation, there is a slight decrease in the throughput averaged over all nodes. However, if we look more closely at the scenario, we see that this is due to the adaptation's reallocation of resources. By improving the nodes with poor channel conditions by up to 120%, the other nodes in good channel conditions experience a slight decrease in throughput. However, the actual experienced throughput of these nodes remains high, not falling below 1 Mb/s.

In the fourth scenario, S4, we consider a network consisting of eight nodes with one node moving away from the access point for a brief amount of time and returning to its original location. We simulate this scenario to see whether the ChaPLeT adaptation can track the dynamic channel conditions caused by the node's movement. Since the other nodes in the scenario are not experiencing poor channel conditions, the change in throughput with and without adaptation is minimal (3%). For the moving node, however, the average throughput improvement for the scenario is 18% with the peak improvement of 84%. Another thing to note is that the adaptation of the moving node has no negative effects on the other nodes in this scenario.

3.4.4 ChaPLeT Adaptation in Quality-Enhanced Networks

Next, we revisit the above scenarios and evaluate them in a quality-enhanced setting. To do this, we classify nodes into different traffic classes and priorities. Specifically, we assign three nodes with H priority, two nodes with M1 priority, one node with M2 priority, and two nodes with L priority. The priorities of the nodes are fixed across scenarios; however, location and experienced channel conditions differ between scenarios. Table 3.4 shows the throughput of each of these nodes. We will next describe the results of each scenario and the effects of the adaptation.

Table 3.4: Effect of ChaPLeT Adaptation in Quality-Enhanced Networks

Node	Pri.	Throughput (Kb/s)							
		Scenario 1 [2]		Scenario 2		Scenario 3		Scenario 4	
		Without	With	Without	With	Without	With	Without	With
1	M2	362	450	1120	1079	**813**	**797**	1058	1052
2	M2	695	926	1030	1090	794	738	1058	1126
3	M1	758	1272	5500	5500	4632	5065	5478	5475
4	H	841	1546	1780	1749	**680**	**830**	1811	1822
5	H	861	1437	1670	1699	2121	2119	**1041**	**1268**
6	H	1856	1877	1870	1870	670	804	1856	1877
7	L	334	259	316	323	**1**	**52**	307	305
8	L	278	185	291	316	**1**	**51**	310	304

[1] Nodes are prioritized into 4 categories: H, M1, M2, L; *Scenario 1*: Nodes experiencing poor channel conditions; *Scenario 2*: Nodes experiencing good channel conditions; *Scenario 3*: Half of the nodes experiencing good channel conditions and other half experiencing poor channel conditions; *Scenario 4*: One node moving away from AP, other nodes experiencing good channel conditions

In our first scenario in S1, the nodes are all suffering from poor channel conditions and experience low throughput. Note that despite the low throughput, the priority is maintained between the nodes when the experienced channel conditions are similar. With the ChaPLeT adaptation, we can maintain the priority of the nodes and also improve the throughput for most of the nodes. We can explain the slight decrease in the throughput for low priority nodes by looking at their achieved throughput more closely and understanding the level of adaptation of the nodes. Since the low priority nodes

are not suffering from starvation in this scenario, the main adaptation that is being used with ChaPLeT is the fragmentation threshold. Since the nodes with higher priority have additional adaptation techniques, they are able to improve their throughput with a slight impact on these lower priority nodes. It is also important to note that although low priority nodes experience decrease in throughput, the degradation is minimal.

For the second scenario, S2, we consider the case where all the nodes are experiencing good channel conditions. It can be seen in Table 3.4 that the results are similar to that without service differentiation and indicate that the change in the throughput experienced is very minimal. Since the ChaPLeT framework is activated infrequently in a scenario where there are good channel conditions, the adaptation does not have a significant negative impact on the flows.

Next, in S3, we consider a case where there are prioritized nodes experiencing various channel conditions. In this scenario, we can observe both problems of starvation and priority inversion. It can be observed that nodes 7 and 8 are starving experiencing very low throughput (1 Kb/s). The starvation occurs because the low priority nodes suffer from poor channel conditions and a large number of deferrals. In addition, it can be observed that a medium priority node (node 1) experiences higher throughput than a high priority node (node 4). The priority inversion occurs because some high priority nodes (nodes 4 and 6) are facing poor channel conditions and achieving a lower throughput than the medium priority nodes (nodes 1 and 2). With the ChaPLeT adaptation, the results from Table 3.4 indicate that the achieved throughput for the high priority node exceeds the medium priority node, indicating that priority inversion has been avoided. Furthermore, the results illustrate that the low priority nodes are able to avoid starvation.

Finally, we look at scenario S4, where one high priority node (node 5) moves away from the AP and returns to its original location. In this scenario, the node moving back and forth experiences throughput degradation and priority inversion (comparison to nodes 1 and 2). From the table, it can be seen that the ChaPLeT adaptation improves the node's throughput from 1041 to 1268 Kb/s (an improvement of 22%) with minimal effects on the other nodes. To analyze further, Figure 3.9 illustrates the moving average

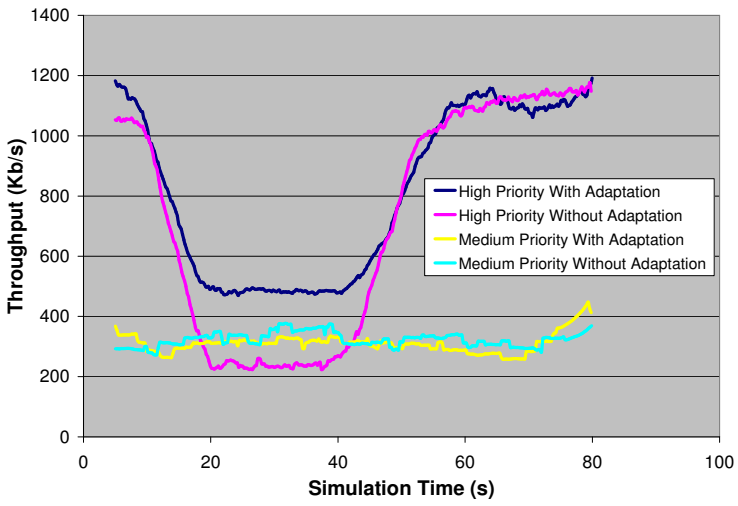


Figure 3.9: Effect of ChaPLeT Adaptation on Moving Node

of the throughput achieved by the observed high priority node and a low priority node. Note as the node moves away from the access point, the throughput decreases significantly, causing a throughput degradation severe enough to cause a priority inversion between the nodes. However, as can be seen in the graph, with the use of the ChaPLeT adaptation, the high priority node is able to achieve higher throughput and improve its throughput by 2x.

Having demonstrated the effectiveness of the proposed framework in combating channel-related problems in providing QoS in a wireless network, we next present some related work in this research area and a comparison of our approach.

3.5 Related Work

In this section, we present a brief summary of previous work and contrast our proposed approach in the area of link layer adaptations for addressing channel conditions and network load in IEEE 802.11. Briefly, the proposed link layer adaptations can be categorized into two main areas: adaptation of channel access parameters and adaptation of transmission related parameters. The adaptation of channel access parameters

affect the medium access delay, where as the transmission related parameters address the delay experienced in transmitting a packet after obtaining access to the medium.

With respects to channel access parameters, most of the previous work has been focused on changing these parameters either to enable service differentiation or to consider network load. One of the parameters discussed and adapted in this work, is the contention window (CW), which is used to determine the backoff duration before accessing the medium. Rather than using the binary exponential expansion described in IEEE 802.11 for this parameter, previous work has suggested that setting the parameter using different values can have a significant impact ([87],[88]). These adaptation techniques have ranged from simple approaches, such as a look-up table approach based on theoretical analysis [37] or by deciding on the value of CW based on the previous packet [89], to more complicated approaches, such as CW scaling using a token-bucket algorithm [90] or a fair queuing mechanism proportionate to flow weights [91]. A related parameter to the CW is the persistence factor (PF), which is the magnification factor used by a node to expand the CW to avoid collisions after retransmissions. In previous work, including earlier versions of IEEE 802.11e, this has been used to provide priority to certain nodes in the collision avoidance phase.

Additionally, there have been some attempts to adapt transmission related parameters at the medium access layer to mitigate the impact of bad channel conditions. Past research efforts have proposed adapting one such parameter, called *fragmentation threshold*, that decides the maximum size of MAC Protocol Data Unit (MPDU) to reduce number of unsuccessful retransmissions in order to meet various objectives such as reducing energy consumption ([72],[92]), improving throughput/goodput and countering different periodic patterns of interference such as microwave and Bluetooth devices [93]. Additionally, several physical layer techniques have been proposed, such as Forward Error Correction (FEC), modulation scaling [36], transmission power adaptation ([74],[76]).

In this chapter, our proposed approach concentrates on mitigating the effect of dynamic channel variations and network load by adapting selected MAC layer param-

eters. The adaptation requires minimal modification in the MAC layer for practical realization. While the majority of the previous techniques have singled out one parameter to adapt, the proposed ChaPLeT framework in this chapter considers multiple parameters affecting various periods of the packet transmission cycle. We used some known parameters, i.e. *fragmentation threshold* and *persistence factor*, and introduced a new parameter called *defer countdown* to address channel condition-related throughput degradations. For the fragmentation threshold, we proposed an enhanced adaptation methodology supported by a closed form analytical evaluation of its effect on goodput. We used the *persistence factor* in a novel way to minimize the effect of poor channel conditions and reduce the unnecessary expansion of contention window in the collision avoidance phase. Finally, we introduced a new parameter called *defer countdown* to speed up the deferral process for nodes experiencing starvation. Similar techniques have been used in [94] to speed up access for all nodes to improve throughput in absence of competing traffic. While the proposed method of defer countdown adaptation is similar, the intended use of adaptation differs, since we focus on using the parameter to help avoid starvation.

Additionally, in terms of enabling service differentiation, none of the known techniques consider the effect of channel conditions on service differentiation. We demonstrated the possible effects of channel error and network load on service differentiation, and proposed adaptations to improve service differentiation.

3.6 Conclusion

In this chapter, we presented an enhanced multi-parameter adaptation methodology to address the impact of channel variations and network load on achieved throughput and service differentiation in a WLAN network. Specifically, we discussed the effects of three medium access parameters, namely *fragmentation threshold*, *persistence factor*, and *defer countdown*. We described a runtime Channel-dependent Packet Level Tuning (ChaPLeT) framework that dynamically configures the above parameters to improve

the achieved throughput under poor channel conditions and implemented ChaPLeT on top of the current IEEE 802.11b scheme. Using the *Opnet* simulation framework, we demonstrated that by dynamically adapting these channel access parameters, we are able to achieve significant improvement in throughput. In addition, we discussed and demonstrated how the ChaPLeT framework can be used in a quality-enhanced WLAN network to improve service differentiation.

The text of this chapter, in part, is based on material that has been published in the International Symposium on Wireless Personal Multimedia Communications in 2003, (N. Ramos, D. Panigrahi, and S. Dey, "ChaPLeT: Channel-dependent Packet Level Tuning for Service Differentiation in IEEE 802.11e", *Proceedings of International Symposium on Wireless Personal Multimedia Communications*, vol. 3, pp. 86-90, Yokosuka, Kanagawa, Japan, October 2003) and material published in IEEE Networks Magazine in 2005 (N. Ramos, D. Panigrahi, and S. Dey, "Quality of Service Provisioning in 802.11e Networks: Challenges, Approaches, and Future Directions", *IEEE Networks Magazine*, vol. 19, no. 4, pp. 14-20, July/August 2005). The dissertation author was the primary researcher and author, and the coauthors listed in these publications collaborated on, or supervised the research that forms the basis for this chapter.

Chapter 4

Quality of Service Improvement through Access Point Resource Allocation and Scheduling Policies

4.1 Introduction

Having discussed the potential of using network adaptation at the user device to improve energy and quality of service, we next investigate network scheduling and resource allocation at the access point to improve quality. As discussed earlier, with the growing availability of multimedia content, there has been a significant increase in the use of multimedia applications over wireless networks, such as streaming video, teleconferencing, and voice over IP. With the popularity and technological advancements in these two areas, there will be a need to support multimedia applications over wireless networks. However, the time-varying nature of wireless access and the diverse requirements of multimedia applications make the task of supporting wireless multimedia services challenging.

In this chapter, we address this problem by describing a scheduling scheme that considers the variability of the application and the network and adjusts allocated re-

sources based on user flow requirements. We focus on making this improvement above the IEEE 802.11e [49] standard, which was partially described in the previous chapter. The 802.11e standard provides a framework called Hybrid Coordination Function (HCF) that multiplexes between two modes of medium access: distributed and centralized. With the distributed channel access scheme, each flow gains access to the channel through a contention-based algorithm. This method, EDCA, is targeted for bursty traffic flows with unknown traffic requirements. With centralized access, a polling-based scheme is used to grant guaranteed access to traffic flows. To determine a polling schedule, the centralized channel access scheme uses the flow requirements specified in the reservation request provided by each flow at initialization. The 802.11e standard provides a reference design [95] of a scheduler that can be used in centralized access.

One major category of applications that must be supported by IEEE 802.11e is real-time multimedia flows, such as Voice over IP (VoIP), video streaming, *etc.*. To provide the desired QoS using the centralized access scheme, it is critical to understand the key characteristics of real-time multimedia flows, which depend on a number of factors, such as the type of application, temporal variation in traffic flows, delay and throughput requirements, *etc.*. An important characteristic of multimedia application is the encoding type, such as Constant Bit Rate (CBR) versus Variable Bit Rate (VBR). Applications such as quality-controlled MPEG4 or video-conferencing use VBR encoding to provide better video quality for the same average bit rate and shorter delay. Additionally, service providers commonly use VBR encoding of multimedia content to increase the capacity of the network by multiplexing between different VBR flows [96] [97].

In order to meet the demands of the above real-time applications, it is critical to implement scheduling policies that use the mechanisms provided by the 802.11e standard and provide adequate support for QoS. One possible scheduling policy for QoS provisioning is a fixed allocation policy, where a fixed time period is given to each flow. An example of a fixed scheduler policy is the reference scheduler, [95], which allocates a fixed polling schedule using the averaged values specified in the reservation request, such as mean packet size and required throughput. The reference scheduler assumes that

real-time flows will reserve time for channel access based on their requirements during the centralized scheme. However, depending on the application, accurate information about requirements may not be available at the beginning of the flow's transmission. For example, with the use of application layer adaptation techniques, the bit rate requirements of a video stream change based on the congestion in the end-to-end network; hence, it is not feasible to get accurate information at initialization. Additionally, due to the use of averaged values in the reservation request, the reference scheduler's time allocation may be suitable for flows with CBR traffic but unsuitable for VBR flows. With possible inaccuracies in received reservations and traffic variation, the reference scheduler may be unable to provide appropriate time allocations and lead to unacceptable delays and impacted multimedia quality.

Rather than use the mean requirements to determine time allocation, another fixed allocation policy is to over-provision for each flow at the reservation time. One example of such a scheduler is one that determines time allocation based on the maximum flow requirement. Using overallocation, the scheduler can provide additional time to improve quality for real-time flows with variable needs. Although this method improves the multimedia quality, each flow is given a reservation above its needs, which limits the number of nodes in the network and severely impacts system capacity. An alternative to the reference scheduler and over-provisioning is a scheduling algorithm that provides dynamic allocation. With this type of scheduler, the variable needs of a flow can be identified, estimated dynamically and given additional channel access when it is appropriate. Multimedia flows with these schedulers are given additional time appropriately and achieve improved quality compared to static schedulers without impacting the system capacity (channel utilization).

In this chapter, we present a dynamic adaptation policy to configure the polling-based scheduling policy and associate real-time traffic flows to the contention-based access mode. The proposed policy takes into account the possible inaccuracies in reservation information, the variance in flow generation and throughput requirements, and current system utilization. Additionally, the policy performs these adaptations with min-

imal effects on other flows in the network. To evaluate the effectiveness of our approach, we compare our adaptation policy with the reference design of a scheduler for 802.11e [95] and another proposed scheduler, called Flexible HCF (FHCF). We demonstrate that through our adaptation, we can achieve significant improvement in QoS in terms of delay and throughput metrics. Additionally, we show the impact of the proposed adaptation on the quality of multimedia applications and system capacity.

The rest of the chapter is organized as follows. We first describe the 802.11e channel access schemes in detail in Section 4.2. Next, in Section 4.3, we motivate the limitations of the current approach of allocating flows to different modes of channel access. In Section 4.4, we present an adaptation algorithm to configure scheduling mechanisms and coordinate flows between two access modes. Experimental results follow next in Section 4.5, demonstrating the effectiveness of our adaptation policy. In Section 4.6, we discuss and contrast our work with other related research efforts. Finally, we conclude in Section 4.7.

4.2 IEEE 802.11e Background

In this section, we provide a brief description of the enhancements proposed in IEEE 802.11e [49] to support service differentiation. The proposed standard defines a new operational mode called the Hybrid Coordination Function (HCF). As shown in Figure 4.1, HCF multiplexes between two modes of medium access: a distributed contention-based approach guided by Enhanced Distributed Channel Access (EDCA), and a centralized polling - based approach called HCF Controlled Channel Access (HCCA). Both access functions enhance functionality of the original access methods specified in 802.11a/b/g: Distributed Coordination Function (DCF) and Point Coordination Function (PCF). A basic concept common to the two access schemes proposed in 802.11e is the notion of a transmission opportunity (TXOP). TXOP is a bounded time interval for which a node is allowed to transmit. While TXOPs in EDCA are decided based on traffic flows, in HCCA, TXOPs are selected on a per-node basis.

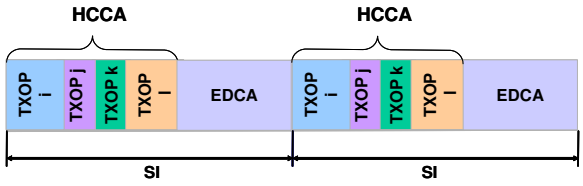


Figure 4.1: Hybrid Coordination Function (HCF)

4.2.1 Enhanced Distributed Channel Access

In order to support service differentiation, 802.11e improves over the legacy 802.11 Distributed Coordination Function (DCF) by providing a differentiated channel access mechanism called EDCA. EDCA is a parameterized version of the previous distributed channel access mechanism of 802.11 and associates different channel access parameters to different traffic flows, categorized into different classes called *access categories* (AC), to prioritize medium access among different flows. The prioritization in channel access through EDCA is shown in Figure 4.2. The modified channel access parameters are listed below:

- *Arbitration Interframe Spacing (AIFS)*: The minimum time interval to wait before starting backoff.
- *Transmission Opportunity (TXOP)*: The maximum duration for which the node can transmit.
- *Contention Window parameters* (CW_{min} and CW_{max}): Decides the random number of slots to wait before starting transmission.

For each AC, the access parameters are decided by the Access Point (AP), and are beaconed to the nodes in the network.

4.2.2 HCF Controlled Channel Access

In addition to prioritized channel access, the 802.11e protocol describes a centralized channel access scheme, called HCF Controlled Channel Access (HCCA), to pro-

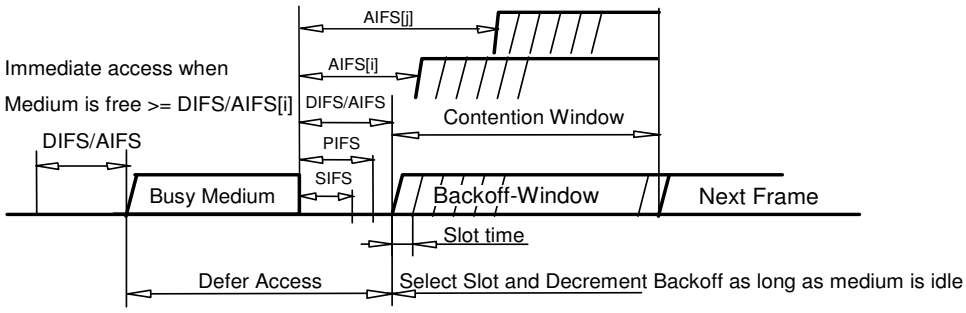


Figure 4.2: Enhanced Distributed Function Channel Access

vide guaranteed QoS. Figure 4.1 illustrates the channel access scheme used by 802.11e HCCA. Similar to the legacy Point Coordination Function (PCF), HCCA uses a polling-based mechanism, where the AP controls the medium access. The main difference between the legacy PCF and HCCA is the flexibility of when the contention-free period can occur. The AP can begin a contention-free HCCA period if the medium has remained idle for a PCF interframe space period, which is shorter than the minimum AIFS.

During the HCCA contention-free period, the AP polls nodes for a fixed time duration, called *TXOP*, which is computed based on reservation information periodically sent to the AP by each of the flows. The *TXOP* for each node is initiated by a poll request from the AP and during this period, transmissions can occur in both the uplink and downlink directions. The periodicity, called service interval *SI*, is decided based on the minimum delay requirements for all nodes present in the network. This period allows for multiple contention-free transmissions and ends if one of the following conditions occur: neither the AP nor the node have any packets left to transmit, the channel idle time has exceeded the timeout period, or the time period expires. Note that the *TXOP* used in HCCA differs from that used in EDCA and is determined by the AP and calculated based on the flow requirements. The use of a fixed duration allows the AP to limit the time allocated to each node and is bounded by the default variable *dot11-DefaultCPTXOPlimit*.

4.2.3 Hybrid Coordination Function Scheduling

Having briefly described the distributed and centralized channel access mechanisms, we now describe a reference scheduler [95] presented by the IEEE 802.11e Working Group. Nodes with strict QoS requirements send reservation requests containing flow information, such as mean application data rate (ρ), mean packet size (L), maximum MSDU size (M), delay bound (D), and minimum physical data rate (R).

Using this reservation request, the scheduling policy decides the periodicity and the duration of the polls. The AP determines the minimum service interval (*SI*) to be used for all of the nodes, where the *SI* is the time duration between successive polls for the node. The selected *SI* is the highest sub-multiple of the 802.11e beacon interval duration that satisfies the delay requirements of each flow; *i.e.* the selected *SI* should be less than the minimum of required service intervals of all flows. After deciding on the *SI* for the flows, the AP also allocates a fixed *TXOP* to each flow depending on the mean application data rate as follows.

$$\begin{aligned} REQ_i &= \frac{\rho_i * SI}{M_i} \\ TXOP_i &= REQ_i * (\frac{M_i}{R} + O) \end{aligned} \tag{4.1}$$

where O is the overhead due to PHY and MAC headers, IFS, acknowledgment frames, and poll frames.

The maximum time spent in HCCA for each *SI* is limited by the *dot11-CAPMax* variable, and the total controlled access time in a beacon interval is limited by *dot11-CAPRate*. The above two variables limit the duration of controlled access period and bound the effect of controlled access mode on traffic flows in contention access mode. If the introduction of a new flow violates any of the above two requirements, the AP does not admit additional requests.

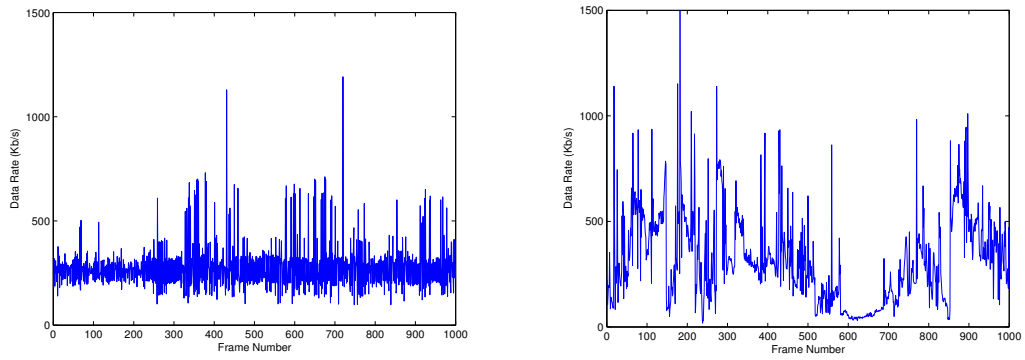
4.3 Motivation

Having briefly summarized the enhancements provided by 802.11e, in this section, we describe the limitations of using a fixed allocation policy, such as the one used in the reference scheduler, and motivate the need for new techniques to support real-time flow requirements. First, we illustrate the need for supporting multimedia streams with variable and unpredictable flow requirements. Next, we present a mathematical analysis of the delay impact on flows with variable bitrate traffic using current scheduling policy. Finally we describe the limitations of the current approaches.

4.3.1 Variable Requirements of Real-Time Flows

In order to provide support for diverse real-time multimedia applications in WLAN, it is important to understand the service requirements and needs of these applications. In this chapter, we focus our attention on video-based applications; however, the findings may be equally applied to other applications. In video applications, streams can be encoded with different objectives resulting in different bitrate requirements. One type of encoding, called Constant Bit Rate (CBR), tries to maintain the bandwidth requirements of encoded streams through appropriate adjustment of compression parameters. However, CBR encoding results in variable quality over time in order to maintain the constant bitrate. Another type of coding tries to maintain the quality of the encoded multimedia through use of constant compression parameters, which leads to variable bitrate requirements over time. Examples of common VBR encoding applications include quality-controlled MPEG4, video conferencing, video multi-casting, *etc.*. Service providers commonly use VBR encoding of multimedia content to increase the capacity of the network by multiplexing between different VBR flows [96] without impacting the multimedia quality.

We use an example video trace to illustrate the differences between CBR and VBR flows. Figures 4.3(a) and 4.3(b) show bitrate requirements over time of CBR and VBR encoding of a movie (Jurassic Parks with 25 frames/sec). It can be seen from the



(a) CBR (b) VBR

Figure 4.3: Throughput Requirement Comparison between CBR and VBR flows

figures that for a VBR-encoded video, the bitrate requirements can vary widely over time.

In addition to VBR encoding, application layer adaptation can lead to variations in bitrate requirements over time. For example, current streaming servers dynamically switch between different streams statically encoded at different target bitrates in order to cope with bandwidth fluctuations in the end-to-end network. This application layer adaptation results in dynamically changing flow requirements in a video stream. Having shown that the flow requirements can change over time, next, we analyze the effect of using the HCF reference scheduler with the variable traffic flows.

4.3.2 Analysis of HCF Reference Scheduling Policy

As described earlier, the reference scheduler allocates a fixed $TXOP$ for each flow based on mean data rate, and each flow is serviced in fixed service intervals. Although this scheduling is well suited for CBR traffic, the queues of HCCA flows with VBR traffic can build up when the requirement exceeds the mean value and eventually lead to large delays and dropped packets.

In order to understand the limitations of the reference scheduler for VBR traffic, we evaluate the delay of a traffic flow due to its residual queue length using the reference

scheduler. We model the VBR traffic data rate as a Gaussian curve probability density function with a mean and standard deviation of μ and σ respectively. The use of a Gaussian curve to represent VBR is motivated from previous studies [98] [99].

Let x denote the queue length value in terms of number of packets and i be the current service interval. We define the probability distribution function of packet arrival and residual queue length for service interval i by $IN_i(x)$ and $RES_i(x)$ respectively.

$$IN_i(x) = (1/\sqrt{2\pi\sigma^2}) * e^{(-(x-\mu)^2)/2\sigma^2)} \tag{4.2}$$

The residual queue length after service interval i is a function of the residual queue length in the last service interval, incoming packets and packets scheduled in the current service interval, as expressed below.

$$RES_i(x) = f(RES_{i-1}(x) + IN_i(x), SCH_i(x)) \tag{4.3}$$

where $SCH_i(x)$ represent number of packets scheduled by the reference scheduler in service interval i.

Using the reference scheduler, the number of packets serviced per *SI* is fixed and is based on the $TXOP$ given by the AP. We assume that ρ is the number of packets serviced per *SI, i.e.*

$$\forall i \quad SCH_i(x) = impulse(\rho) \tag{4.4}$$

Hence, the residual queue length is given by the following equation.

$$RES_i(x) = \begin{cases} 0 & \text{if } \; RES_{i-1}(x) + IN_i(x) < \rho \\ RES_{i-1}(x) + IN_i(x) - \rho & \text{otherwise} \end{cases}$$

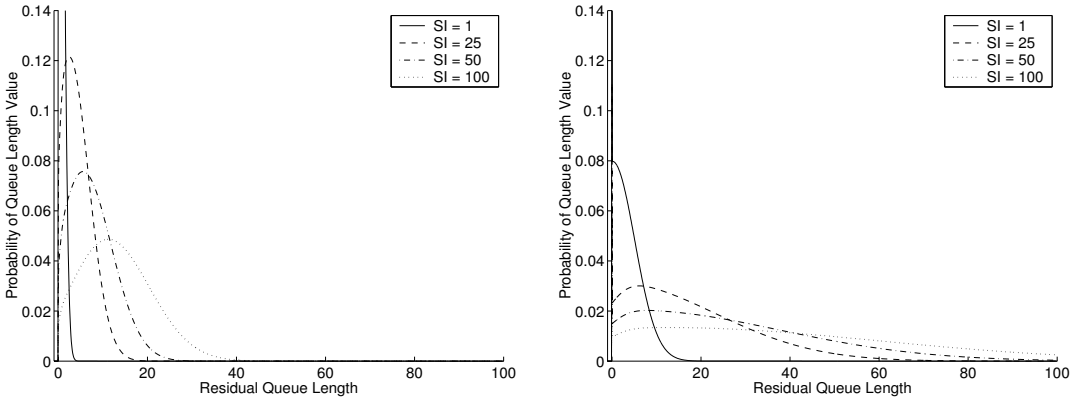
Note that the probability density function of the residual queue length is a discontinuous function at zero. Also, the probability of having no packets at the end of a service interval is equal to the probability that the packets needing to be serviced, given

by the incoming packets and the residual queue length of the previous service interval, is less than the number of packets serviced per service interval, ρ.

Since the closed form of determining the probability curves of the above equations after a fixed number of service intervals is difficult due to the discontinuous function for the residual queue length, $RES_i(x)$, we evaluated these equations using discontinuous probabilistic analysis. In order to understand how the residual queue length of a node can vary after a fixed number of service intervals, we used *Matlab* to input the Gaussian probability density function with the service rate of the reference scheduler and had the program output the probability density function of the residual queue length after a fixed number of service intervals.

We evaluated these equations numerically considering two Gaussian inputs, (μ, σ): [(20, 5), (20, 1)], to find the probability density function of the residual queue length of the flow. Since the 802.11e reference scheduler provisions for the mean value, for this evaluation, we consider a service rate equal to the mean, $\rho = \mu$. Figures 4.4(a) and 4.4(b) illustrate the probability density functions of the residual queue length after $(1, 25, 50, 100)\ SIs$ for the two Gaussian input curves respectively. Note that the curves indicate that the expected values of the queue length after $100\ SI$ are not zero. In fact, the expected values of the queue length for the two Gaussian inputs are 41 packets and 17 packets respectively.

With an increase in the residual queue length, the delay of each packet increases and can potentially be larger than the tolerable delay limit. This high delay leads to degraded video playback, such as stalls and blurs, depending on the transport protocol used. For instance, when TCP is used, the delay in packets causes stalls due to its reliable packet delivery mechanism. However, when UDP is used, the default packet loss concealment technique uses the previous frame if the packet does not arrive in time. The concealment method can lead to degraded video playback. Next, we discuss alternative techniques in addressing variability due to VBR encoding and their limitations.



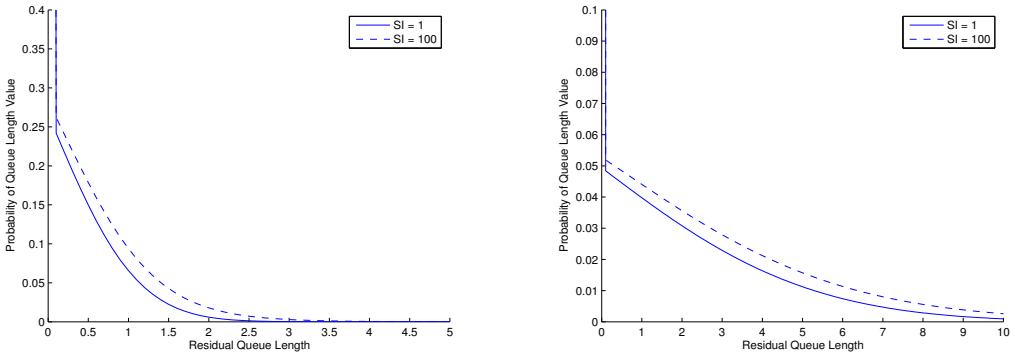
(a) $\rho = 20$ and $\sigma = 1$ (b) $\rho = 20$ and $\sigma = 5$

Figure 4.4: PDF Plot of Residual Queue Length with Incoming Gaussian Input

4.3.3 Limitations of Current Approaches

A possible solution to reducing the experienced delay and residual queue length buildup shown above is to over-provision for each flow and allocate additional time for every service interval. One example would be to use both the mean and standard deviation parameters in determining the number of packets serviced per service interval. Figure 4.5(a) and 4.5(b) illustrate the PDF of residual queue lengths of the same Gaussian inputs with a modified ρ given by $\mu + \sigma$. The expected value of the residual queue length decreases significantly. The above example illustrates that if it is possible to allocate more time for each flow, a queue buildup can be avoided. However, there is a tradeoff between the number of flows that can be serviced the AP and the amount of time allocated to each flow. In order to satisfy the VBR requirements without reducing capacity, there is a need to allocate additional time to VBR flows dynamically.

Additionally, in the reference scheduling policy, there is another limitation. The AP maintains a clear separation between the centralized HCCA and distributed EDCA periods. According to the separate resource allocation schemes, the real-time flows are not allowed to transmit using the contention based mechanism even when the load in EDCA period is low. This leads to a poor utilization of channel resources during the



(a) $\rho = 20$ and $\sigma = 1$ (b) $\rho = 20$ and $\sigma = 5$

Figure 4.5: Effect of Over-provisioning for Traffic Flow Variation

contention period, and reduces capacity and multimedia quality in HCCA flows because of the limitations imposed by the $dot11CAPRate$ and $dot11CAPMax$.

An alternative to over-provisioning is to dynamically allocate time based on monitored information. One proposed dynamic algorithm can be found in [98]. In this work, authors observe similar limitations of the current scheduler for VBR traffic and propose Flexible HCF (FHCF) framework to address some of the above limitations. Briefly, the FHCF algorithm monitors queue length and expands or shrinks the *TXOP* of each flow based on this information. In the case where the additional time is limited, the algorithm divides the time between the flows in a fair manner. However, there are a few limitations with this approach: (1) Fair division of allocated time to each of the flows may be insufficient in providing relief for residual queue buildup, (2) Aggressive adaptation to adjust TXOP may lead to further delays in cases where traffic variability is high, and (3) Frequent adjustment of TXOP may interfere (high jitter) with other nodes scheduled in the HCCA period. We present a comparison of our proposed technique with FHCF in Section 4.5.4, and further describe these limitations. Having discussed these various approaches and some of their limitations, we next describe our proposed dynamic adaptation framework to address application variability.

4.4 Adaptation Policy

We now propose a dynamic adaptation framework to improve Quality of Service (QoS) for flows with variable requirements and address limitations described in the previous section. We begin by describing the goal of the system, summarizing the overall adaptation approach, and finally, discussing the details of the adaptation algorithm and framework.

4.4.1 Problem Statement and Goal

To improve the QoS for multimedia flows, the goal of the adaptation is to minimize the aggregated utility value of the experienced packet delay. We can express this objective by the following equation,

$$\min\left(U_{HCCA}\right) = \min\left(\sum_{j=1}^{N} U_j(d_j)\right) \tag{4.5}$$

where j represents the node index, d_j is the average packet delay experienced by node j, and N is the total number of nodes.

Since the experienced packet delay for a flow is a function of the residual packet queue size, we specifically look at minimizing the utility of the aggregated residual queue size in each service interval.

$$\min\left(U_{HCCA}\right) = \min\left(\sum_{j=1}^{N} U_j(res_j)\right) \tag{4.6}$$

where res_j represents average residual packet queue size for node j.

The overall approach of our algorithm is to minimize the residual queue size by dynamically configuring the polling schedule in the HCCA period, called *HCCA Allocation*, and selectively allowing real-time flows to transmit in the EDCA period if needed, called *EDCA Mapping*.

HCCA Allocation

The goal of HCCA allocation is to determine a polling schedule so as to meet the objective expressed by Equation 4.6. The polling schedule can be expressed as [(PU_1,$PTXOP_1$),... ,(PU_l,$PTXOP_l$),... (PU_T,$PTXOP_T$)], where PU_l and $PTXOP_l$ represent the index of the node being polled and the TXOP allocated for the l^{th} poll respectively, and T is the maximum number of polls. The mean time requested for each flow j is denoted as $TXREQ_j$, and the time allocated for each flow in a service interval, called $TXOP_j$ can be expressed as

$$TXOP_j = \sum_{\forall(l) where PU_l=j}^{T} PTXOP_l \tag{4.7}$$

We assume that the polling schedule solution should meet the constraints inherent to the reference scheduler; (1) the total time allocated must be less than the total time available in each service interval, (2) each node must be polled at least once during the HCCA time period and polling times are non-overlapping, and (3) the time allocated to each node must be greater than or equal to the flow's requested mean requirements. The above constraints can be expressed by the following equations.

$$\sum_{j=1}^{N} TXOP_j < MAX\_POLL\_PERIOD \tag{4.8}$$

$$\forall(j)\ \exists(PU_l)\ s.t.\ PU_l = j \tag{4.9}$$

$$TXOP_j \geq TXREQ_j \tag{4.10}$$

In the above equations, MAX_POLL_PERIOD denotes the maximum polling duration available in each service interval determined by the *dot11-CAPMax* and *dot11-CAPRate* settings.

EDCA Mapping

In addition to allocating available time in HCCA period, we consider the case when there is no time available for reallocation in the HCCA period. The current HCF

framework assumes that the real-time flows will only send packets in their allocated time. However, by selectively allowing some flows into the EDCA period, we attempt to decrease the queue buildup when increasing $TXOP_j$ is not possible in the HCCA period.

The objective of *EDCA Mapping* is to select the flows, expressed by S, that can benefit from additional time in EDCA with the constraint that the impact on the EDCA flows due to the additional real-time flow is minimal. We need to understand the tradeoff between the benefits of HCCA delay reduction and negative impact on EDCA flows. This goal is to select the set can be expressed by the following expression:

$$\max\left(U_{SYS}\right) = \max\left(U(\sum_{m\in S} B_m, \sum_{m\in S} C_m)\right) \tag{4.11}$$

where B_m represents the total benefit in terms of the reduction in delay for node m, and C_m represents the cost of mapping node m in EDCA. Mainly, C_m is the impact on EDCA flows that can be given in terms of throughput, congestion, and increased collision.

4.4.2 Overall Approach

Having described the goals of our adaptation, we now describe our overall approach and state our assumptions. First, we assume that the AP uses the reference scheduler to determine the initial polling schedule. With this scheme, the AP receives reservation requests, schedules the appropriate transmission opportunities to each node in the network, and polls all nodes using a fixed service interval, as defined by [95] and described in Section 4.2.

Hence for each flow j, the following parameters are determined: the original service time based on the reservation request, $TXREQ_j$, and a polling slot PU_l for the node j. Note that using the original scheduler $TXOP_j = TXREQ_j = PTXOP_l$, meaning the total transmission opportunity allocated to node j is equal to the mean service time $TXREQ_j$ and $PU_l = j$. Additionally, each flow is limited to one poll

during one service interval. In other words,

$$\sum_{\forall(l) where PU_l=j} PU_l = 1 \tag{4.12}$$

With these assumptions, the ideal queue size for flows scheduled in HCCA after each service interval, *SI*, is zero. We assume that flows that have been scheduled in HCCA only send packets in the polling-based period and that the remaining flows without reservations are restricted to sending in the EDCA period. In addition, all flows provide queue size information to the AP using the control field of the 802.11e packet header. Finally, a real-time flow can be notified of the possibility of transmitting in the EDCA period.

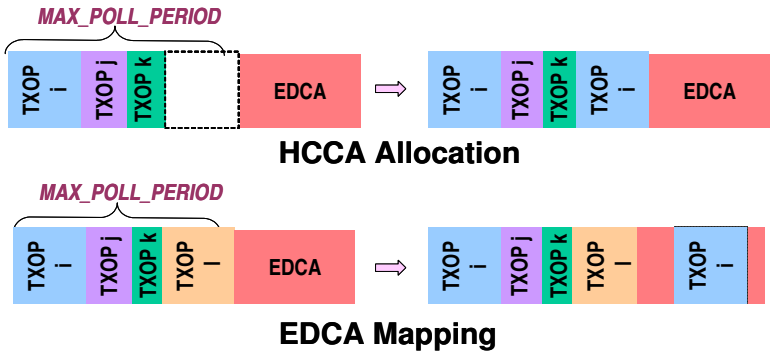


Figure 4.6: Illustration of Adaptation Algorithm: HCCA Allocation and EDCA Mapping

We next discuss our algorithm in greater detail. Figure 4.6 illustrates the overall policy of our adaptation. The top figure illustrates the adaptation algorithm for HCCA allocation. Without adaptation, flow i, j, and k are given the appropriate $TXOP$ values. However, flow i is in greater need of more time and since time is available in HCCA, extra time is allocated to flow i with adaptation. The bottom figure of Figure 4.6 summarizes the EDCA mapping algorithm. In this figure, there are four flows i, j, k, and l and node i is again in greater need of more time. Unlike the previous case, there is no additional time available in the HCCA period, which is limited by

the MAX_POLL_PERIOD. Hence, our algorithm encourages flow i to send in the EDCA period depending on the estimated load in the EDCA period. To provide more details on how we modify channel access privileges and allocate additional time for flows, we answer the following questions:

1. *How do we recognize which flows need to be adapted and select between flows in a fair manner?*

2. *How should we perform HCCA allocation so as to minimize the effects on other flows in the network?*

3. *What are the acceptable conditions to run the EDCA mapping adaptation and when should it be run?*

Determining Flows to be Adapted

We begin with describing how flows are selected for adaptation in the proposed algorithm. Briefly, the flows are sorted based on an adaptation priority, ap_j, which is calculated based on the following metrics; a flow's current need for time allocation, estimated variability, delay tolerance, and last adaptation time. In order to insure fairness, the algorithm uses a weighted round-robin selection scheme based on these priorities, ap_j, to determine which flow will be selected for adaptation.

We now describe the different factors used to calculate the adaptation priority, ap_j. The first metric that is considered is a flow's current need for time allocation. Mainly, the algorithm wants to understand how each node is performing given their original transmission allocation. This metric is given by the flow's utility value, which calculates the number of service intervals needed for a flow to empty its current queue size. Using the flow's residual queue size, res_j, original time allocation, $TXREQ_j$, and physical data rate, R_j, one definition of the utility function can be given with the following equation,

$$u_j = (res_j/(R_j * TXREQ_j))^2 \tag{4.13}$$

To meet the end goal of minimizing overall delay described in Section 4.4.1, the algorithm must serve the nodes that are in greater need of resources and additional time allocation. One approach in determining a node's adaptation priority would be to base it solely on the impact of the adaptation on reducing its current utility value, *i.e.* $ap_j = \Delta u_j$. With this approach, the algorithm would provide additional time to the flow with the highest potential delay. However, this may be the incorrect prioritization if we consider the possibility that a flow j is experiencing delays not based on its variability but because of other reasons, such as channel conditions or incorrect reservation request. With $ap_j = \Delta u_j$, we may be allocating additional time to a flow whose requirements cannot be met by occasional time allocation and preventing other variable nodes in need from being adapted.

The next metric we consider for ap_j is the estimated flow variability in order to avoid this problem. Mainly, the algorithm estimates the variability of the traffic flow to insure that the time allocation given by the adaptation will be beneficial for the node. The variability of a flow and the extent of its variability can be calculated using two methods. The first is dependent on the reservation request, where the variability of a flow is determined by directly using information about their variability sent in their reservation request. The second option is to estimate the residual queue variance using the monitored value of res_j over a time period. We compute the estimated residual queue variance weight, rqv_j ($0 \leq rqv_j \leq 1$), using the standard deviation, σ_{resj}, of res_j and comparing with the mean residual queue size, μ_{resj} over a fixed duration, as given in the following equation.

$$rqv_j = \min\left(\sigma_{resj}/\mu_{resj}, 1\right) \tag{4.14}$$

Note that larger standard deviation means a larger variance in the traffic profile.

Having explained how a_j incorporates the utility value, u_j, and estimated resid-

ual queue variance, rqv_j, we now motivate the use of the tolerable delay limit of a flow, D_j, as an additional metric. If we sort the nodes using a_j calculated solely based on u_j and rqv_j, we could potentially allocate time to a node that had the highest variability and the greatest need for time allocation, but actually had the highest tolerable delay limit. In this case, since it could have tolerated the experienced delay, the adaptation would give this node additional time unnecessarily. To address this, our third metric is the node's delay limit and we use the inverse ratio of a node's delay, $1/D_j$, to help prioritize flows that are less tolerable to delay.

We finally consider an additional factor in calculating a_j to make the adaptation fairer for all nodes. Let us consider a scenario where all the additional time is allocated to one HCCA flow because it has the highest variability, the greatest need for adaptation, and the lowest tolerable delay limit. In such a scenario, we risk the chance of an unfair allocation, where a node is able to monopolize the additional time available by adaptation algorithm. In order to prevent this, we added our final metric based on a node's last adaptation time, called the duration of last adaptation, dla_j. This duration is calculated using the current time and the last time the flow was last adapted. With this factor, the algorithm prioritizes nodes that have not been recently adapted.

Hence, using these four independent metrics and giving them equal weights, the adaptation priority of a node, a_j is determined by the following equation,

$$a_j = (dla_j * rqv_j * \Delta u_j)/D_j \tag{4.15}$$

These priority values are used to sort the flows and determine which node has higher priority to being selected by the adaptation in its weighted round-robin scheme.

HCCA Allocation

Upon deciding which flows need to be adapted, the algorithm needs to decide whether it is possible and beneficial to allocate additional time and modify channel access privileges. For HCCA allocation to be possible, there has to be sufficient time

remaining in HCCA for the flow to be polled, *i.e.*, after $TXOP_j$ has been adjusted, $\sum_{j=1}^{N} TXOP_j < MAX\_POLL\_PERIOD$ constraint should still hold.

In our scheme, as shown in Figure 4.6, we allocate additional time by re-polling the flow i after all previously scheduled flows have been polled. We maintain the original polling list and hence, between polls 1 to N, the nodes scheduled, *i.e.* PU_1 to PU_N, and the time allocated are left fixed. However, for the remaining polls, from $N+1$ to T, we select flows that have been identified as possible VBR flows and allocate the appropriate $TXOP_j$ for these flows.

We use this approach rather than extend the time duration for the flow in order to avoid unnecessary delays for other flows scheduled in HCCA. Recall that the FHCF scheme discussed earlier in Section 4.3 configures the fixed $TXOP_j$ of a node in a poll, by adjusting the $PTXOP_l$ for flow j starting at poll l, where $PU_l = j$. With this approach, other flows may be delayed due to additional time allocation to flows scheduled prior to them. By maintaining the original schedule, we avoid affecting other flows. Although there is an extra overhead of the additional poll, this overhead is minimal and only consists of a poll frame and a $SIFS$ period. Additionally, the FHCF algorithm attempts to take advantage of the "lag" of a flow by reducing the $TXOP_j$ of the node. Although this can provide for more allocation time, this reduction means that at some intervals, the allocated time is less than the requested time, *i.e.* $TXOP_j < TXREQ_j$. This may violate the assumption that the flow will have at least the mean requirement reserved for each service interval. Although the FHCF algorithm eventually adjusts for the additional time when needed, the change may lead to a constant oscillation in allocated time.

EDCA Mapping

In the case where additional allocation is not possible because of limited time in the HCCA period, another method used to minimize the residual queue size is to encourage flows to send during the EDCA period, as shown in Figure 4.6, where node i

is allowed to send in EDCA. Since EDCA uses a distributed channel access scheme, the adaptation does not guarantee a flow with fixed time in EDCA but provides a signal to guide a flow to compete for the channel in EDCA. One consideration of this adaptation is the negative impact, C_j, on the flows in EDCA due to the mapping of node j. If an HCCA flow is encouraged to send in EDCA when the network is congested, the traffic flow may suffer from retransmissions due to collision and cause negative effects on other traffic flows. Hence, to avoid such problems, the network load must be estimated by the AP. Once the load estimation is performed, the selection of flows to be mapped in the EDCA period is decided by the priorities described earlier in Section 4.4.2

In our approach, the network load is estimated using two statistics: *utilization ratio* and *collision ratio*. The AP calculates the utilization by monitoring the time used in EDCA, summing the transmission durations of successfully received packets over the total time available. Although the utilization ratio is a good indicator of the network load under low load conditions, the value saturates as the load increases due to collisions. Under such conditions, we additionally use a collision metric that is determined by averaging the number of retransmissions that occur over the EDCA period over the total number of packets sent. Since collisions can be caused by poor channel conditions, it cannot be solely used as a metric to determine load since the collision ratio may be skewed by transmission errors caused by a poor channel. Additionally, certain nodes may be experiencing poorer channel conditions than others. If needed, to counter this problem, the collision ratio can be adjusted using weights based on observed retransmissions for each unique EDCA flow. In other words, if the AP observes high retransmissions for a particular flow, the AP can decrease the impact of that flow's collisions on the overall collision ratio. However, for the sake of simplicity, we assume that the collision ratio is not impacted by poor channel conditions. Experiments on the *utilization ratio* and *collision ratio* can be found in Section 4.5.

In our proposed adaptation scheme, after a fixed number of SIs, we compare the current utilization and collision metrics with preset threshold values, determined based on monitored values, and decide on whether it is appropriate to map additional VBR

flows to the EDCA period. This is to minimize the impact on EDCA flows and avoid additional collisions experienced by the introduced VBR flow. Note that determining the actual cost of EDCA mapping for the utility value described in Section 4.4.1 is difficult prior to adaptation. However, since the cost of the adaptation is proportional to the estimated load in EDCA, *i.e.* the cost is high when the load is high, the load estimation is used before the adaptation as a heuristic function to decide when the adaptation is performed. After performing the EDCA mapping, we need a performance metric that allows us to understand the impact of the adaptation. One possible metric for evaluating the cost and benefit after the adaptation can be given by the following equation.

$$U(\sum_{m\in S} B_m, \sum_{m\in S} C_m) = (\sum_{j=1}^{N} \max(dl_j/d_j, 1))/2N + (\sum_{i=1}^{M} \max(t_i/tr_i, 1))/2M \quad (4.16)$$

Mainly, in this equation, we calculate the inverse ratio of the delay experienced, d_j, to the maximum delay limit, dl_j, for all of the HCCA flows, N, and determine the average inverse delay ratio of an HCCA flow. We next calculate the ratio of the throughput experienced, t_i, to the throughput required, tr_i, for all the EDCA flows, M, and determine the average throughput ratio of an EDCA flows. The utility value factors both the impact of the EDCA mapping on HCCA delay and the impact on EDCA throughput evenly.

4.4.3 Adaptation Algorithm and Framework

After having described our overall approach, we provide further details of the algorithm. The pseudocode of the overall adaptation is provided in Figure 4.7, with further details of the coordination summarized in Figure 4.8. The first step of the algorithm is to monitor the current network conditions. In this step, we need to determine the estimated load in EDCA. As described earlier, we do this by calculating the utilization and collision ratios. The AP then proceeds with the original polling schedule and updates the queue statistics accordingly. After having finished polling the original schedule, the

algorithm then allocates additional time in the HCCA scheduling period. By calculating the available time in HCCA (*hcca_slack*), the algorithm determines iteratively if there is available time to schedule in HCCA. The selection process of the flows is based on the weights described earlier in this section. We estimate the flow variability, the utility value given the residual queue size, data rate, and given time allocation, the maximum delay, and the last adaptation time. We next sort the flows according to these factors. In the case where the HCCA period is unable to support additional polling and there are remaining HCCA flows in need of additional scheduling, the AP explicitly signals to the selected real-time flow to send during the EDCA period, depending on the estimated load.

```
/* Performed every service interval */
AP_schedule() {
  /*Monitor network conditions*/
  SCHED_EDCA_OK = monitor();
  /*Perform original polling schedule and update stats*/
  original_schedule_poll();
  /*Decide if additional nodes need to be polled*/
  coordinate_hcca_edca(SCHED_EDCA_OK);
}
```

Figure 4.7: Overall Adaptation Pseudocode for Improved Time Allocation for Real-Time Applications

```
/*Evaluates hcca and edca to determine nodes to adapt*/
coordinate_hcca_edca(SCHED_EDCA_OK) {
  /*Determine leftover space, assume hcca_slack*/
  if (hcca_slack >= avgTXOP) {
    /*First deal with hcca and sort according to weights*/
    /*Returns list of flows needing to be scheduled*/
    sorted_hcca_list = sort_priorities(hcca_flows);
    /*We break if there is no more space and/or
      all the nodes have been satisfied*/
    while (hcca_slack > avgTXOP &&
        sorted_hcca_list != NULL) {
      /*Adds next node in sorted_hcca_list for polling*/
      schedule_hcca_flow(sorted_hcca_list);
      update_hcca_slack();
    }
  }
  /*We check to see if all hcca nodes are finished*/
  if (sorted_hcca_list != NULL) {
    /*Based on load, signal additional flows for EDCA*/
    if (SCHED_EDCA_OK)
      signal_EDCA(sorted_hcca_list);
  }
}
```

Figure 4.8: HCCA Allocation and EDCA Mapping Adaptation Pseudocode

Finally, in order to run the described dynamic adaptation, we implement an adaptation framework consisting of two main functional components: *monitor* and *adaptor*, as shown in Figure 4.9. The monitor tracks the current conditions in both the HCCA and EDCA periods. Additionally, it provides the adaptor with the flow weights and load information of the two periods, through a number of tracked statistics. Using these values and information given by the monitor, the adaptor decides which flows can benefit from the adaptation, and when and how the flows can be nodes can be provided additional time.

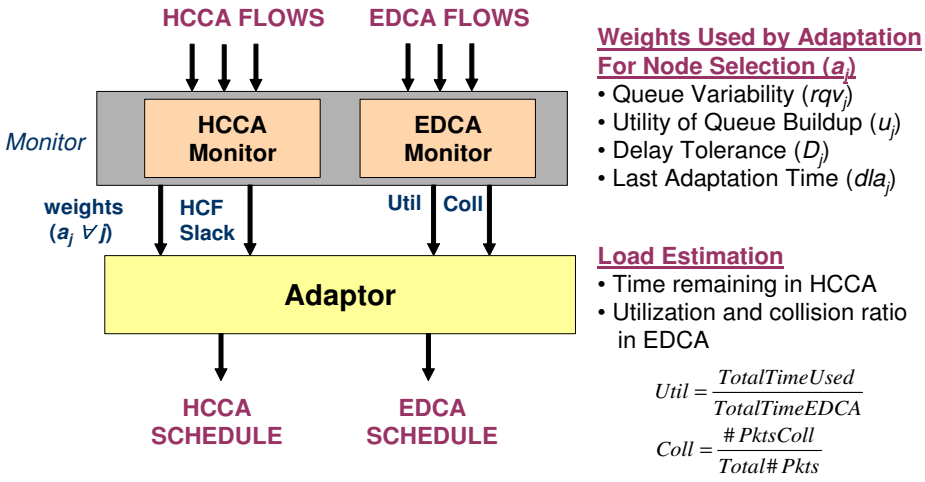


Figure 4.9: Scheduling and Resource Allocation Adaptation Framework

Having described the adaptation algorithm, in the next section, we evaluate the performance of the proposed adaptation framework and compare with other known techniques.

4.5 Experimental Results

In this section, we present the experimental evaluation of our adaptation framework over a diverse set of multimedia traffic profiles and network load. We begin by briefly describing the system configuration. Next, we evaluate the improvement in performance due to HCCA flow reallocation in both the HCCA and EDCA periods. We

then present a comparison of the framework with the Flexible HCF (FHCF) mechanism, proposed in [98] and finally, we describe the effect of the adaptations on the quality of the video and overall system capacity.

4.5.1 System Configuration

To understand the experimental results, in this subsection, we describe the simulation setup and system parameters, as well as 802.11e parameters. Next, we describe the application profiles, summarizing both the profiles of model-based modeling and trace-based modeling of multimedia applications. Finally, we summarize the performance metric used in our experiments for both HCCA allocation and EDCA mapping.

Simulation Setup

For our experiments, we used the *Opnet* simulation environment [84]. We modified the IEEE 802.11b MAC layer models to incorporate HCCA and EDCA service differentiation, as presented in the IEEE 802.11e standard. We have validated our implementation of 802.11e with other known models and the default characteristics of the simulated wireless network are listed in Table 4.1.

Table 4.1: Simulation Parameter Settings

Parameter	Value
PHY Rate	11 Mb/s
Beacon Interval	100 ms
Fragmentation Threshold	1024 bytes
RTS Threshold	1024 bytes
SIFS	10 us
DIFS	50 us
PIFS	30 us
SlotTime	20 us

In our network configuration, we assume that each node is associated with only one flow and remains at a fixed location that is distributed uniformly in the area of 300m

x 300m, *i.e.* the range of an access point. The number of nodes in each of the simulation runs varies and will be listed in the individual result descriptions. For simplicity, we have assumed in our simulations that each node is experiencing similar network conditions, leading to the same PHY rate. In the experiments presented in this chapter, the physical rate is set to 11 Mbps. We believe that the observations and results would be similar under different physical data rates.

In addition to the link access parameters, there were additional parameters that were used for the 802.11e enhancements in HCCA and EDCA. Some of the parameters for EDCA include the contention window range, given by CW_{min} and CW_{max}, the transmission opportunity per node, TXOP, and the Arbitration Interframe Spacing, AIFS. The main parameters for HCCA include specifying the fraction of the Contention-free Access Period (CAP) using the dot11-CAPRate, as well as determining the maximum CAP duration and default TXOP for flows with the dot11-CAPMax and dot11-DefaultCPTXOPLimit parameters respectively. The above parameters and their values are listed in Table 4.2. The EDCA parameters are listed for four access categories, namely background, best effort, video, and voice traffic.

Table 4.2: EDCA and HCCA Parameter Settings

EDCA Parameters	Access Category Values			
	AC_BK	AC_BE	AC_VI	AC_VO
TXOP (us)	0	0	6016	3264
AIFS	7	3	2	2
CW_{min}	31	31	15	7
CW_{max}	1023	1023	31	15
HCCA Parameters	Value			
dot11-DefaultCPTXOPLimit	2000 us			
dot11-CAPRate	21 us			
dot11-CAPMax	8000 us			

For our experiments, we assume that all the nodes that access the channel during EDCA belong to the best effort (AC_BE) access category. Note that the choice of AC for the EDCA nodes in these simulations does not impact the time allocation for the

nodes adapted in HCCA, which is the main focus of the proposed adaptation and does not negatively impact the results of the experiments. Additionally, we maintain fixed values for the EDCA parameters in our experiments.

Application Profiles

Having described the overall network configuration, we now describe two types of application profiles used in our experiments, namely model-based profiles and trace-based profiles. Model-based application profiles are generated using a distribution with a given mean data rate and a standard deviation. By changing the standard deviation, we provide a method of changing the variability of the application traffic to model Variable Bit Rate (VBR)-encoded traffic. For most of the model-based experiments, we looked at a normal distribution for the application traffic, which has been used in previous studies, to represent VBR traffic. We also evaluated the performance of the reference scheduler and proposed adaptation in the case of a uniform distribution and generate the application traffic, using the Opnet application and profile configuration to set the packet size and inter-packet arrival time. Table 4.3 summarizes the various model-based application profiles and characteristics used in our experiments. For each profile, we present the distribution used to generate the profile, mean bandwidth requirement, and standard deviation of bandwidth. Note the table includes the profile used for EDCA flows in all the scenarios that consists of a fixed data rate of 256 kb/s.

Table 4.3: Model-Based Application Profiles

Profile Name	Distribution Type	Data Rate (kb/s)	
		Mean	Standard Deviation
N1	Normal	300	60
N2	Normal	300	100
N3	Normal	300	180
U1	Uniform	300	60
U2	Uniform	300	100
U3	Uniform	300	180
EDCA	Constant	256	–

The second type of profile used in our experiments is collected from real application traces. In order to evaluate how the proposed adaptation performs on actual traces of VBR and CBR encoded sources, we incorporated various packet traces of H.263 video streams collected from [44], using the actual packet sizes and interframe packet time of the video in the simulations. We summarize the various application profiles used in our evaluation in Table 4.4 and for each profile, we present the data rate requirement (mean and standard deviation), mean packet size, and delay requirements.

Table 4.4: Trace-Based Application Profiles and Characteristics

Profile Name	Data Rate (kb/s)		Mean Packet Size (bytes)	Delay Required (ms)
	Mean	Standard Deviation		
Video1_VBR	445	282	4000	60
Video1_CBR	445	20	4450	60
Video2_CBR	256	26	1024	60
Video3_VBR	340	90	1700	60
EDCA	256	-	512	-

As seen in the above table, we looked at mainly three video sources, Video1-3. Video1 is the video trace of Jurassic Park at 25 frames/sec, encoded using Variable Bit Rate and Constant Bit Rate. The difference in the trace properties can be seen in the table above in the standard deviation. In one of our experiments, we partition Video1 into various segments and evaluate the adaptation over these sequences. Details about these particular sequences will be provided in the later sections. Video2 is primarily a CBR traffic flow that is used as additional load scheduled in HCCA in some of the experiments. For Video3, the application profile is of a shorter video sequence, called *GlobalMedia*, used in the application quality assessment in Section 4.5.5. Each of the video profiles has a maximum tolerable delay requirement of 60 ms.

Performance Metrics

In this subsection, we briefly summarize the performance metrics used in our experimental results. Primarily, for the real-time application traffic, we use the average

packet delay as the metric of evaluation, since the packet delay can directly impact the QoS experienced. In some of the experiments, we plot the cumulative distribution function (CDF) of the packet delay in order to present the statistical distribution of the experienced delay. For nodes with EDCA traffic, the performance metric for these nodes is average packet throughput. Specifically, we present the percentage throughput degradation of EDCA throughput when EDCA mapping occurs. Finally, in simulations where we are evaluating the impact of the adaptation on the node being given additional time, as well as the nodes using EDCA, we use the utility function described in Section 4.4.2 that combines the ratio of required throughput to experienced throughput of all the EDCA flows and the ratio of required delay of the HCCA flow to the experienced delay.

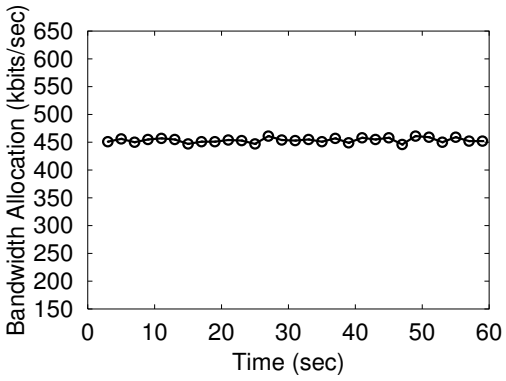
4.5.2 Evaluation of HCCA Allocation

In this subsection, we perform four groups of experiments to show the effectiveness of our HCCA allocation. We start with understanding the limitations of the reference scheduler and looking at the impact of the proposed adaptation on one VBR node. Next, we evaluate the impact of variability on the experienced delay and the adaptation using model-based application profiles. After looking at model-based profiles, we focus our attention on evaluating four actual trace-based application profiles and understanding the effectiveness of the adaptation. Finally, we look at the impact of multiple flows for one model-based and one trace-based application profile.

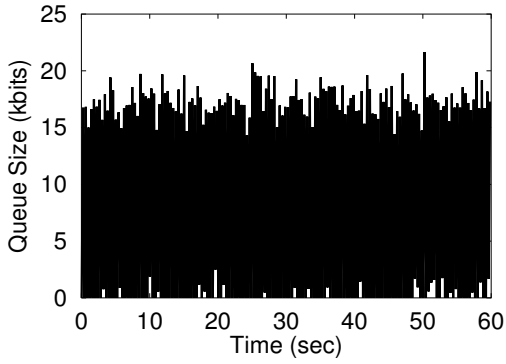
Impact of HCCA Allocation on VBR Flow Performance

Before evaluating our proposed adaptation, we begin by illustrating the limitations of the reference scheduler for VBR traffic. For comparing the performance of the algorithms, we investigate the impact of scheduling policies in a scenario consisting of one node with the Video1 application trace. We assume the node has been allocated its mean data rate of 445 kb/s by the reference scheduler and observe how the reference scheduler performs with the CBR application profile (Video1_CBR) shown in Figure

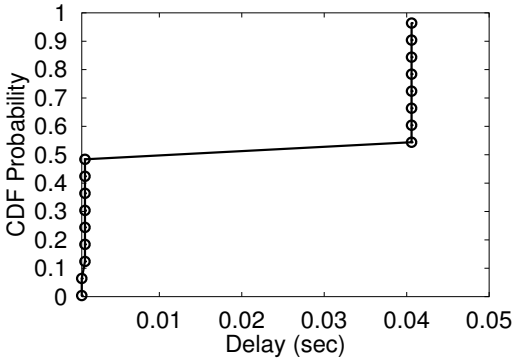
4.3(a) and VBR application profile (Video1_VBR) shown in Figure 4.3(b). Finally, we look at the potential of using the proposed adaptation on the VBR profile. For each scenario, we show the bandwidth allocation given to each node (in kb/s), the node's residual queue length (in packets), and the CDF plot of the delay experienced over the duration of the simulation.



(a) Allocation



(b) Residual Queue Size

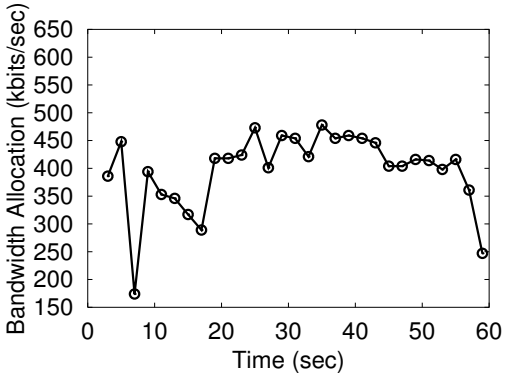


(c) Experienced Delay

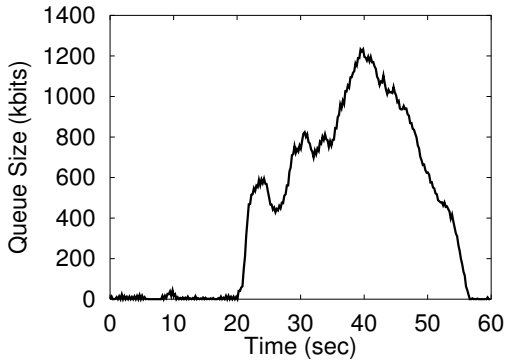
Figure 4.10: CBR Flow Performance with Reference Scheduler

We begin with the Video1_CBR application trace, which has a mean of 445 kb/s and a standard deviation of 20 kb/s. Using the reference scheduler, time is allocated based on the mean data rate requested by the node. Figure 4.10(a) illustrates the bandwidth allocation and shows that the node is given 445 kb/s based on its requirements. With this allocation, the CBR flow does not experience any residual queue buildup and

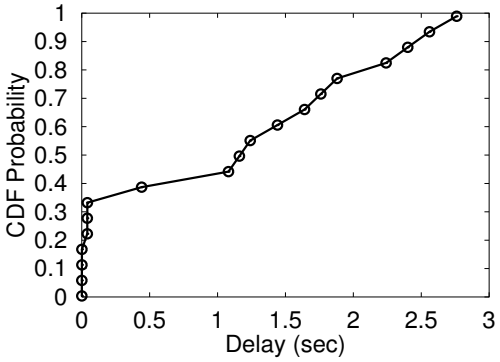
the queue size remains low with approximately one packet left in the queue (less than 20 Kb) throughout the simulation, as can be seen in Figure 4.10(b). Figure 4.10(c) shows the CDF plot of the delay experienced by the CBR flow given the reference scheduler's time allocation. Since the node does not experience high residual queue lengths, the delay experienced is low and well within the desired delay limit of 60 ms, as shown in the figure.



(a) Allocation



(b) Residual Queue Size

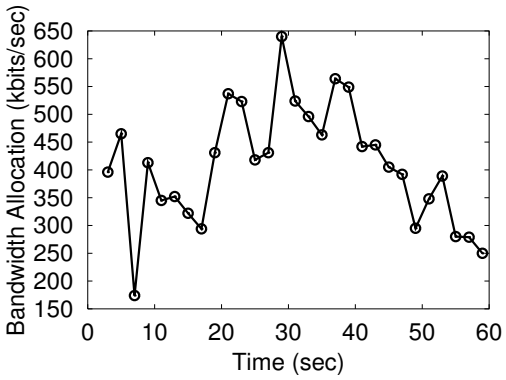


(c) Experienced Delay

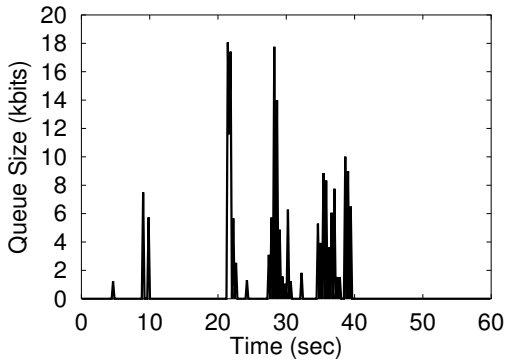
Figure 4.11: VBR Flow Performance with Reference Scheduler

Having looked at a CBR flow, we look at the impact of the same allocation on an application traffic flow encoded using VBR. Using the application trace of Video1_VBR, which has the same data rate as the CBR node (445 kb/s) but a higher standard deviation of 282 kb/s, we can see in Figure 4.11(a) that the bandwidth allocation given by

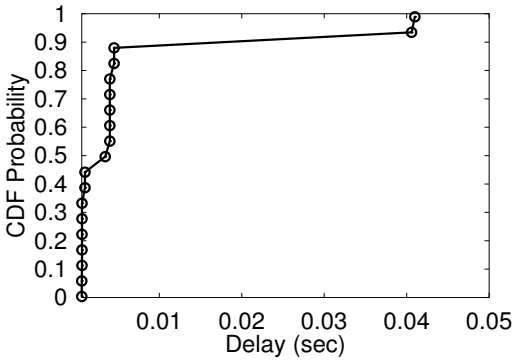
the reference scheduler varies over time but is limited to the mean data rate. With this allocation, we can see in Figure 4.11(b) that unlike in the CBR case, the node experiences a high residual queue buildup, indicating the need for additional time allocation. The time allocation given by the reference scheduler for Video1 is inadequate due to its higher variability. The consequence of this high queue buildup can be seen in CDF of the experienced delay, illustrated in Figure 4.11(c). Note that the delay experienced significantly exceeds the maximum tolerable delay of Video1, *i.e.* 60% of the packets experience delays higher than 60 ms.



(a) Allocation



(b) Residual Queue Size



(c) Experienced Delay

Figure 4.12: VBR Flow Performance with Proposed Adaptation

We next evaluate the performance of the proposed algorithm using dynamic time allocation of resources. The proposed adaptation allocates additional time appropriately

based on the monitored residual queue size. Figure 4.12(a) shows the additional allocation provided with the proposed adaptation mechanism. Note that through the reallocation of the remaining time of the HCCA period to the VBR traffic flows during critical periods (around 20 sec), the adaptation algorithm is able to relieve the queue and prevent a high buildup, *i.e.* maximum of 3 packets (queue size of 18 Kb) as opposed to 80 (queue size of 1.2 Mb) without adaptation. Figures 4.12(b) and 4.12(c) illustrate the queue size and experienced delay respectively using the algorithm's time allocation in HCCA. As can be seen from the graphs, the proposed algorithm can significantly reduce the delay experienced by the HCCA flow and hence, the algorithm can avoid exceeding the maximum delay and decrease the packet loss experienced by the flow. For instance, 90% of the packets transmitted fall within the tolerable delay limit.

Adaptation Performance Under Different VBR Variability

Having looked at the impact on one VBR application profile, we next evaluate our adaptation algorithm for VBR flows with different variability. For this evaluation, we look at a model-based application profile for one VBR flow and the effect of the algorithm given a specific variability and distribution type. We record the average delay experienced by one VBR flow over a period of three minutes, using the model-based application profiles, with the reference scheduler and our adaptation. For each profile, we performed five simulation runs and calculated the average delay over the five runs. We found the standard deviation for the mean delay across our sample runs with our adaptation was less than 5%. Table 4.5 shows the results of our evaluation for both the normal and uniform distribution profiles. For each video profile, we present the data rate requirement and the delay with and without adaptation.

The table indicates that with the reference scheduler, the experienced delays without adaptation increase as the standard deviation increases. For example, from N1 to N3, the delay increases from 209 to 467 ms and from U1 to U3, the delay increases from 159 to 286 ms. Hence, the reference scheduler becomes more limited in providing

Table 4.5: Effect of Video Variability on Adaptation

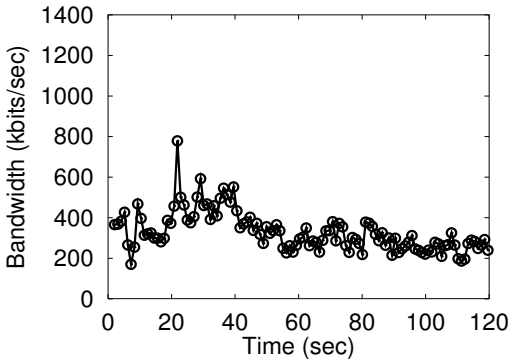
Profile Name	Data Rate (kb/s)		Delay (ms)	
	Mean	Standard Deviation	Without Adaptation	With Adaptation
N1	300	60	209	39
N2	300	90	385	39
N3	300	180	467	38
U1	300	60	159	42
U2	300	90	188	39
U3	300	180	286	39

adequate time allocation with greater variability. Additionally, from the table, it can be seen that the reference scheduler performs better with a uniform distribution of packets than a normal distribution. Mainly, the uniform distribution provides appropriate slack for the scheduler to better allocate time for the VBR flow. It is important to observe that the delay with the reference scheduler still exceeds the tolerated delay of 60 ms, even when the data rate variability follows a uniform distribution.

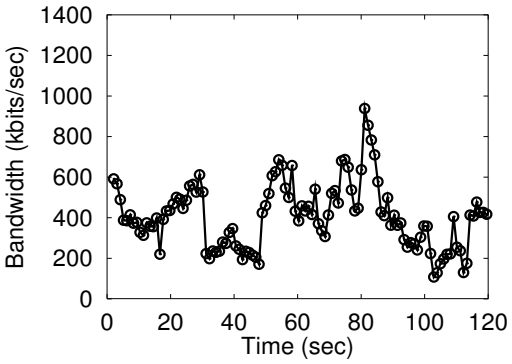
Using the proposed adaptation of HCCA allocation, we can see that for both the normal and uniform distribution, the adaptation algorithm is able to decrease the experienced delay significantly so that it falls within the delay bounds and remain about the same, around 40 ms, independent of the variance. By dynamically monitoring the queue size, the algorithm is able to handle the observed variance and maintain a tolerable packet delay.

Adaptation Performance Using Trace-Based Application Profiles

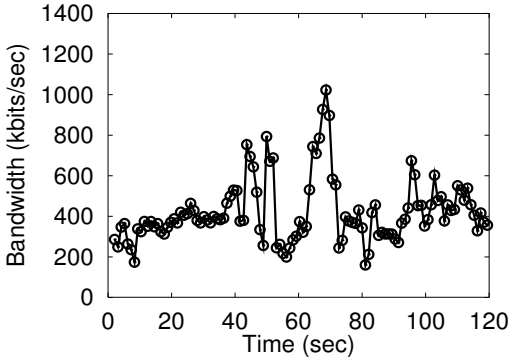
For the next set of experiments, to understand how the proposed algorithm works in the case of real application traces, we look at four different trace-based application profiles. In particular, we look at four two minute sequences of the Video1_VBR flow and show the data rate requirement for each of the profiles in Figures 4.13(a), 4.13(b), 4.13(c), and 4.13(d) over a two minute duration. Additionally, we provide summarized statistics (mean data rate and standard deviation) for each of the profiles in Table 4.6.



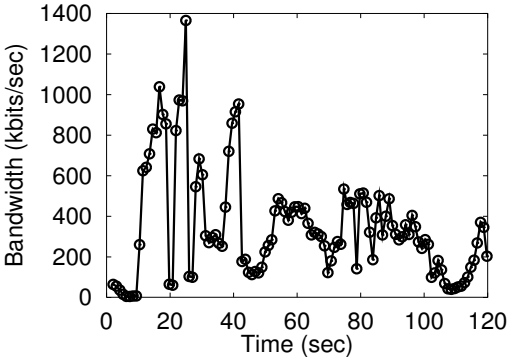
(a) Video1_1 Data Rate Profile (Mean-330 kb/s, Stdev-85 kb/s)



(b) Video1_2 Data Rate Profile (Mean-409 kb/s, Stdev-154 kb/s)



(c) Video1_3 Data Rate Profile(Mean-422 kb/s, Stdev-146 kb/s)



(d) Video1_4 Data Rate Profile (Mean-349 kb/s, Stdev-240 kb/s)

Figure 4.13: Trace-Based Application Profiles

Looking at the profiles, we can see that the bandwidth experienced in Video1_1 remains for the most part below the mean with occasional exceptions at around 25 and 40 seconds. Video1_2 and Video1_3 are highly varying with multiple periods where the bandwidth requirement is significantly higher than the requested data rate. The maximum requested bandwidth for Video1_2 is at around 900 kb/s near 80 seconds and for Video1_3, the maximum bandwidth is 1 Mb/s near 70 seconds. Finally, we look at Video1_4 profile, and observe a very high variance throughout the two minute duration. The standard deviation to mean data rate ratio is significantly higher with this profile compared to the other profiles and the maximum bandwidth required is near 1.4 Mb/s earlier at around 25 seconds.

Table 4.6: Experienced Delay of Trace-Based Application Profiles with Adaptation

Profile Name	Data Rate (kb/s)		Standard Deviation to Mean Ratio	Delay (ms)	
	Mean	Standard Deviation		Without Adaptation	With Adaptation
Video1_1	330	85	0.254	614	19
Video1_2	409	154	0.374	1854	38
Video1_3	422	146	0.346	1999	63
Video1_4	349	240	0.688	1558	121

Having looked closely at these four profiles, we now present the corresponding experienced delay with and without adaptation and also the standard deviation to mean ratio of each of the application flows in Table 4.6. We start by looking at the delay experienced without adaptation for the four video profiles and observe that for all profiles, the delay experienced is higher than the maximum tolerable delay. We also observe that the average delay without adaptation depends on the range of variability observed and temporal properties of variation, *i.e.* the time duration where the demand exceeds the mean requirements. For instance, Video1_1 has low variability with a standard deviation to mean ratio of 0.254 and therefore, it experiences the lowest delay with the reference scheduler allocation. We also can observe from the table that although Video1_2 and Video1_3 have similar variability, the average delay experienced by Video1_3 is slightly

higher than Video1_2. The difference between the experienced delay is caused by the higher application demand of Video1_3 than Video1_2. From the profile graphs of both videos, Video1_3 has more and higher peaks for its application demand than Video1_2. Finally, we look at the delay experienced by Video1_4 without adaptation. From the table, we see that although Video1_4 has the highest variability, the delay is not the highest. This can be explained by looking at the mean data rate over the simulated period. Since the average requirement of this flow is less than the mean allocation, the reference scheduler, despite the variation, is able to relieve the queue of excess packets when the flow demand is lower than the time allocated. In this scenario, the higher variation in Video1_4 in combination with the lower mean allow this video profile to experience less of a delay than Video1_2 and Video1_3.

Having seen that the desired delay cannot be met with the allocated time given by the reference scheduler, we describe the results obtained with adaptation. As can be seen in Table 4.6, there is a significant delay improvement achieved through the use of the adaptation. With adaptation, the trend of the delay is dependent on the range of the variability, temporal variability, and allocation restrictions. For instance, the adaptation provides the lowest delay, 19 ms, to Video1_1, which has the lowest variability and lowest average data rate demand. Similarly, Video1_2 is able to maintain a delay within the tolerable delay limit using the adaptation. For Video1_3, we observe a significant delay reduction but observe that the flow experiences an average delay higher than Video1_2 with the adaptation. This is mainly caused by the configured polling restrictions of the system, *i.e.* a limit on the additional time allocation per service interval. In this evaluation, we have assumed that at most one additional poll is allowed for each flow in a given service interval. Hence, in the periods where Video1_3 has the higher data rate requirements, our polling restriction limits the adaptation's ability to reduce the average delay below the tolerable delay limit. Similar behavior can be observed in Video1_4, where the observed delay is higher because of the extremely high requirements early in the simulation. We believe that the impact of the adaptation can be improved by allowing more than one poll per service interval and this parameter can be configured

appropriately to support traffic flows with widely varying requirements.

Adaptation Performance with Multiple VBR Nodes

Next, we look at the impact of the adaptation when there are multiple VBR flows. We consider two application profiles, first looking at a model-based application and then looking at a trace-based application. In the topology, we consider one CBR flow (Video2_CBR) scheduled in HCCA and vary the number of HCCA flows with VBR traffic from one to sixteen flows, using N2 (mean-300 kb/s, standard deviation-100 kb/s) for the model-based and Video1_VBR for the trace-based application. For these experiments, the objective is to see how the adaptation performs when there are multiple VBR nodes being adapted.

Table 4.7 presents the results of our evaluation in terms of the delay experience averaged over the model-based HCCA flows considered. Looking at the delay without adaptation, we can see that the performance remains similar as we increase the number of nodes from two to sixteen. This is expected because the reference scheduler provides guaranteed time allocation per service interval. Without the proposed adaptation, it can be seen that as the number of nodes increases, the average delay experienced by the nodes increases slightly. Although the adaptation is able to reduce the delay significantly, in the case where are many nodes, *i.e.* sixteen nodes, the experienced delay is higher due to lack of time in HCCA for additional allocation.

Table 4.7: Effect of Adaptation on Delay of Multiple HCCA Model-Based VBR Nodes

Number of VBR Flows	Delay (ms)	
	Without Adaptation	With Adaptation
2	234	39
4	236	39
8	238	40
12	239	41
16	246	54

In terms of a trace-based application profile, we use the Video1_VBR profile

and observe the improvement achievable with and without the adaptation. Table 4.8 presents the results of our evaluation. Without adaptation, the delay experienced by the VBR flow seems to vary slightly as the number of nodes increase. This may be due to simulation variation and possibly higher packet sizes, which may allow a flow to slightly exceed its TXOP and cause a small delay for other flows. The average delay between the five scenarios without adaptation is 836 ms. If we look at the performance with adaptation, we can see a slight increase in the delay with adaptation as the number of nodes increase, especially in the case of sixteen nodes. In addition to slight simulation variations, this may also be caused by the actual HCCA time allocation policy. The ability of the adaptation to reduce the experienced delay is dependent on the available time that can be allocated and hence, on the number of nodes in the system. With a higher number of nodes, *i.e.* sixteen nodes, the adaptation does not have sufficient time to allocate additional transmission opportunities to the VBR flows.

Table 4.8: Effect of Adaptation on Delay of Multiple HCCA Trace-Based VBR Nodes

Number of VBR Flows	Delay (ms)	
	Without Adaptation	With Adaptation
2	731	20
4	830	23
8	837	39
12	883	52
16	899	500

From these experiments on HCCA allocation, we have seen that the adaptation can improve the experienced delay in a number of cases, including in cases where there is higher variability, where actual application traces are used, and where there are multiple nodes using the adaptation. Next, we look at the second part of the EDCA Mapping algorithm.

4.5.3 Flow Reallocation in EDCA

In the case where the HCCA period has limited available time, the effectiveness of flow reallocation in HCCA is reduced to the case without adaptation. In this subsection, we look at the effectiveness of the second method in our proposed adaptation, *EDCA mapping*, when reallocation in HCCA is not possible and the load in EDCA can support additional flows.

Load Estimation using Utilization and Collision Ratio

As described in Section 4.4.2, EDCA mapping is allowed only when the monitored load is low in EDCA. In order to estimate the load in EDCA, we look at two values, the *utilization ratio* and the *collision ratio*. The utilization ratio is defined as the ratio between the transmission duration of packets successfully sent and the total available time. The second observed value is the collision ratio, which is the ratio of collisions to the number of packets sent per SI. To understand the impact of the number of nodes on the utilization ratio and collision ratio, we consider a scenario where there is one VBR node, using the Video1_VBR profile, scheduled in HCCA and the number of EDCA nodes, using the EDCA profile of 256 kb/s, is increased from one to sixteen. In these experiments, the EDCA nodes are given the same access category, AC_BE, and hence, the same EDCA parameters. We do this in order to simulate the scenario where the EDCA nodes are using best effort and can be impacted by additional nodes.

Table 4.9: Effect of Load on Utilization and Collision Ratio

Number of EDCA Flows	Utilization Ratio	Collision Ratio
1	0.036	0.000
2	0.072	0.052
4	0.144	0.124
8	0.287	0.193
12	0.388	0.466
16	0.400	0.604

As can be seen in Table 4.9, as the number of nodes increases in EDCA, the utilization increases because there is a greater number of packet transmissions being sent on the wireless channel. Note however that the value of the utilization ratio begins to plateau as nodes are added to the network. This is mainly due to EDCA's contention-based channel access mechanism. Although the channel is being utilized more with the increase of nodes in the network, the probability of collisions increases as well. In Table 4.9, it can be seen that the collision ratio is 0% with one EDCA flow and increases up to 60.4% in the case of sixteen nodes. Although the collision ratio could be used as the only metric to estimate load, varying channel conditions can impact the number of retransmissions experienced. Hence, we use both the utilization ratio and collision ratio in our load estimation. From these results, we can see that the utilization and collision ratios are affected by the load and in our experiments, we use thresholds for the utilization and collision ratio and monitored values as an indicator of the whether EDCA mapping is appropriate.

Performance of EDCA Mapping Adaptation

In order to evaluate the performance of the EDCA mapping adaptation, we look at scenarios where there are four VBR nodes scheduled in HCCA needing additional time, which have the Video1_VBR profile. For these scenarios, we vary the number of EDCA nodes, from four to twelve, and evaluate two EDCA mapping algorithms, one with load estimation and one without. When EDCA mapping occurs, the HCCA nodes are given the AC_VI parameters in EDCA. We include the EDCA mapping without load estimation to show the negative impact on EDCA throughput that can occur if EDCA mapping is always allowed. In addition to delay, we record the percentage decrease in EDCA throughput due to the increased load of the VBR nodes, as well as provide the value of a utility metric, described in Section 4.4.2, which considers both the delay improvement in HCCA and the throughput degradation in EDCA.

We now present a summary of the results in Table 4.10 and 4.11. In the case

without adaptation, the VBR node experiences high unacceptable delay for each of the scenarios. Note that the values of delay are similar because of the fixed allocation policy used by the reference scheduler. Although the VBR nodes experience high delays, the EDCA throughput is above the 256 kb/s requirement when no adaptation is used and the utility metric, which factors in both the average delay of the HCCA flow and the average throughput of the EDCA flows, in this scenario is at 0.53.

Table 4.10: Effect of EDCA Mapping on Delay and Throughput

Number of EDCA Flows	Delay (ms)			% EDCA Throughput Red.	
	Without Adapt	With Adapt Load Est.	With Adapt No Load Est.	With Adapt Load Est.	With Adapt No Load Est.
4	1001	28	28	11%	11%
8	952	45	36	11%	14%
12	964	597	35	35%	91%

Table 4.11: Effect of EDCA Mapping on Utility Value

Number of EDCA Flows	Utility Value		
	Without Adapt	With Adapt Load Est.	With Adapt No Load Est.
4	0.53	1	1
8	0.53	1	1
12	0.53	0.78	0.54

We next look at the impact of our EDCA mapping adaptation using load estimation. As can be seen in the table, the delay for all the scenarios (with 4, 8, 12 flows) have been reduced significantly, with the maximum delay reduction from 1008 ms to 28 ms. Although there is a reduction in the average delay experienced by the VBR flows, a trend can be seen that as the number of EDCA flows increase, the delay with adaptation also increases from 28 ms to 597 ms. This reason for this increase is that our load estimation algorithm limits the amount of EDCA mapping when the network has a high load in EDCA. Hence, the total amount of time allocated to a VBR flow is less in the scenarios where there are more EDCA nodes. Next, we look at the utility value for

the scenarios with adaptation. From the results, we can see that for the scenarios with four and eight nodes, the utility value is at its maximum, indicating that the throughput requirement of EDCA and delay requirement of HCCA are both met. Although there is around a 12% reduction in EDCA throughput due to the additional EDCA node, the average EDCA throughput remains to be higher than its required throughput and so this decrease is not reflected in the utility value. If we look at the case of 12 nodes with our adaptation, the average VBR flow experiences a delay of 597 ms and the 35% decrease in EDCA throughput leaves the utility value at 0.78. Note that although the full utility value is not met, this is still an improvement from the case without adaptation.

Next, we consider the case when EDCA mapping occurs but no load estimation is used. We evaluate the impact of allowing all flows in need of time allocation in HCCA to send in EDCA. In these results, the main objective is to show that the excess load placed on EDCA by the extra VBR flows can lead to significant system degradation. Due to EDCA's contention-based channel access, additional nodes lead to an increased probability of collision, larger waiting times, and lower achieved performance. Table 4.10 provides the results of EDCA mapping without load estimation. From this table, we observe no improvement in the case of four nodes, mainly because the load estimation in this scenario indicates that the load is low enough for the VBR nodes to send in EDCA. In the case of eight nodes, we see that there is a slight improvement in the delay if we allow EDCA mapping whenever needed but it leads to an increase in the EDCA throughput degradation from 11% to 14%. The utility value remains at 1 since both the delay and throughput requirements are met. Finally, in the case of twelve nodes, although there is an improvement in VBR delay, we see a high cost in terms of the percentage reduction of EDCA nodes when no load estimation is used. This cost is reflected in the utility value for this adaptation which is 0.54, indicating that the decrease in throughput outweighs the benefits of adapting the flows. Hence, from these results, it is clear that some load estimation should be used to prevent unfairness to EDCA nodes and system performance degradations.

From the experiments showing results from HCCA allocation and EDCA map-

ping, we have shown that the adaptation framework is able to significantly decrease the delay experienced by VBR flows. Using these two techniques, reallocation of time in HCCA and guided transmission in the EDCA period, we were able to reduce delay noticeably below the tolerated delay and in the case of EDCA mapping, we were able to do this without heavily impacting EDCA throughput. In our next section, we compare our work to a similar dynamic scheduler and highlight the differences between the mechanisms.

4.5.4 Comparison with Flexible HCF

In addition to comparing our work with the reference scheduler, we have also compared the proposed algorithm with Flexible HCF (FHCF), proposed in [98], which adjusts the transmission opportunity based on the dynamic profile. To perform this comparison with FHCF, we implemented the FHCF algorithm in the Opnet framework and considered a number of scenarios. We revisit the scenarios used in Table 4.5 and evaluate the performance of the FHCF algorithm. The experimental results, found in Table 4.12, show that the delay experienced by the VBR flow with FHCF is higher than with our adaptation. We believe this is due to the burstiness of the traffic flow, which can cause the FHCF algorithm to shrink the TXOP prematurely and cause higher delays. Especially in the case of Video1_4, FHCF is unable to reduce the delay below the tolerable threshold.

Table 4.12: Comparison of Variability with FHCF

Profile Name	Data Rate (kb/s)		Delay (ms)		
	Mean	Standard Deviation	Without Adaptation	With Adaptation	With FHCF
Video1_1	329	84	614	19	35
Video1_2	409	153	1854	38	58
Video1_3	421	146	1999	63	121
Video1_4	347	240	1558	121	414

After looking at how one VBR flow fares with FHCF, in the next scenario, we

Table 4.13: Comparison of Proposed Approach with FHCF

Number of VBR Flows	Delay (ms)		
	Without Adaptation	With Adaptation	With FHCF
4	916	12	59
8	814	15	111
16	773	37	491

vary the number of VBR flows between four and sixteen and use similar traffic profiles (Video1_2). We use the same traffic profiles for all flows in order to simulate the case when there is a high correlation between the demands of all the flows. The results are summarized in Table 4.13. In order to compare FHCF and our adaptation, we consider these cases to highlight the differences between the two approaches and potential advantages. In the case of four nodes, the time available in HCCA is sufficient to incorporate the variable requirements in each approach; hence our proposed adaptation and FHCF are expected to have similar performance. However, we observe that our adaptation fares better than FHCF and this is potentially due to burstiness of the traffic flow as observed in the earlier result. In the case of eight nodes, the time available is limited and the AP has to fairly allocate the additional transmission opportunity. In order to achieve fairness, FHCF divides the remaining time evenly between the flows. However, the segmented transmission opportunities may not be sufficient for a flow to overcome traffic variability leading to higher delay as observed in the table. Alternatively, our proposed adaptation attempts to achieve fairness over a number of SIs using a weighted round robin approach. In each SI, the algorithm selects a set of flows to poll and allocates additional time based on their reservation parameters. We believe this approach leads to improved delay performance by our algorithm. Finally, in case of sixteen nodes, there is no time available in the HCCA period, hence, the FHCF flow is unable to use any of its mechanism to deal with the variability in the flows. Our proposed adaptation fares better because of the additional opportunity to transmit in EDCA.

The final aspect we evaluate in this section is the delay effect on other flows scheduled in HCCA period by the FHCF approach. Since FHCF elongates TXOP for

flows, flows already scheduled in HCCA will experience an increased delay due to adaptation. For this evaluation, we observe a case with one VBR flow (Video1_VBR) and multiple CBR flows (Video2_CBR). The performance of the VBR flow improves similar to the previous simulations. However, for the CBR flows scheduled after the VBR node, we observe an increase of 10% in the average delay due to the FHCF adaptation.

4.5.5 Video Quality

With an understanding of the impact of the adaptation, we now evaluate the adaptation on video quality. Video quality is degraded by the player if the video packets are not received in the expected effects of the playout period. Although buffering can be used to smooth out variation in packet delays during playback, large buffering leads to high video startup latency. Additionally, even with nominal buffering, very high packet delay may lead to buffer underflow affecting the video playback. In absence of the desired frame/packet in the buffer, the default error concealment method is to redisplay the previous frame leading to a number of stalls and possible blurring in the video playback.

For the quality evaluation experiments, we use VideoSim [100] to evaluate the effect of our proposed adaptation on end-to-end video quality. Figure 4.14 illustrates the overall framework for our quality evaluation. First, the video stream is encoded at a Variable Bit Rate (VBR) and the traced video is used as an input to the Opnet simulator. The Opnet simulator is then used to evaluate the experienced delay of the video flow over the WLAN topology. The observed Opnet delay is used as an input to VideoSim, which simulates the end-to-end transmission of the video stream, including Internet and buffering delay. Finally, the video is decoded and compared with the original video and the Peak Signal to Noise Ratio (PSNR) is used in this framework as a measure of video quality.

With the above framework, we evaluated an example video (Video3_VBR) and looked at the effect of our proposed adaptation assuming a 5 second buffering at the

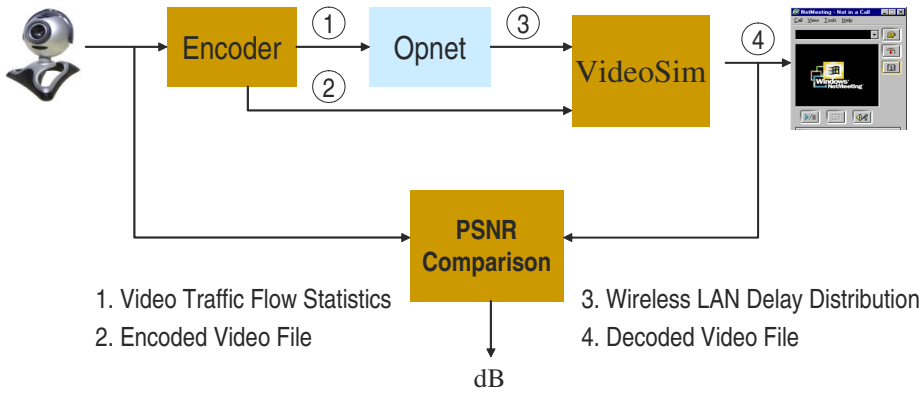
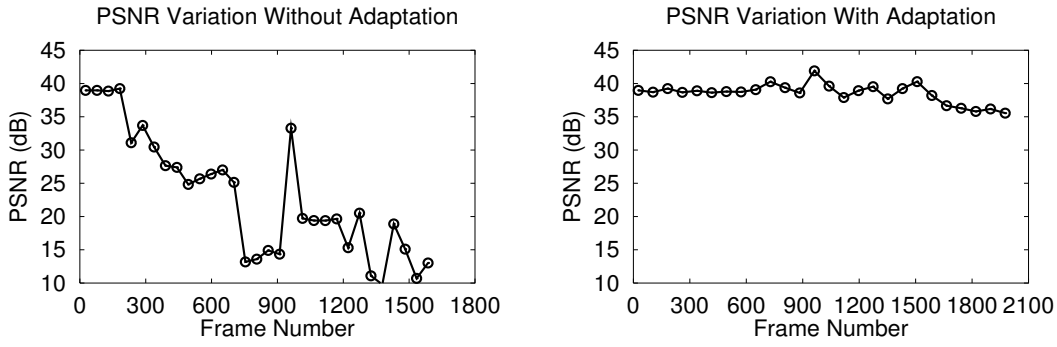


Figure 4.14: Video Quality Measurement Framework



(a) Experienced PSNR Without Adaptation

(b) Experienced PSNR With Adaptation

Figure 4.15: Impact of Adaptation on Quality

player. We observed that the proposed adaptation enables a 16dB improvement (from 22.3db to 38.6db) in video quality. The impact of the adaptation on quality can be seen in Figure 4.15, where we show the PSNR values with and without adaptation for the video frames. Additionally, a snapshot of the video can be seen in Figure 4.16, illustrating potential video quality degradation.

Figure 4.16: Still Shot of Decoded Video Without and With Adaptation

4.5.6 Capacity

The final aspect we investigate is the impact of the adaptation framework on network capacity. As stated in the previous sections, static adaptation of a flow may increase the quality experienced, but will lead to an overall reduction of capacity of the network. Hence, although it is possible to allocate TXOPs large enough to handle the maximum variation of the flow, there is a tradeoff with network capacity.

For these experiments, we evaluate the capacity of a network, assuming similar traffic flows, and calculate the delay experienced by a flow given that time allocation. For these experiments, we consider the Video3_VBR profile used earlier in the quality experiments. Given a minimum tolerable video quality criteria, a maximum tolerable delay of 60 ms, we look at the number of VBR flows that can be supported in the HCCA period. As seen in Table 4.14, the number of nodes that can be supported is higher when the TXOP granted is based on the mean requirements but the total capacity of the network is reduced when the TXOP allocation is based on maximum requirements. However, with the use of the proposed adaptation, the quality experienced by the HCCA flows can not only be improved, but the capacity of the network can also be increased.

Table 4.14: Capacity and Delay Experienced Achieved with Adaptation

Time Allocation	Capacity	Delay (ms)
Mean Allocation	16	275
Extended Time Allocation (2x)	8	25
With Adaptation	24	33

4.6 Related Work

Providing QoS over Wireless LAN has been a prime topic of interest in the recent past. The techniques can be broadly classified into four categories: (1) QoS support for legacy 802.11, (2) 802.11e investigation and adaptation algorithms, (3) Coordination between centralized and distributed channel access, and (4) Cross-Layer adaptation.

4.6.1 QoS Support for Legacy 802.11

Since the legacy 802.11 system only provides best-effort access, initial research work investigated service differentiation by assigning different WLAN channel-access parameters to each of the traffic flows [88, 87]. Additionally, there has been some work concerning individual parameter adaptation to achieve higher throughput [101, 90]. All these attempts, and many others, were targeted towards providing QoS within legacy 802.11 and have led to the design of the new standard IEEE 802.11e.

4.6.2 802.11e Investigation and Adaptation Algorithms

EDCA

Regarding IEEE 802.11e, the majority of published works have been mainly to model, simulate, and analyze the performance of IEEE 802.11e EDCA in comparison to the legacy IEEE 802.11 standard [102, 103, 104, 105, 85]. In addition to these performance evaluations, there has been some work evaluating possible adaptations in IEEE 802.11e. By changing channel access parameters of the contention-based EDCA access

period, work in [89, 92, 106] propose dynamic adaptations to deal with varying network conditions, such as load and channel variance. Additional work has been done in the area of distributed admission control and airtime allocation in [107, 108]. In these works, the main focus is to determine the values of the EDCA parameters, such as CW and TXOP, to meet the goals of the admission control algorithm and node requirements. With these adaptations, support for QoS can be improved in EDCA and as pointed out by [108], EDCA can have advantages over centralized channel access. Some of these advantages include less coordination between the nodes and the AP, flexibility for flows entering and leaving the network, and faster ability to adjust to changing requirements, *i.e.* there is no need to renegotiate allocation when requirements change. However, despite these listed advantages, HCCA can provide for higher efficiency and provide guaranteed time allocation and channel prioritization to real-time flows. Our proposed work can work complementary to these EDCA schemes, using the EDCA adaptation schemes to improve the flows using the distributed access and our proposed algorithm for flows scheduled in HCCA.

HCCA

In terms of related work for the centralized HCCA aspect of 802.11e, there has been limited work mainly focusing on the impact of using a different scheduler than the TGe-reference scheduler scheme, based on Weighted Round Robin (WRR) [95]. For example, [109] showed the improvement of throughput using an Earliest Deadline First (EDF) scheduler. Similarly, [110] proposed the static scheduled contention free burst (S-CFB) scheduling framework that minimizes the number of contention free bursts for energy-efficiency. While the goal of the above work has been to improve scheduling policies in general, in this work, we focus on improving resource allocation to address one prominent behavior of the real-time flows, *i.e.* variable flow requirements.

To support variable flows in 802.11e, there has been some work that addresses this need by statically increasing the TXOP, such as [111] and [112]. In [111], the

TXOP is statically increased in order to meet a minimum packet loss rate and the calculation of this TXOP is used for admission control. In [112], the maximum burst size of a flow is used to statically allocate more time per flow. While these techniques reduce the delay, they negatively impact the system capacity by reducing the total number of flows admitted. Another piece of work addressing variable sources is FHCF [98], where a scheduler is proposed to deal with the limitations of VBR traffic using the reference scheduler. However, as described in Section 4.3.3 and seen in Section 4.5.4, the approach has some limitations. While they propose to change the transmission opportunity duration for a VBR flow, our algorithm attempts to minimize negative effects on other flows by allocating time only after scheduled flows have been serviced and by choosing not to decrease other flows' transmission opportunities.

4.6.3 Coordination Between Centralized and Distributed Channel Access

While there have been a number of techniques to improve aspects of EDCA and HCCA, there has been little literature on the coordination of the two periods. To the best of our knowledge, the interaction of centralized and distributed channel access has only been investigated in [113]. This work focuses on improving the aggregate throughput of access points by dynamically changing the ratio between the legacy IEEE 802.11 channel access periods, DCF and PCF. Although our adaptation does not change the ratio between the centralized and distributed channel access, the mapping allows for increased system utilization and therefore, increases overall system throughput. The proposed adaptation is able to take advantage of the two types of channel access by dynamically changing which stations are in each channel access period based on need.

4.6.4 Cross-Layer Adaptation

The final area of related work is cross-layer adaptation for QoS in WLAN systems. The main contribution of the work presented in [114, 115] is to show the benefit of adapting parameters in multiple layers of the protocol stack to improve video quality. For example, in [114], the authors investigate adaptation of parameters at the application level, specifically of UMCTF video, such as Group of Frames (GOF) size, temporal decomposition levels, and reference frames used in the future and past, in addition to adapting parameters at the MAC layer, such as the frame length and retransmission limit. While these cross-layer schemes have been shown to improve QoS, we believe that the proposed adaptation in this chapter can work complementary to this scheme. In a scenario where all the knobs at the various layers have been set, such as data rate, retransmission limit, and scaling factor, there still remains a problem of allocating sufficient time-to-time varying flows. We believe that our proposed adaptation algorithm can be used to address variability in flows and inaccurate reservations. Combined with our proposed adaptations, we believe that the QoS performance using cross-layer adaptation can be further improved.

From our related work study, we have seen work towards improving QoS through the use of various schemes, including parameter adaptations, admission control, cross-layer adaptations, coordination, *etc*., as described above. While these works all have the same end goal, the proposed method differs and is novel in that it can provide adequate time allocation to scheduled flows, despite the flows' variability. We believe that the proposed technique can complement the approaches of previous work mentioned above and be used to achieve further QoS improvements.

4.7 Conclusions

In this chapter, we identified the limitations of the reference scheduler for 802.11e HCF in supporting real-time multimedia flows. Specifically, the current ref-

erence scheduler is unable to support multimedia flows with variable requirements or flows using dynamic application-level adaptations. Under variable flow requirements, the current reference scheduler leads to high delay, severely impacting the real-time multimedia quality. Additionally, we observed that the HCF framework maps different flows to different modes of medium access with real-time flows being confined to polling-based mechanisms. Although the above separation leads to simplified medium resource management, this can lead to poor channel utilization under heavy HCCA load.

We addressed the above limitations by monitoring the status of each flow for variations in the flow requirements, developing an algorithm to dynamically associate traffic flows appropriately, configuring a polling schedule to allow additional polls to the lagging flows, and selectively mapping real-time multimedia flows to EDCA mode of access under low load in EDCA. We compared our proposed approach with other known techniques using realistic application data profiles collected from video traces with an Opnet-based simulation environment. Our comparative evaluation demonstrates that the adaptation reduces the delay observed in the real-time flows, and fairs better than other known techniques in terms of delay, multimedia quality and capacity of the network.

In developing this adaptation framework, we observed that the variation in the traffic flows under the reference scheduler can significantly increase the delays and even the minimal variability in reservation can lead to high delays. Since the AP allocates time solely based on reservation requests, the accuracy of these requests is important in maintaining the desired QoS. From our experience, we conclude that the reservations should be used as a guideline, whereas additional resources should be allocated dynamically based on monitored requirements of a flow.

The text of this chapter, in part, is based on material that has been published in the ICST/IEEE Intl. Workshop on Broadband Wireless Multimedia in 2004 (N. Ramos, D. Panigrahi, and S. Dey, "Dynamic Adaptation Policies to Improve Quality of Service of Multimedia Applications in WLAN Networks", *Proceedings of ICST/IEEE Intl. Workshop on Broadband Wireless Multimedia*, San Jose, California, October 2004) and

material published in ACM Springer Wireless Networks Journal in 2007 (N. Ramos, D. Panigrahi, and S. Dey, "Dynamic Adaptation Policies to Improve Quality of Service of Real-Time Multimedia Applications in IEEE 802.11e WLAN Networks", *ACM Springer Wireless Networks Journal*, vol. 13, no. 4, pp. 511-535, August 2007). The dissertation author was the primary researcher and author, and the coauthors listed in these publications collaborated on, or supervised the research that forms the basis for this chapter.

Chapter 5

Device and Network Aware Video Scaling for Enhanced Quality and Network Efficiency

5.1 Introduction

In the previous chapter, we focused on improving quality of service through scheduling and resource allocation at the access point. We next target the content source to improve quality and bandwidth efficiency, as well as serve a diverse set of platforms. As described in Chapter 1, there is a diverse set of user devices accessing multimedia over wireless networks. Traditional wireless devices, such as cellular phones and PDAs, are being transformed into multimedia-capable platforms. Additionally, traditional multimedia terminals, such as PCs and TVs, are now being designed to support wireless connectivity. With these developments, the devices requesting multimedia data over wireless networks vastly differ in terms of capabilities, such as screen resolution, supported frame rate, as well as supported bit rate. Content service providers are now faced with the challenging problem of enabling seamless and efficient delivery of multimedia content to viewers using different platforms.

To provide appropriate video content to varying devices and networks, video systems have been reliant on two device-centric schemes. The first is to encode and store various versions of the video at the server *a priori* and send the prepared video stream based on device capabilities. Another method is to use transcoding techniques [62] to decode and re-encode the video before delivery. While both of these schemes can be currently supported, the increasing availability of video content and the greater diversity in device capabilities will make it difficult to continue using these methods in future video systems. The first method of preparing and storing multiple versions suffers from the major drawbacks of needing significant content management and storage space, while the second method of transcoding is costly in terms of the processing power and the number of servers needed to support quick decoding and re-encoding of each video stream.

An alternative to these delivery systems is the recent work in scalable video coding. Over the past few years, significant strides have been made in this area in improving coding efficiency and reducing decoder complexity. These developments have provided the motivation for the Joint Video Team (JVT) to develop and standardize a more powerful scalable video codec, called the H.264/AVC Scalable Video Coding (SVC) Extension [67]. The proposed SVC is a layered video codec that provides bit-stream scalability and allows video to be encoded once at the server in such a way that various scaled streams can be extracted and decoded from the single stream, without the need for re-encoding the video. Building on scaling techniques in previous video standards, SVC makes it possible to extract video scaled in multiple dimensions: spatial, temporal, and quality dimensions. Future video systems using SVC will allow devices to extract a suitable video stream scaled to match their capabilities.

Although SVC provides a notable advantage over previous video standards, the penalty for this scalability comes in terms of the network delivery costs. With SVC, the entire stream, allowing for video extraction at the highest possible quality, will be sent over the network without regard for the capabilities of each receiving device. In the case of wireless networks, this is a significant problem because it causes bandwidth ineffi-

ciency and limits the overall number of devices that can be supported simultaneously. Additionally, there is a cost for each device for processing and receiving the additional scalability layers unnecessarily, in terms of energy, processing power, and memory.

To address the above problem, we describe a Device and Network-Aware Scaling framework, called DeNAS, that will provide efficient delivery of scalable video over wireless networks. Based on device capabilities, network capacity, user channel information, and the desired service objective, DeNAS determines the suitable scaling level for each stream. In this chapter, we consider two different sets of video scaling schemes: user device-centric and network cooperative. In the user device-centric scaling scheme, the DeNAS framework considers only the needs of the individual user device and has the objective of providing the best quality video solely based on a user device's capabilities. With the network cooperative schemes, the objective is to improve overall network efficiency, and therefore improve the aggregate quality of the users, rather than simply meeting individual user requirements. Hence, DeNAS considers the requests of all the devices and uses known information about the network, such as capacity or individual user channel limitations, prior to making a decision on the scalability levels. By performing the appropriate scaling prior to video delivery over the wireless network, DeNAS is able to retain the advantages of SVC of using a single-encoded stream, but is also able to increase bandwidth efficiency, support a greater number of simultaneous users, and improve quality satisfaction.

DeNAS can be used in conjunction with most of the recent SVC-related work, as will be discussed in Section 5.5. While this set of related work looks at supporting full SVC streams, DeNAS differs in that it focuses on scaling based on device information and makes decisions for all devices in the network in the network-cooperative schemes.

Having briefly introduced our work and provided a brief discussion on SVC and related work, the rest of the chapter is organized as follows. We first describe the inefficiencies of sending the full SVC stream and motivate our work in Section 5.2. We next present the proposed framework in Section 5.3 and follow with a description of our proposed scaling algorithms in Section 5.4. In Section 5.5, we describe some of re-

search related to scalable video coding. Experimental results are provided in Section 5.6 demonstrating the effectiveness of DeNAS in improving resource efficiency and quality satisfaction. Finally, in Section 5.7, we conclude and present our chapter conclusions.

5.2 Motivation

In this section, we start by providing motivation for our proposed Device and Network-aware Scaling Framework. As stated earlier, SVC provides many advantages in terms of scalability; however, it can be significantly bandwidth inefficient. As an example, consider a scenario where a video stream is being requested over a wireless network by a set of devices. To keep this example simple, we consider an equal distribution of three possible devices; a laptop, a PDA, and a cellular phone with capabilities that limit viewing quality to a resolution, frame rate (frames per second), and bit rate (kb/s) of CIF 30-512, CIF 15-256, and QCIF 15-128 respectively.

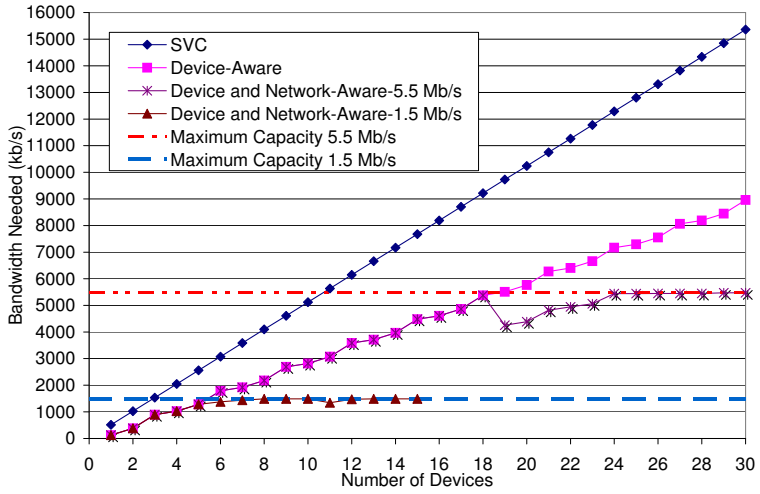


Figure 5.1: Bandwidth Efficiency with SVC and Proposed Adaptations

Comparison of needed bandwidth for SVC, Device-Aware, and Device & Network Capacity-Aware Scaling as the number of user devices increases.

With SVC, the video stream is encoded once at the server, such that the maxi-

mum quality level can be extracted from the stream by the receiving device. In this scenario, the *BUS* video is encoded and sent to each device so that the maximum scalability level (CIF 30-512) can be extracted, irrespective of the device's capabilities. Figure 5.1 shows the bandwidth usage of this method. Note that the bandwidth needed with SVC increases linearly with the number of devices in the network. If a basic admission control scheme is used to limit the number of devices based on the required bandwidth, the network can only support a limited set of devices with SVC. For example, in a network with an application layer throughput limit of 5.5 Mb/s, such as in an 802.11b network, only 10 devices can be supported with SVC. Similarly, in a cellular network with a 1.5 Mb/s application layer throughput limit, only 2 devices can be supported simultaneously in a cellular site.

Next, we look at what can be achieved using the proposed Device and Network-Aware Scaling (DeNAS) framework. In the example scenario, the DeNAS system would receive a request from the devices, determine the suitable video stream, and scale the requested video stream prior to sending over the network. As stated earlier, we consider two types of video scaling schemes: user device-centric and network cooperative. In the first scaling scheme, the DeNAS framework considers only the needs of the individual user device. For example, if the laptop and cellular phone made requests for video streams, the DeNAS framework would consider the scaling of the laptop and phone independently with the user device-centric scaling scheme. On the other hand, in the network cooperative scheme, DeNAS would consider the requests of all the devices in the network prior to making a decision in scaling the video stream.

The bandwidth usage of a user device-centric video scaling algorithm is presented in Figure 5.1 and is labeled in the graph as *device-aware*. Prior to sending the video over the wireless network, this algorithm extracts a scaled video stream matching the maximum capabilities of each device. In this scenario, the laptop would be sent the full CIF 30 video, while the phone would be sent the QCIF 15 video. As seen in the figure, the device-aware scheme achieves significant improvement over SVC, with up to a 42% reduction of bandwidth usage in the case of 30 devices. Additionally, the

device-aware scaling algorithm increases the number of devices that can be supported, from 10 devices with SVC to 18 devices given the application throughput limit of 5.5 Mb/s. In the case of 1.5 Mb/s, the device-aware scheme is able to increase the number of supported devices from 2 to 5 devices.

With the above algorithm, the scaling is performed based solely on device limitations. We next consider a network cooperative scheme where the DeNAS framework uses network information, such as network capacity, to determine the appropriate scaling levels for each device in the network. The device and network capacity-aware scaling algorithm, which is described in Section 5.4, focuses on maximizing quality satisfaction, while improving bandwidth efficiency and supporting a greater number of users. The results with this proposed scaling algorithm can be seen in Figure 5.1, labeled as *device and network-aware*. Bandwidth usage is reduced by 64% in the case of 30 devices compared to SVC. Additionally, the device and network capacity-aware scaling algorithm increases the number of simultaneous video users by 3x, from 10 devices with SVC to 30 given a 5.5 Mb/s application throughput limit. With a 1.5 Mb/s limit, we can increase the number of users from 2 to 15 devices.

As can be seen in this example, DeNAS can be used to increase the number of clients supported in a given network. Additionally, DeNAS is also able to improve the aggregate quality satisfaction across all of the clients using the network, with DeNAS improving quality satisfaction by 3x in comparison to SVC in the scenario of 30 devices as will be shown later in Figure 5.6. We show further results in Section 5.6 on DeNAS' effectiveness in other scenarios, its impact on quality, and also show its ability to handle user channel limitations.

5.3 DeNAS Framework

Having shown the drawbacks of SVC and the potential for efficiently delivering video to a diverse set of devices, we now describe the proposed Device and Network-Aware Scaling framework. In order to perform the scaling prior to the wireless network,

the DeNAS framework must be located such that it can access information about the network, device, and application. The DeNAS framework can be placed in a proxy-type device on the edge of the wireless network or on a high-end Access Point (AP), as shown in Figure 5.2. Clients make requests to the server for a given video and DeNAS intercepts these requests and obtains the scalable video stream from the appropriate server. DeNAS uses the SVC video stream, scalability property information, and network bandwidth limitations to determine the appropriate scaling for each of the devices.

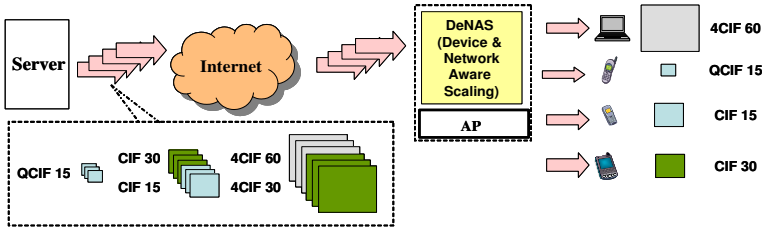


Figure 5.2: DeNAS Framework in a Wireless LAN Network

We now discuss DeNAS' three major components: Network and device monitor, Video characteristics source, and a Video scaling controller and extractor. Figure 5.3 illustrates the components and their interactions with one another.

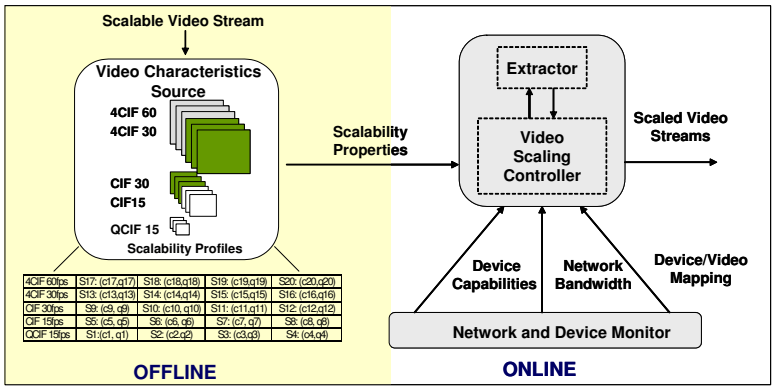


Figure 5.3: DeNAS Components and Interactions

5.3.1 Network and Device Monitor

The first component of the DeNAS framework is the Network and Device Monitor. This component gathers information about the capabilities of the devices, determines which videos are being requested by each device, and obtains the available bandwidth of the network. The network monitor either queries the AP to determine the number of devices or directly obtains current channel condition information from individual devices to estimate the current network bandwidth available. For the device information, the monitor's first task is to determine what type of device is requesting the video stream. The device type can be discovered during the RTP/RTSP handshake. The next step for the monitor is to perform a device lookup in a Wireless Universal Resource File ($WURFL$) [116] database and obtain information about the device capabilities, *e.g.* maximum supported resolution, frame rate, and bit rate.

5.3.2 Video Characteristics Source

The Video Characteristics Source component receives the video stream and provides information about the stream to the DeNAS controller. For each video stream in the network, the component determines a set of enumerated scalability profiles. Each of these profiles represents a scalability level determined by a spatial, temporal, and bit rate combination. For example, the video stream shown in Figure 5.3 has scalability profiles given by a combination of different spatial (4CIF/CIF/QCIF), temporal (15/30/60 fps) and associated bit rate levels. The Video Characteristics Source next determines the attainable quality and associated cost for each profile. The quality information can be calculated offline by the server during the encoding stage and sent with the video stream. Alternatively, a dynamic video quality profiler can be used to determine the attainable quality of the scalable profiles. For the associated cost of the scalability profile, we make use of the profile's bit rate requirements.

5.3.3 Video Scaling Controller and Extractor

The final component of the DeNAS framework is the Video Scaling Controller and Extractor. For a new set of incoming devices, the Video Scaling Controller uses the received and monitored inputs to determine the scaling level for each of the devices. The scaling levels chosen are dependent on the scaling algorithm's desired service objective. For example, with a user device-centric scaling algorithm, the controller bases the scaling solely on the capabilities of individual devices. On the other hand, for a network cooperative algorithm, the controller considers information about the devices in the network, as well as the available network bandwidth and user device channel constraints. We present algorithms to be used in the controller in the following section. Once a scaling level is chosen, DeNAS extracts the appropriate scalable video stream from the original stream. Note that the extractor software can support various scaling input parameters, including spatial level, temporal level, and bit rate desired.

5.4 Proposed Scaling Algorithms

In this section, we present three different scaling algorithms to be used with DeNAS. As illustrated in Section 5.2, the first video scaling algorithm is a user device-centric scheme with the objective of providing the best quality video for a given user solely based on its capabilities and limitations. The latter two algorithms are suitable for network cooperative systems, where the objective is to improve the overall network performance. The first of the network cooperative scaling algorithms works with the network's capacity limitation and determines the appropriate scaling for each user in the network based on capacity, device constraint, and expected video quality impact. This algorithm is suitable when the network operator can allocate the required bandwidth for the stream selected by DeNAS for the entire duration of the video. However, it may be the case that the network is unable to allocate the required bandwidth or is unable to maintain the bandwidth through the entire video stream due to characteristics of the

network, such as the location of the user, channel access scheme, network topology or channel fluctuations. For this case, we describe a second network cooperative video scaling to handle user channel constraints. The proposed algorithm can be utilized to further improve resource usage and quality satisfaction when network scheduling constraints are known. Note that since the network cooperative algorithms need information about the network, these schemes are best deployed by network operators. On the other hand, with the user device-centric scheme, only information about the device is needed and so it can be deployed by both the service provider, as well as the network operators.

We next present the three video scaling algorithms in greater detail. For each, we provide a description of the service objective and details about the algorithm. In our service objective formulations, we consider a wireless network with *N* clients each requesting a unique video stream. For each video stream, the set of scalability profiles is given by S_j:[$S_{j1}, S_{j2}, ...S_{jL}$], where j indicates the video stream index and L represents the maximum number of scalable profiles available for the stream. The chosen profile for all clients in the network is given by $P : [P_i \mid i = 1 : N, P_i \in S_{v(i)}]$, where i is the client index and $v(i)$ is the video for client i. Each scalability profile has a quality, cost, and requirements component. For a given video j and a profile k, these components are given by $S_{jk}.qual$, $S_{jk}.bw$, and $S_{jk}.req$ respectively. For the quality measurement of each profile, $S_{jk}.qual$, there are various subjective and objective metrics that can be used to measure video quality. As described later in Section 5.6, for our experiments, we use a quality metric based on NTIA's VQM because it has been shown to produce quality estimation techniques close to human perception and measures the impact of various quality components. Although we use VQM in our experiments, the algorithms below are agnostic of the quality measure and other metrics can be used. In terms of the profile cost, we focus mainly on the bandwidth requirement of the profile, $S_{jk}.bw$. However, the algorithm can be extended to reflect costs, such as energy or processing costs, if the total energy and processing resources are known. Finally, each profile has a requirement component $S_{jk}.req$ that is in terms of the profile's spatial and temporal properties.

5.4.1 User Device-Centric Video Scaling Schemes

As stated earlier, the objective of a user device-centric scheme is to provide the best quality video for a given user device based solely on its capabilities and limitations. We showed in Section 5.2 that we can improve bandwidth efficiency and quality using a user-device scheme. This set of schemes is ideal for scenarios where the DeNAS framework is working with devices in multiple access networks and network impact is not a concern.

The service objective of our proposed user device-centric algorithm is to maximize individual quality given device capability constraints. In other words, we want to find a set P such that $\forall i \in N\ P_i.qual \equiv maxQual_i$, where $P_i.qual$ is the quality for the selected scalability profile and $maxQual_i$ is maximum attainable quality of the video stream given the device's capabilities. In this chapter, we consider the case when the device capability information reflects the maximum supported capabilities, rather than the run-time capabilities given by the current processing load. We believe that the work can be extended to support run-time processing load by performing periodic updates.

In order to find the appropriate scalability profile, the algorithm performs a search for each user to find the highest quality profile that does not exceed the limitations of the device. The algorithm's complexity is mainly found in sorting the scalability profiles on the quality values and is at most $O(NlogL)$. With this algorithm, the suitable scaling profile is extracted prior to delivery over the wireless network, avoiding using network resources unnecessarily. Since this is a network independent algorithm, it can be used to support devices in different access networks and requires minimal information from the network access point or base station.

5.4.2 Network Cooperative Video Scaling Schemes

We next consider schemes for a network cooperative system, where the video-scaling framework is able to work with the network resource allocation scheme to improve efficiency and meet network-wide service objectives.

Device and Network Capacity-Aware Video Scaling

The first network cooperative video scaling scheme we consider is a device and network capacity-aware scaling algorithm. In contrast to the user device-centric scaling algorithm, this algorithm makes use of the available network capacity information and makes a decision considering all the devices in the network.

For this work, we look to improve the achieved quality and quality satisfaction of each user device in the network. Recall from the earlier discussion in this section, for a user i with a selected video profile P_i, the achieved quality is reported as $P_i.qual$. In addition to the quality, we look at an additional metric that can relate the achieved quality with the maximum quality possible given the user device capabilities, which we term as the user's quality satisfaction. We use quality satisfaction as a metric because this additional metric can relate the achieved quality with the maximum quality possible given the user device capabilities. We define the quality satisfaction, $QS(P_i)$, with the following expression.

$$QS(P_i) = w * P_i.qual/maxQual_i \tag{5.1}$$

where i is the client index, $P_i.qual$ is the quality achieved with the chosen profile, $maxQual_i$ is the maximum achievable quality of user i, and w is the weight factor of the client. In this chapter, we consider equal weights and hence $QS(P_i)$ is defined by the following expression with N representing the total number of users in the network.

$$QS(P_i) = 1/N * P_i.qual/maxQual_i \tag{5.2}$$

Having described the quality satisfaction of a user, we can now define the service objective of our algorithm. The objective is to maximize the overall quality satisfaction given the network capacity and device constraints. The constraints of the objective are to maintain a minimum quality requirement, ensure that the bandwidth allocated does

not exceed the overall network bandwidth, and choose a profile that does not exceed what can be supported by the device. The objective can be expressed by the following equation, where i is the client index, P_i is the selected profile index with $P_i.bw$, $P_i.qual$, and $P_i.req$ representing the bandwidth needed, the quality achievable, and the requirements of the given profile respectively, QS is a function of quality satisfaction as defined above, $bwAvail$ is the total bandwidth available, $minQual_i$ is the minimum required quality guaranteed in the network, $maxCap_i$ is the maximum capabilities of user i, and $aggQS$ is the aggregate quality satisfaction of the network.

$$\begin{array}{lll} \max_P & aggQS = \sum_{i=1}^{N} QS(P_i) & \\ \text{subject to} \quad \forall i \in N & P_i.qual > minQual_i & \\ & P_i.req \leq maxCap_i & \\ & and \quad \sum_{i=1}^{N} P_i.bw < bwAvail & \end{array} \tag{5.3}$$

In order to meet the above objective and constraints, we consider a video scaling algorithm that uses an iterative approach to achieve maximum quality satisfaction. Figure 5.4 presents a flow chart of the proposed scaling algorithm. Briefly, we can summarize the flow diagram in the following steps. The assumed inputs to this algorithm are the video stream, information about scalability properties, network bandwidth availability, and device information, which are gathered by the other components.

- *Step 1:* Using device capability information, determine if the maximum quality obtainable ($maxQual_i$) for each device can be supported. If so, extract the device-appropriate stream but if not, select a set of devices in the network such that the minimum quality requirements ($minQual_i$) of each device can be met.

- *Step 2:* If the network is not overloaded, determine potential change in aggregate quality ($\Delta aggQual$) and aggregate bandwidth cost ($\Delta aggBW$) if all video streams are upgraded to a higher scalability profile. The upgrade can be based on simple bandwidth allocation or on expected quality impact given video scalability properties. Based on device capability information, limit the maximum

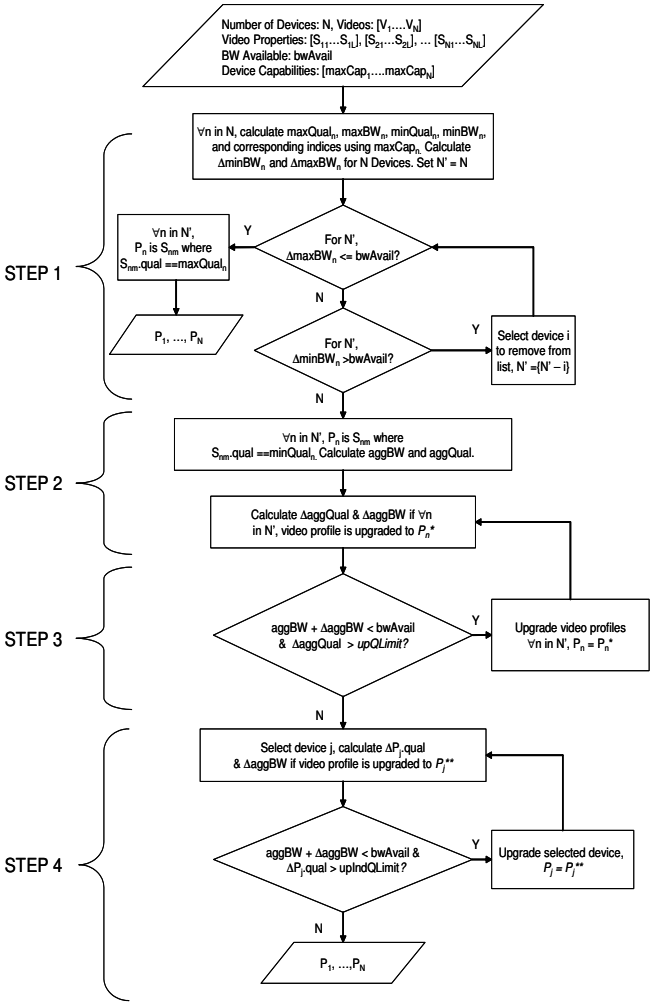


Figure 5.4: Algorithm Flowchart - DeNAS: Network Capacity

profile obtainable with these upgrades.

- *Step 3:* Check for the following criteria: (1) the potential change in aggregate quality is significant and over a set threshold ($upQLimit$) and (2) the aggregate bandwidth cost is within the available network bandwidth. If the above criteria are met, upgrade the profiles for each device and repeat Step 2, considering additional network-wide upgrades.

- *Step 4:* When network-wide upgrades are no longer possible, check to see if it can improve the quality satisfaction of individual devices. Select an individ-

ual device (j) based on weights to upgrade if bandwidth is still available and calculate the potential impact on individual quality ($\Delta P_i.qual$) and cost of the upgrade. If the upgrade is possible given the device capabilities and network constraints, upgrade the scalability level and repeat until no additional upgrades are possible.

By the end of the process, the proposed device and network-aware scaling algorithm is able to select streams that maximize the aggregate quality satisfaction of the network. The algorithm insures a minimum quality for each video stream, performs network-wide upgrades for fairness, maintains bandwidth efficiency by not exceeding device limitations, and stays within the network bandwidth limitations.

Device and Network-Aware Scaling using User Channel Constraints

In the previous network cooperative scheme, the video scaling algorithm determines the appropriate video scaling for each of the devices in the network and assumes that the network access point or base station is able to allocate the necessary bandwidth for each user. However, depending on user location, wireless technology, channel access scheme, physical layer limitations, and network topology, a network access point or base station may be unable to provide the necessary bandwidth to a particular user device. For example, in the case of IEEE 802.11b/a/g, maximum link layer throughput is dependent on the distance from the AP and hence, devices farther away may not be able to achieve the desired throughput. Additionally, since channel access in 802.11 networks is distributed, the achievable bandwidth per device is dependent on the number of devices sharing the channel at any given time. The recent IEEE 802.11e provides some bandwidth guarantees with Hybrid Coordination Function Controlled Channel Access (HCCA) but depending on the available resources, the bandwidth reserved for a user may still be less than the required bandwidth. Additionally, it can be the intent of the network scheduler to limit the achievable data rate of each user to meet additional ser-

vice objectives. For example, the maximum achievable data rate of each user would be limited if the network wanted to ensure equal or proportional bandwidth allocation to the devices to achieve fairness.

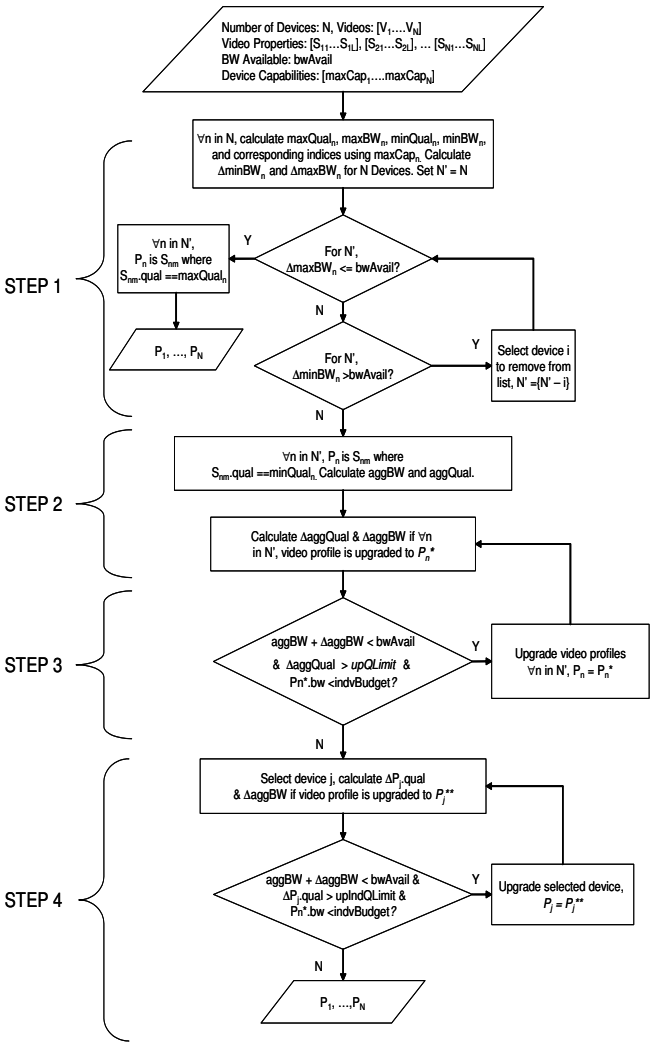


Figure 5.5: Algorithm Flowchart - DeNAS: User Channel Constraints

In order to incorporate these constraints and make better decisions about scalability of each device, we modified the previous video scaling algorithm to include the individual channel limitations. Mainly, we restrict the extent of the scaling upgrade based on the individual user channel limitations and prevent preparing video streams with greater requirements than what can be handled by the user device and network

constraints. In order to handle these user channel limitations, we make changes in the previous algorithm, shown in Figure 5.4, mainly in Steps 3 and 4, where the video scaling levels are upgraded. The changes can be summarized in the flowchart of the algorithm in Figure 5.5. By scaling based on known channel limitations, DeNAS can improve the quality achieved across the network and provide higher quality videos only to the users that are capable of viewing them. We show the impact of the algorithm on quality satisfaction and individual quality in Section 5.6. Note that this algorithm assumes the condition of the channel remains static over the duration of the video stream. In the case where the channel conditions fluctuate, a reevaluation of the situation can be made periodically to adjust to the channel changes.

5.5 Related Work

Before we move to the experimental section, we first provide a brief summary of related research to scalable video coding. To provide appropriate video content to varying devices and networks, video systems were once reliant on storing and serving multiple versions different resolutions at the server or on transcoding methods [62] [117] [118] [63] [119] [64]. Over the past few years, there have been significant research efforts in the area of scalable video coding to improve coding efficiency and decoder complexity. These efforts have made scalability more feasible and recent video standards have included some scalability features, such as quality scalability using fine-grained scalability in MPEG4 [66], as well as spatio-temporal scalability in H.264/AVC [65] [120]. A summary of advancements in scalable video coding can be found in [121].

These developments have renewed interest in scalability and have provided the motivation for the Joint Video Team (JVT), composed of the MPEG and the ITU-T Video Coding Experts Group, to develop and standardize a more powerful scalable video codec, called the H.264/AVC Scalable Video Coding (SVC) Extension [67]. The primary objective of the proposed SVC standard is to provide scalability in multiple dimensions: spatial, temporal, and quality. As described earlier, the proposed SVC is

a layered video codec that provides this scalability in multiple dimensions through bit-stream scalability. In addition to work directly related to the standard, there has been research work on related topics, including techniques for supporting SVC in the lower layers, such as providing transport layer support [122] and prioritization schemes and error protection at the link layer [123], as well as methods of enabling scalability layer switching at the device decoder with little or no change to the encoder [124] [125]. Other related research to SVC includes security streaming and adaptation using SVC discussed in [126] for encryption and authentication.

In addition to research on SVC, there has been some previous research discussing video adaptation with other encoding standards, such as MPEG4 and H.264. For example, there has been research that looks at switching between enhancement layers based on expected quality, such as the research found in [127], [128], and [129]. These techniques focus on trying to minimize the variability in the enhancement layers, the perceptual video quality, short-term rate variability for CBR streams and VBR streams. Out of order buffering as a means of data shaping has also been shown to improve quality, as seen in [100]. Additionally, there have been some high-level discussions of creating an adaptive framework for scalable video, such as the work described in [130] and [131]. In [130], the article discusses the potential of a high-level framework that can combine scalable video representations, network-aware systems, and adaptive services. In [131], the authors discuss the differences between network-centric and end-system centric solutions and some of the advance research that can be used to combine the approach.

The final related research area that we discuss investigates cross-layer strategies for video transmission. There have been a number of techniques proposed in this area, including [132], [133], [134], [135], [136], [114], and [115]. These research techniques evaluate how lower layer techniques can be combined with choosing higher layer encoding parameters to improve end-to-end quality. For example in [135] and [134], error protection schemes for scalable video are investigated at both the higher and lower layers. In [136], the authors look to combine a TCP-friendly video rate adaptation scheme

with unequal forward error protection.

Our DeNAS framework differs from the above mentioned research because it investigates the potential benefits of scaling based on device information, as well as network information. Additionally, in our network-cooperative schemes, DeNAS makes decisions in terms of the entire network and decides the appropriate scalability level for all of the users in the network based on network capacity and user limitations. All the previous research discussed are user device-centric schemes and look to optimize video streams for a single user. DeNAS can be also used in conjunction with a number of the research discussed above. For example, DeNAS can be combined with the research concerning scalable video delivery in the transport and link layers, addressing error concealment and appropriate network parameters, for robustness and improved delivery. Finally, in comparison to SVC, DeNAS proposes the appropriate scaling be made prior to delivery over the wireless networks. Rather then simply putting the burden on the network and devices, DeNAS can be used to avoid inefficient bandwidth use and avoid the delivery and processing cost of including unusable layers unnecessarily.

5.6 Experimental Results

In this section, we present the experimental evaluation of the DeNAS framework and video scaling algorithm. For our experiments, we implemented the proposed video scaling algorithms for the Video Scaling Controller. For the scalable video encoding, decoding, and extraction of streams, we incorporated the JSVM6 Scalable Video Coding Software [67]. Due to the limitation of the JSVM6 encoder, we limit our spatial scalability to CIF and support a temporal scalability of up to 30 frames per second. With these spatial and temporal limitations for the requested video stream, we consider networks, such as WLAN or cellular networks, with capacity limits up to 5.5 Mbps. However, we believe that the experimental results will be applicable to higher levels of scalability and in networks with higher network capacity. We consider three different video sequences: BUS, FOREMAN, and AKIYO, in our experiments.

We report on the average quality experienced by users in the network, as well as the aggregate quality satisfaction introduced in Section 5.4. We use the NTIA's Video Quality Metric (VQM) as the basis of our video quality metric in our experiments. VQM has been shown to produce quality estimation techniques close to human perception and is able to measure the impact of various quality components, including temporal characteristics. To make the quality metric more intuitive, we report the video quality level using the following definition: $QL = 1 - VQM$, where $QL = 1$ signifies the highest quality with no distortion and $QL = 0$ signifies the lowest quality. Hence, each user i with a given profile P_i experiences a quality $P_i.qual$ measured in QL. In scenarios with multiple users, we report the average quality for the users. The value of $avgQ$ is given by the following expression, with i representing the user, $P_i.qual$ being measured using QL, and N representing the total number of users in the network.

$$avgQ = (1/N) * \sum_{i=1}^{N} P_i.qual \tag{5.4}$$

In addition to average quality, we also consider a quality satisfaction metric that considers the achieved quality in relation to the maximum achievable quality for each device. We consider quality satisfaction because it is able to reflect whether the quality level achieved by the user is acceptable given the user device's capabilities. Recall from Equation 5.3, aggregate quality satisfaction is given by $aggQS = \sum_{i=1}^{N} QS(P_i)$ where $QS = (1/N) * (P_i.qual/maxQual_i)$, and $P_i.qual$ and $maxQual_i$ are the achieved and maximum quality measured in QL for user i .

5.6.1 User Device-Centric Video Scaling

We first report on the user-device scaling algorithm discussed in Section 5.4 that selects the correct profile using the maximum capabilities of the device. We first revisit the scenario used in Section 5.2 and consider three set of devices in the network with the viewing capability translating to CIF 30, CIF 15, and QCIF 15. We assume that each device is requesting one *BUS* video stream and a basic admission control is used to limit

the number of devices in the network based on the bit rate needs of the stream. In this scenario, the AP is able to allocate the needed bandwidth to each device and the only limitations considered are the device and network capacity constraint of 5.5 Mb/s.

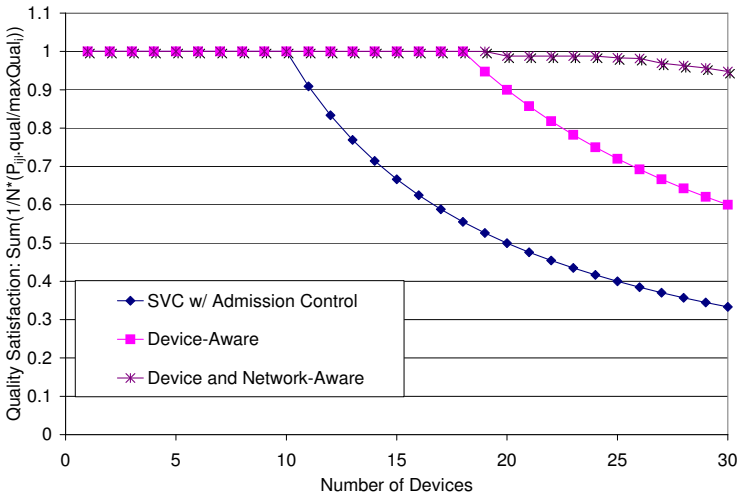


Figure 5.6: Aggregate Quality Satisfaction

Comparison of Aggregate Quality Satisfaction ($aggQS$) for SVC, Device-Aware, and Device & Network Capacity-Aware Scaling as the number of user devices increases.

Figure 5.1 in Section 5.2 showed the benefits of the DeNAS framework in terms of bandwidth efficiency using the device-aware algorithm. In comparison to SVC, the device-aware algorithm reduces bandwidth usage by 42%. We now look at the impact of DeNAS on the aggregate quality satisfaction ratio $aggQS$. Recall that $aggQS$ reflects the average ratio of achieved quality versus the maximum achievable quality of the devices. As seen in Figure 5.6, $aggQS$ is high when the number of devices in the network can be fully supported. However, as the number of users increases beyond the maximum number of devices supported, the $aggQS$ decreases significantly with SVC and the device-aware scheme because some devices are left unserved. The device-aware scheme, however, fares better than SVC with the quality satisfaction at 0.60 with 30 devices versus the 0.33 achieved with SVC.

Next, we look at the average achieved quality, $avgQ$, in the network with the

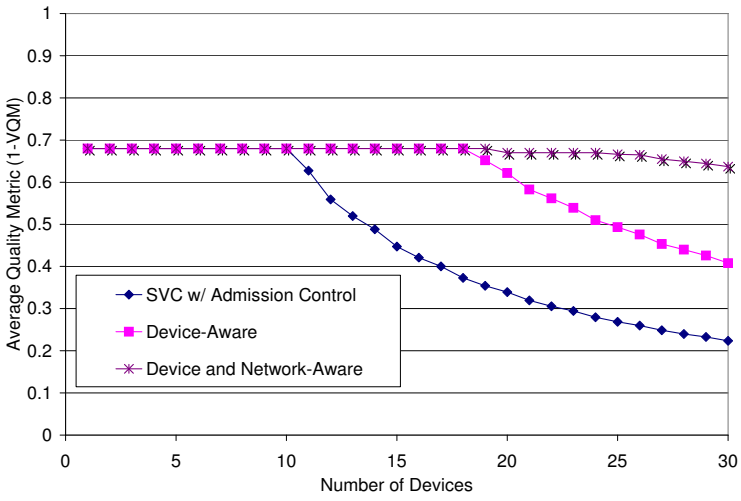


Figure 5.7: Average Experienced Quality

Comparison of Average Experienced Quality ($avgQ$) for SVC, Device-Aware, and Device & Network Capacity-Aware Scaling Algorithms as the number of user devices increases.

various schemes in Figure 5.7. As the number of users increases beyond the maximum number of devices possible with SVC and for the device-aware scheme, the average achieved quality of the network degrades heavily since some devices are left unserved. The device-aware scheme, however, fares better than SVC with the average quality metric at 0.41 with 30 devices versus the 0.22 average quality level experienced with SVC. These results indicate that the user device-centric scheme is able to improve over SVC in terms of bandwidth, quality satisfaction, and experienced quality.

5.6.2 Network Cooperative Video Scaling

In this subsection, we present results using the network cooperative video scaling schemes. We first begin with the scaling using network capacity information and then present results when the algorithm incorporates user channel information.

Device and Network Capacity-Aware Video Scaling

We begin by presenting the results of DeNAS using network capacity information. Recall from Section 5.4, the algorithm's objective is to maximize aggregate quality satisfaction based on the constraints of the device, as well as the available network bandwidth.

We look at the scenario described in the previous section, which has an equal distribution of user devices requesting the BUS video. Figure 5.1 in Section 5.2 showed that in a scenario with an equal distribution of devices, the device and the network capacity-aware algorithm, reduces bandwidth usage by up to 64%. In terms of aggregate quality satisfaction, $aggQS$, we can maintain an aggregate quality satisfaction of 0.95 with 30 devices and outperforms SVC. By using network capacity information as well as device information, we see that more devices are able to access the network and achieve a satisfiable quality. Similarly if we look at the $avgQ$ of the device and network capacity-aware algorithm, we see that the proposed algorithm achieves the highest average quality obtainable in this scenario, achieving a 0.65 average quality level at 30 users as seen in Figure 5.7.

After looking at a scenario with an equal distribution of devices, we next evaluate the proposed device and network-aware scaling algorithm over multiple test cases. For these experiments, we look over a set of test cases, changing the scenario characteristics by changing the number of devices, varying the device capabilities by choosing different capabilities for each user for each test case, and looking at three different video sequences: *AKIYO*, *BUS*, and *FOREMAN*.

Table 5.1 summarizes the results of these experiments. Each row represents the average of five test cases with a given video stream and number of devices combination. This table reports the number of devices supported in the network, the average quality score, and the aggregate quality satisfaction over the five runs. In the experiments with SVC, we assume a favorable admission control policy that serves the devices with the highest possible achievable quality.

Table 5.1: Experiments with DeNAS using Network Capacity Information

Comparison of SVC and Device & Network Capacity-Aware Scaling in terms of Number of Devices Supported, Average Experienced Quality, and Aggregate Quality Satisfaction with different videos, number of devices, and distributions of device type.

Video Name and Number of Devices	Number of Devices Served		Average Experienced Quality		Aggregate Quality Satisfaction	
	SVC	DeNAS	SVC	DeNAS	SVC	DeNAS
Bus-10	10	10	0.670	0.670	1.00	1.00
Bus-20	10	20	0.383	0.619	0.500	0.920
Bus-30	10	30	0.263	0.617	0.333	0.927
Foreman-20	20	20	0.638	0.638	1.00	1.00
Foreman-30	21	30	0.462	0.645	0.700	1.00
Foreman-40	21	40	0.353	0.619	0.525	0.965
Akiyo-20	20	20	0.882	0.882	1.00	1.00
Akiyo-30	21	30	0.625	0.854	0.700	0.964
Akiyo-40	21	40	0.477	0.834	0.525	0.944

We first start by looking at the number of devices that can be supported in the network given the different scaling methods. The table shows that our proposed scaling algorithm is able to support a greater number of devices in the network across all the test cases. In the particular combination of the BUS video stream with 30 users, we observe the maximum improvement of 3x with only 10 users supported with SVC in compared to 30 users with DeNAS using network capacity. Next, we look at the average quality, $avgQ$, achieved in the different scenarios. While the $avgQ$ is similar in the cases where the number of devices is small, DeNAS achieves a significantly higher average quality improves over SVC in the cases where there are a greater number of user devices. Additionally, we observe a similar trend with the aggregate quality satisfaction in the different scenarios because more user devices are able to access the network and are satisfied with DeNAS. While the quality satisfaction and achieved quality for SVC quickly deteriorates with more devices, DeNAS scales well with increasing number of devices, with minimum degradation in quality satisfaction and average quality experienced. DeNAS is able to achieve a quality satisfaction score in the range of 0.92-1,

while the quality satisfaction using SVC is as low as 0.33 with the *BUS* stream. In all cases, DeNAS is able to achieve high quality satisfaction, while maintaining the minimum quality requirement.

Device and Network-Aware Scaling Using User Channel Constraints

We now focus on evaluating the performance of DeNAS in the case where there are channel limitations for individual users. In the previous scenarios, the scaling decision is made assuming that the access point is able to provision the necessary bandwidth for each user. However, as discussed in Section 5.4, this may not always be possible due to various reasons, including channel conditions resulting from distance and topology, as well as the wireless network's channel access scheme.

Table 5.2: Experiments with DeNAS using User Channel Limitations: Aggregate
Comparison of Aggregate Quality Satisfaction using SVC, DeNAS using Network Capacity, and DeNAS using User Channel Constraints.

Type & Number of Devices	Aggregate Quality Satisfaction		
	SVC	DeNAS using Network Capacity Information	DeNAS using User Channel Information
ALL-15	0.750	0.997	0.998
ALL-20	0.500	0.983	0.997
ALL-25	0.400	0.982	0.996
ALL-30	0.333	0.991	0.997
LAP-15	0.750	0.963	1.00
LAP-20	0.500	0.952	1.00
LAP-25	0.400	0.976	1.00
LAP-30	0.333	0.985	1.00

The proposed scaling algorithm can improve the chosen scaling levels for each user by ensuring the chosen scalability can be supported by the device's capabilities, as well as their channel limitation. To show the impact of this algorithm, we evaluated DeNAS using channel information and observed the improvement in network quality

satisfaction. For these experiments, we consider topologies consisting of 15, 20, 25, and 30 devices, where each of the devices is requesting the BUS video and the maximum capacity is 5.5 Mb/s. For each network topology, we performed 20 test runs to vary the maximum achievable bandwidth of each user in a given network. Table 5.6.2 summarizes the results of these experiments, where each row represents results with a topology for a fixed number of users. The table provides the achieved quality satisfaction with SVC, with DeNAS using network capacity information, and finally, with DeNAS using user channel information. Additionally, the maximum improvement of an individual user achieved by with DeNAS using channel information is given in terms of the discussed quality metric, QL, and the corresponding percentage improvement. Recall that the quality satisfaction score is the ratio between the quality of the received video over the maximum achievable quality. We adjust the maximum achievable quality to reflect the user channel impairments.

Table 5.3: Experiments with DeNAS using User Channel Limitations: Individual *Comparison of Maximum User Improvement in Experienced Quality of DeNAS using Channel Information versus DeNAS using Network Capacity.*

Type & Number of Devices	Maximum User Improvement			
	QL (VQM-based)		QL (PSNR-based)	
	Quality Difference	Percentage	Quality Difference (dB)	Percentage
ALL-15	0.07	10.14%	0	0%
ALL-20	0.16	24.24%	2.86	10.52%
ALL-25	0.16	25.40%	7.21	32.27%
ALL-30	0.13	20.63%	6.03	27.08%
LAP-15	0.13	18.84%	2.86	10.52%
LAP-20	0.17	26.15%	4.91	22.05%
LAP-25	0.16	25.40%	7.28	32.69%
LAP-30	0.13	20.63%	6.16	27.28%

In the first set of experiments, we assume an even distribution of all three devices (laptop, PDA, and cellular phone) in the network. These experiments are labeled as *ALL* in the results. We start by looking at the quality satisfaction scores with SVC, DeNAS

using only network capacity information, and DeNAS using channel information. In all of the scenarios, the DeNAS algorithm based on network capacity information continues to outperform SVC even with the user channel limitations. We next look at what improvements can be achieved using channel information. As seen from the quality satisfaction scores, the majority of users are able to reach their maximum achievable quality given their channel and device limitations with DeNAS using user channel information. While the quality satisfaction score shows improvement over the entire network, we next look at individual users and discuss the maximum improvement achieved with a single user in these scenarios. As can be seen in the table, the potential impact for a single user can be significant. For example, in the case of 25 devices, the maximum user improvement is 0.16, when a laptop originally obtaining a VQM-based quality of 0.63 is able to improve its quality up to a 0.79. This is approximately a 25% improvement in the laptop's experienced quality. We also calculated the quality impact using PSNR as the quality metric. The columns of Table 5.3 show the maximum individual user improvement measured in PSNR for the discussed scenarios. The maximum individual user improvement with DeNAS using channel information is 7.21 dB in the case of 25 users.

In our next set of experiments, we look at scenarios where the bandwidth demand is higher and consider the case where all the devices are laptops. We consider similar scenarios to the above experiments with 15, 20, 25, and 30 devices and perform 20 test runs to vary the maximum achievable data rate of each device. These results are also provided in Table 5.6.2 and are labeled as *LAP* in the table. As can be seen in the quality satisfaction scores, the DeNAS adaptation using network capacity information outperforms SVC. Since the devices have higher bandwidth needs, the quality satisfaction scores are slightly lower than the previous results. We next look at DeNAS' performance using user channel information. The results indicate that DeNAS is able to improve the quality satisfaction score to almost 1, meaning that the majority of devices can reach their maximum achievable quality. Finally, in terms of the maximum individual quality improvement with DeNAS using channel information, we observe in

Table 5.3 results similar to the previous scenarios with a VQM-based quality metric improvement of 0.17, approximately 25%, and a PSNR maximum improvement of 7.28 dB.

Having shown the summarized data for these scenarios, we next show the quality experienced by the devices measured using the VQM-based quality metric QL in a chosen scenario in order to better understand how the quality improvement is distributed over the network. Figure 5.8 shows the LAP-20 scenario, where DeNAS is able to achieve the maximum improvement in quality satisfaction using additional channel information. The X axis provides the user number for the 20 devices and their maximum achievable data rate. The Y axis shows the VQM-based quality metric experienced. For each device, we compare the experienced quality with DeNAS using network capacity versus scaling using channel information. As can be seen in the figure, many of the individual devices, such as N1, N2, and N4, are able to experience significantly higher quality video when DeNAS can use additional channel information to determine video scaling. When the quality of each profile is measured in PSNR, DeNAS is able to achieve the maximum improvement in quality satisfaction in the LAP-25 scenario. Figure 5.9 shows the quality impact in this scenario of using channel information with DeNAS. Many of the individual user devices, such as N1, N2, and N4 are able to achieve a significant improvement of up to 5 dB when user channel information is considered.

5.6.3 Evaluating DeNAS in a WLAN Network

In the previous experiments, we evaluated the proposed video scaling algorithms with abstracted information of the network capacity and channel conditions. In addition to these experiments, we wanted to evaluate DeNAS in a simulated WLAN environment. For this work, we used the Opnet tool to simulate an IEEE 802.11b environment. Rather than choose a default packet distribution to represent the video stream, we incorporated actual packet traces of the SVC encoded video stream and the scaled video streams created by DeNAS in order to have an accurate application load in our simulations. The

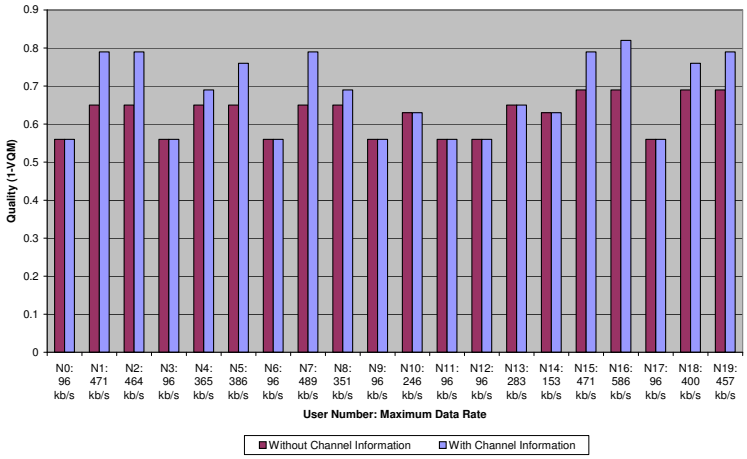


Figure 5.8: Experienced Quality Improvement (VQM-based) with DeNAS Using Channel Information

Results showing distribution of improvement in Experienced Quality based on VQM metric with DeNAS using Network Capacity versus DeNAS using User Channel Information in a 20 device scenario.

packet traces of the scalable video streams were incorporated into the Opnet Application and Profile configuration. In our experiments, we considered the default simulation parameter settings for an 11 Mb/s IEEE 802.11b network and assumed that no admission control scheme is used for the network. We varied the number of users from 10 to 20 with an equal distribution of Laptops and PDAs and used a 20 second looped sequence of the BUS video for the video sequence. To understand the quality of the video received at the user device, after each simulation, we used the set of successfully received packets over the simulated network and reconstructed the video that would be viewed given the device's capabilities.

The results from this set of simulations show that the use of SVC can lead to serious congestion problems in the network. As the number of devices increases, the observable delay experienced by each device increases significantly with SVC and the device-aware scheme, in comparison to the DeNAS algorithm. The delay observed is dependent on the setting of the number of nodes in the network, the AP buffer size,

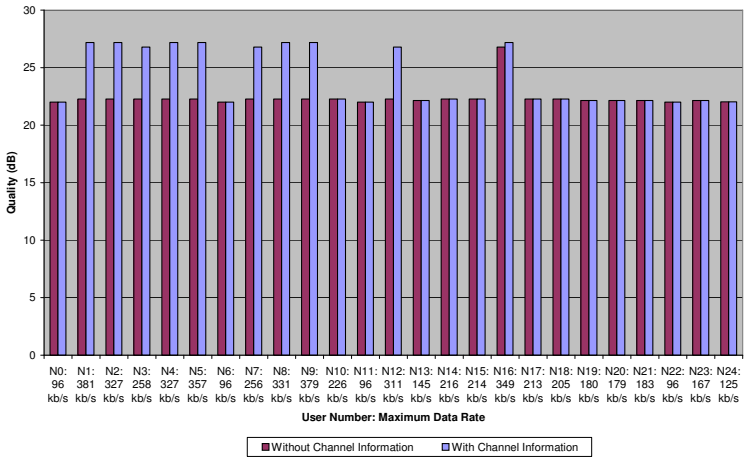


Figure 5.9: Experienced Quality Improvement (PSNR) with DeNAS Using Channel Information

Results showing distribution of improvement in Experienced Quality in PSNR with DeNAS using Network Capacity versus DeNAS using User Channel Information in a 25 device scenario.

and the amount of background traffic. As the number of devices in the network increased, the AP experiences large delays, collisions and retransmissions because of the contention-based medium access control used in 802.11. For example, in the case of 20 devices, devices are only able to receive one viewing of the video with SVC but are able to observe more than two full viewings of the looped video with DeNAS. Hence, the delay with SVC is 2x greater than DeNAS due to the amount of traffic the AP has to handle. Additionally, in the test cases where the SVC traffic overloaded the 2 MB AP buffer, several SVC streams suffer from significant packet loss, making the video stream undecodable. In contrast, DeNAS is able to support 20 devices comfortably without any packet loss and significantly alleviate the load of the network.

5.7 Conclusion

In this chapter, we presented the DeNAS framework. By finding the appropriate scaling level for each video stream prior to its delivery over the wireless network, DeNAS is able to provide a high quality stream to each device, while increasing network efficiency and satisfying a greater number of clients than SVC. We demonstrated through our experimental results the effectiveness of DeNAS in achieving high quality satisfaction, relieving network load and increasing network capacity.

The text of this chapter, in part, is based on material that has been published in the IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications in 2007 (N. Ramos and S. Dey, "A Device and Network-Aware Scaling Framework for Efficient Delivery of Scalable Video over Wireless Networks", *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Athens, Greece, September 2007) and material submitted to the IEEE Transactions on Broadcasting (N. Ramos and S. Dey, "User Device-Centric and Network-Cooperative Video Scaling Algorithms for Efficient High Quality Video Delivery", *IEEE Transactions on Broadcasting*). The dissertation author was the primary researcher and author, and the coauthors listed in these publications collaborated on, or supervised the research that forms the basis for this chapter.

Chapter 6

Future Work

As described in this thesis, there are a number of challenges in supporting multimedia over wireless networks. These challenges include energy-efficiency, network efficiency, quality of service, and serving diverse platforms. This thesis addressed these challenges directly by enabling network adaptation schemes at the user device, access point, and content source. In particular, it introduced two user device schemes; a dynamic network parameter adaptation algorithm to adapt link layer parameters to reduce energy consumption and a channel-dependent packet level tuning scheme to adjust link layer parameters to achieve higher quality and throughput. The thesis also proposed an access point scheduling algorithm to provide a means of allocating resources to users depending on their application needs. By recognizing the variation in application requirements and dynamically scheduling additional resources when needed, we were able to improve several quality metrics and overall network capacity. Finally, we looked to adapting the content source and developed a framework to scale multimedia video over the network for a diverse set of devices. We showed the potential of improving quality and network efficiency by considering properties of the network and the device in determining the appropriate video scaling.

In this final chapter, we discuss future research directions and opportunities that can be pursued given the work presented in this thesis. We group the possible future

work into two categories. The first set of extensions look at broadening the scope of each of the individual network adaptations. The second set extend the entire scope of the thesis and look into possible applications of the developed work.

6.1 Extensions of Individual Network Adaptations

The individual network adaptations proposed in this thesis were shown to improve quality and efficiency of wireless multimedia transmissions. In this section, we discuss the scope of possible extensions for each of the individual network adaptations. We briefly describe future work that can be pursued and discuss the potential improvements that can be made with each of the adaptations.

6.1.1 Energy Parameter Adaptation

For this thesis, we focused at energy adaptations only at the user and looked primarily at the data link layer. One possible extension is to look for other parameters in the data link layer and investigate the appropriate tuning for these parameters for minimizing energy. Another possible extension is to look at other networks and determine the feasibility of using the developed work in the context of these networks. This can include other wireless technologies with different physical layers or can also include IEEE 802.11 ad-hoc networks. Note that the developed work has been investigated in the context of IEEE 802.11 infrastructure network but the potential impact of energy improvement can also be evaluated in an IEEE 802.11 ad-hoc network.

6.1.2 Quality of Service Parameter Adaptation

In terms of parameter adaptation for improved quality, we can further improve quality by determining the appropriate settings for the IEEE 802.11-specific parameters and determining the mapping from DiffServ to these parameters. While we looked at specific problems like priority inversion and starvation, further work can be done to

maintain different service levels. By monitoring the channel conditions and whether users are achieving their expected throughput and delay requirements, dynamic adaptation of channel access parameters can ensure that the appropriate resources are provisioned for each user. Additionally, coordination with a centralized controller or with other users in the network for parameter settings can be used as another means of improving service differentiation using dynamic parameter adaptation.

6.1.3 Quality of Service Scheduling and Resource Allocation

In the service differentiation work using reservations, there are a number of directions that can be pursued. First, there is a possibility of incorporating varying channel conditions in determining the appropriate scheduling. The next is to focus on resource allocation to meet variable application requirements and investigate the impact of this work in terms of energy consumption. A combined approach of parameter adaptation and scheduling can also be developed to coordinate the access point work and the device adaptations to ensure that the adaptations do not negatively impact each other. Finally, a possible direction is to look at admission control and determining which users are the best to serve given their capabilities, requests, and channel conditions.

6.1.4 Content Scaling for Diverse User Devices

The final network adaptation we discuss is the developed video scaling work. A future direction would be to investigate how video scaling can take into account different priority level and service differentiation. Additionally, we can also look into the impact of scaling on energy and choose the scaling based on available energy and current processing capabilities. Finally, we can consider a joint scheme where we combine video scaling and link layer scheduling to improve delivery of multimedia video. Although the approach developed in this thesis can greatly enhance the capacity of the video delivery system, individual streams delivered may still face network and quality problems. In order to be able to allocate network resources judiciously to each appropri-

ately scaled SVC stream, a future research direction would be to develop a joint video scaling-network scheduling technique, where both the video scaling choices and network allocation choices are performed simultaneously. For example, in the context of an IEEE 802.11e-based network, resource allocation and network parameters would be decided to match the scaling chosen for the individual stream by DeNAS. The proposed approach would enable a high capacity video delivery system, capable of streaming high quality video to a large number of video terminals, under any given network condition.

6.2 Application of Developed Network Adaptations

Having described possible improvements to the individual network adaptations, we next describe potential extensions and applications of the entire body of research developed in this thesis. One possible future direction is to investigate the work in the context of large-scale wireless networks. The developed adaptation schemes have focused and evaluated on a single network and wireless technology. In the near future, we envision scenarios where it is necessary to support the delivery of multimedia applications with high requirements to a large number of users using multiple wireless technologies, and different methods of connectivity. We believe the developed work can be enhanced such that the adaptation techniques can be used in the context of these networks. The second possible future direction is in creating an adaptation manager for these schemes. While we were able to show the potential of these individual network adaptations, we believe that further improvement can be made by looking at joint and coordinated network adaptation. We describe these two possible extensions in the following subsections.

6.2.1 Large Scaled Wireless Networks

The first potential extension of the work presented in this thesis is to explore the developed adaptations in the context of large scaled wireless networks. Here, we

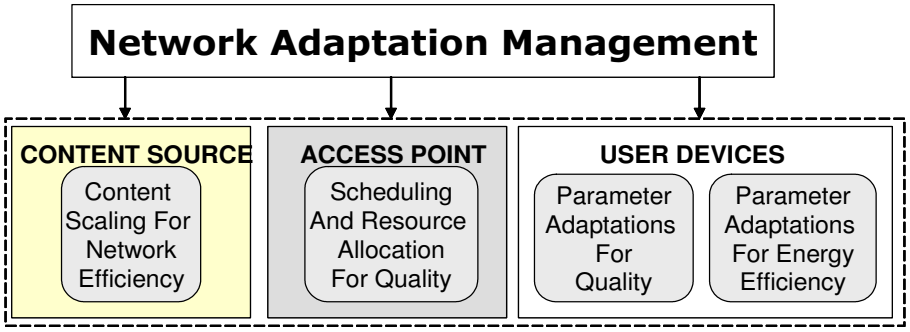


Figure 6.1: Future Work

consider network adaptation in systems with multiple wireless connectivity points and wireless technology supporting a large number of users requesting a rich diverse set of applications. In these systems, network adaptation will be necessary to support user requirements since quality and resource efficiency will be essential in these networks.

We believe that this work can be extended to work in these large scaled networks and that there is potential of the network adaptation working at additional network elements. This extension will include investigating how to work over multiple access points, wireless technologies, and even different channel access schemes. Some of the key challenges that have to be investigated include load balancing between access points, handoffs, and ensuring connectivity. These challenges will have to be investigated in order to understand the potential of the proposed work in these type of networks. Additionally, a possible extension is to create a demonstration of these adaptations and evaluate their effectiveness outside of the simulation framework.

6.2.2 Coordinated Adaptation Management

Another future direction would be to evaluate coordinated adaptation management. This thesis proposed different network adaptation focusing on different metrics including energy efficiency, network capacity, and quality of service. A natural step would be to coordinate these network adaptations based on current conditions and on the network service objective. The adaptation management would determine which adap-

tations should be active and also decide individual parameters for each of the schemes. As illustrated in Figure 6.1, the adaptation management would receive information from the content source, access points, and user devices. This could be done passively by actively monitoring the channel and overhearing packets exchanges or may be performed more directly by performing a periodic query to the network elements to determine the current state. Also, this management system could maintain current information about the users, which include device type, capabilities, and subscribed service levels, that could be used by the adaptation algorithms.

References

[1] "TIA telecommunications market review and forecast," WWW page, May 2007. [Online]. Available: http://www.tiaonline.com/business/research/mfr/index.cfm

[2] "US connected home forecast 2010," WWW page, Jan. 2006. [Online]. Available: http://www.strategyanalytics.net/default.aspx?mod=PressReleaseViewer&a0%=2776

[3] "Power consumption of energy efficiency comparisons of WLAN products," WWW page, 2003. [Online]. Available: http://www.atheros.com/pt/whitepapers/atheros_power_whitepaper.pdf

[4] J. Min and H. Samueli, "Analysis and design of a frequency-hopped spread-spectrum transceiver for wireless personal communications," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 5, pp. 1719–1731, Sept. 2000.

[5] L. Larson, "Radio frequency integrated circuit technology for low-power wireless communications," *IEEE Wireless Communications*, vol. 5, no. 3, pp. 11–19, June 1998.

[6] I. O'Donnell and R. Brodersen, "An ultra-wideband transceiver architecture for low power, low rate, wireless systems," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 5, pp. 1623–1631, Sept. 2005.

[7] F. Ellinger and W. Bachtold, "Compact and low power consuming frequency/phase multiplier MMICs for wireless LAN at S-band and C-band," *IEE Proceedings- Circuits Designs and Systems*, vol. 150, no. 3, pp. 199–204, June 2003.

[8] D. Panigrahi, C. Chiasserini, S. Dey, A. R. Rao, and K.Lahiri, "Battery life estimation for mobile embedded systems," in *Proceedings of International Conference on VLSI*, Bangalore, India, Jan. 2001, pp. 55–63.

[9] R. Rao, S. Vrudhula, and D. Rakhmatov, "Battery modeling for energy aware system design," *IEEE Computer*, vol. 36, no. 12, pp. 77–87, Dec. 2003.

[10] B. Prabhu, A. Chockalingam, and V. Sharma, "Performance analysis of battery power management schemes in wireless mobile devices," in *Proceedings of IEEE Wireless Communications and Networking Conference*, Mar. 2002, pp. 825–831.

[11] C. Chiasserini and R. Rao, "Improving energy saving in wireless systems by using dynamic power management," *IEEE Transactions on Wireless Communications*, vol. 2, no. 5, pp. 1090–1100, Sept. 2003.

[12] M. Liebsch and X. Perez-Costa, "Utilization of the IEEE 802.11 power save mode with IP paging," in *Proceedings of IEEE ICC*, May 2005, pp. 1383–1389.

[13] M. Sarkar and R. L. Cruz, "An adaptive sleep algorithm for efficient power management in WLAN," in *Proceedings of IEEE VTC*, May 2005.

[14] X. Perez-Costa and D. Camps-Mur, "APSM: bounding the downlink delay for 802.11 power save mode," in *Proceedings of IEEE ICC*, May 2005, pp. 3616–3622.

[15] D. Qiao and K. Shin, "Smart power-saving mode for IEEE 802.11 wireless LANs," in *Proceedings of IEEE INFOCOM*, Mar. 2005, pp. 1573–1583.

[16] Y. Wei, S. Bhandarkar, and S. Chandra, "A client-side statistical prediction scheme for energy aware multimedia data streaming," *IEEE Transactions on Multimedia*, vol. 8, 2006.

[17] V. Erramilli, I. Malta, and A. Bestavros, "On the interaction between data aggregation and topology control in wireless sensor networks," in *Proceedings of Sensor and Ad Hoc Communications and Networks*, Oct. 2004, pp. 557–565.

[18] Y. Chen, A. Liestman, and J. Liu, "A hierarchical energy-efficient framework for data aggregation in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 3, pp. 789–796, May 2006.

[19] S. J. Baek, G. de Vecancia, and X. Su, "Minimizing energy consumption in large-scale sensor networks through distributed data compression and hierarchical aggregation," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1130–1140, Aug. 2004.

[20] Y. Yu, B. Krishnamachari, and V. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks," in *Proceedings of IEEE INFOCOM*, Mar. 2004, pp. 7–11.

[21] F. Lin, H.-H. Yen, S.-P. Lin, and Y.-F. Wen, "MAC aware energy-efficient data-centric routing in wireless sensor networks," in *Proceedings of IEEE ICC*, June 2006, pp. 3491–3496.

[22] M. Youssef, M. Younis, and K. Arisha, "A constrained shortest-path energy-aware routing algorithm for wireless sensor networks," in *Proceedings of IEEE WCNC*, Mar. 2002, pp. 794–799.

[23] S. Gandham, M. Dawande, and P. Prakash, "An integral flow-based energy-efficient routing algorithm for wireless sensor networks," in *Proceedings of IEEE WCNC*, Mar. 2004, pp. 2341–2346.

[24] M. Pursley, H. Russell, and J. Wysocarski, "Energy-efficient routing in multimedia ad hoc networks," in *Proceedings of IEEE WCNC*, Mar. 2004, pp. 1311–1316.

[25] Z. Zhao, X. Zhang, P. Sun, and P. Liu, "A transmission power control MAC protocol for wireless sensor networks," in *Proceedings of IEEE International Conference on Networking*, Apr. 2007, pp. 5–7.

[26] K. Lenung and L.-C. Wang, "Integrated link adaptation and power control to improve error and throughput performance in broadband wireless packet networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 619–629, Oct. 2002.

[27] B. Radunovic and J. L. Boudec, "Optimal power control, scheduling, and routing in UWB networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp. 1252–1270, Sept. 2004.

[28] D. Li, Y. Sun, and Z. Feng, "Joint power allocation and rate control for real-time video transmission over wireless systems," in *Proceedings of IEEE Globecom*, Dec. 2005, p. 5.

[29] X. Guo, S. Roy, and W. Conner, "Spatial reuse in wireless ad-hoc networks," in *Proceedings of IEEE VTC*, Oct. 2003, pp. 1437–1442.

[30] F. Ye and B. Sikdar, "Distance-aware virtual carrier sensing for improved spatial reuse in wireless networks," in *Proceedings of IEEE Globecom*, Dec. 2004, pp. 3793–3797.

[31] J.-S. Park, A. Nandan, M. Gerla, and H. Lee, "Space-mac: enabling spatial reuse using MIMO channel-aware MAC," in *Proceedings of IEEE ICC*, May 2005, pp. 3642–3646.

[32] J. Zhu, X. Guo, L. Yang, and W. Conner, "Leveraging spatial reuse in 802.11 mesh networks with enhanced physical carrier sensing," in *Proceedings of IEEE ICC*, June 2004, pp. 4004–4011.

[33] T. Krauss, T. Thomas, and F. Vook, "Direction of arrival and capacity characteristics of an experimental broadband mobile MIMO-OFDM system," in *Proceedings of IEEE VTC*, Apr. 2003, pp. 29–33.

[34] H. Bolcskei, D. Gesbert, and A. Paulraj, "On the capacity of OFDM-based spatial multiplexing systems," *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 225–234, Feb. 2002.

[35] M. Torabi, S. Aissa, and M. Soleymani, "[mimo-ofdm," in *Proceedings of IEEE ICC*, 2006.

[36] D. Qiao and S. Choi, "Goodput enhancement of IEEE 802.11a wireless lan via link adaptation," in *Proceedings of IEEE ICC*, Helsinki, Finland, June 2001, pp. 1995–2000.

[37] D. Qiao and K. G. Shin, "Achieving efficient channel utilization and weighted fairness for data communications in IEEE 802.11 WLAN under the DCF," in *Proceedings of IEEE International Workshop on Quality of Service*, Miami Beach, FL, May 2002, pp. 227–36.

[38] O. Chaker and J. Conan, "Multicast capacity of wireless ad hoc networks with hierarchical routing," in *Proceedings of IEEE VTC*, Apr. 2007, pp. 95–99.

[39] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 89 – 103, Jan. 2005.

[40] T. Liu and W. Liao, "Capacity-aware routing in multi-channel multi-rate wireless mesh networks," in *Proceedings of IEEE ICC*, June 2006, pp. 1971 – 1976.

[41] P. Wang, H. Jiang, and W. Zhuang, "Capacity improvement and analysis for voice/data traffic over WLANs," *IEEE Transactions on Wireless Communications*, vol. 6, pp. 1530 – 1541, Apr. 2007.

[42] L. Cai, X. Shen, J. Mark, L. Cai, and Y. Xiao, "Voice capacity analysis of WLAN with unbalanced traffic," *IEEE Transactions on Vehicular Technology*, vol. 55, pp. 752 – 761, 2006.

[43] Z. Li, Y. Cui, H. Wang, and W. Wu, "Call admission control scheme in multi-service DWCS," in *Proceedings of IEEE ICC*, May 2005, pp. 3349 – 3353.

[44] "Video traces for network performance evaluation," http://trace.eas.asu.edu.

[45] Y. Sun and J. Daigle, "A source model of video traffic based on full-length VBR MPEG4 video traces," in *Proceedings of IEEE Globecom*, 2005, p. 5.

[46] J.-A. Z. B. L. I.Ahmad, "Traffic modeling for layered video," in *Proceedings of IEEE ICME*, July 2003, pp. 497–500.

[47] C.-N. Chuah and R. Katz, "Characterizing packet audio streams from internet multimedia applications," in *Proceedings of IEEE ICC*, 2002, pp. 1199–1203), month = may.

[48] "IETF DiffServ working group," WWW page, Feb. 2003.

[49] IEEE-802.11WG, "Draft supplement to standard for telecommunications and information exchange between systems - LAN/MAN specific requirements - part 11: MAC enhancements for quality of service (QoS)," IEEE 802.11e Standard Draft/D8.0, Feb. 2004.

[50] V. Pandey, D. Ghosal, and B. Mukherjee, "Exploiting user profiles to support differentiated services in next-generation wireless networks," *IEEE Network*, vol. 18, pp. 40–48, 2004.

[51] J.-Y. Chang and H.-L. Chen, "A resource reservation scheme with dynamic grouping for multimedia wireless networks," in *Proceedings of IEEE WCNC*, Mar. 2003, pp. 1337–1340.

[52] "IETF integrated services in the internet architecture: An overview," WWW page, June 1994.

[53] N. Nasser and H. Hassanein, "Bandwidth reservation policy for multimedia wireless cellular networks and its analysis," in *Proceedings of IEEE ICC*, June 2004, pp. 3030–3034.

[54] H. B. Kim, "An adaptive bandwidth reservation scheme for multimedia mobile cellular networks," in *Proceedings of IEEE ICC*, May 2005, pp. 3088–3094.

[55] Y. Xiao and H. Li, "Voice and video transmissions with global data parameter control for the IEEE 802.11e enhance distributed channel access," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, pp. 1041–1053, 2004.

[56] G. Paschos, I. Politis, and S. Kotsopoulos, "A quality of service negotiation-based admission control scheme for WCDMA mobile wireless multiclass services," *IEEE Transactions on Vehicular Technology*, vol. 54, pp. 1875–1886, 2005.

[57] S. Choi and K. Shin, "Adaptive bandwidth reservation and admission control in QoS-sensitive cellular networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, pp. 882 – 897, 2002.

[58] M. Hollick, T. Krop, J. Schmitt, H. Huth, and R. Steinmetz, "Comparative analysis of quality of service routing in wireless metropolitan area networks," in *Proceedings of IEEE LCN*, Oct. 2003, pp. 470–479.

[59] A. Bashandy, E. Chong, and A. Ghafoor, "Generalized quality-of-service routing with resource allocation," *IEEE Journal on Selected Areas in Communications*, vol. 23, 2005.

[60] G. Raju, G. Hernandez, and Q. Zou, "Quality of service routing in ad hoc networks," in *Proceedings of IEEE WCNC*, 2000.

[61] J.-R. Ohm, *Multimedia Communication Technology: Representation, Transmission and Identification of Multimedia Signals*. Springer Multimedia, 2004, ch. 15.

[62] A. Vetro, C. Christopoulos, and H. Sun, "An overview of video transcoding architectures and techniques," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18–29, 2003.

[63] S. Chandra, C. Ellis, and A. Vahdat, "Application-level differentiated multimedia web services using quality aware transcoding," *IEEE Journal on Selected Areas in Communication*, vol. 18, pp. 2544–2565, 2000.

[64] Y. Liang and Y.-P. Tan, "A new content-based hybrid video transcoding method," in *Proceedings of International Conference on Image Processing*, Oct. 2001, pp. 429 – 432.

[65] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, 2003.

[66] "MPEG4 moving picture experts group," WWW page. [Online]. Available: http://www.chiariglione.org/mpeg/

[67] J. Reichel, H. Schwarz, and M. W. (eds), "Scalable video coding - Joint draft 6, JVT-S201," Geneva, Switzerland, Apr. 2006.

[68] R. Krashinsky and H. Balakrishnan, "Minimizing energy for wireless web access with bounded slowdown," in *Proceedings of ACM MobiCom*, Atlanta, GA, September 2002.

[69] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *ACM Wireless Networks*, vol. 8, no. 5, September 2002.

[70] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed priority scheduling and medium access in ad hoc networks," *Wireless Networks*, vol. 8, no. 5, pp. 455–466, 2002.

[71] B. P. Crow, I. Widjaja, J. G. Kim, and P. Sakai, "Investigation of the IEEE 802.11 medium access control (MAC) sublayer functions," in *Proceedings of IEEE INFOCOM*, Kobe, Japan, Apr. 1997, pp. 126–33.

[72] P. Lettieri and M. B. Srivastava, "Adaptive frame length control for improving wireless link throughput, range, and energy efficiency," in *Proceedings of IEEE INFOCOM*, San Francisco, CA, Apr. 1998, pp. 564–571.

[73] S. Agarwal, R. H. Katz, S. V. Krishnamurthy, and S. K. Dao, "Distributed power control in ad-hoc wireless networks," in *Proceedings of IEEE PIMRC*, San Diego, Sept. 2001, pp. 59–66.

[74] J. Ebert and A. Wolisz, "Combined tuning of RF power and medium access control for WLANs," *ACM Mobile Networks and Applications*, vol. 6, no. 5, pp. 417–26, 2001.

[75] N. Poojary, S. V. Krishnamurthy, and D. Son, "Medium access control in a network of ad hoc mobile nodes with heterogeneous power capabilities," in *Proceedings of IEEE ICC*, Helsinki, Finland, June 2001, pp. 872–7.

[76] E. S. Jung and N. Vaidya, "A power control mac protocol for ad hoc networks," in *Proceedings of ACM MobiCom*, Atlanta, Georgia, Sept. 2002.

[77] M. Zorzi and R. Rao, "ARQ error control for delay-constrained communications on short-range burst-error channels," in *Proceedings of IEEE VTC*, Pheonix, AZ, May 1997, pp. 1528–1532.

[78] J. P. Ebert, B. Burns, and A. Wolisz, "A trace-based approach for determining the energy consumption of a WLAN network interface," in *Proceedings of European Wireless*, Florence, Italy, Feb. 2002, pp. 230–236.

[79] J. Ebert, S. Aier, G. Kofahl, A. Becker, B. Burns, and A. Wolisz, "Measurement and simulation of the energy consumption of a WLAN interface," Telecommunication Networks Group, Technische Universitt Berlin, Tech. Rep., 2002.

[80] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proceedings of IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 1548–57.

[81] J. Ebert and A. Willig, "A Gilbert-Elliot bit error model and the efficient use in packet level simulation," Telecommunication Networks Group, Technische Universitt Berlin, Tech. Rep., Mar. 1999.

[82] M. K. A. Willig and A. Wolisz, "Measurements and stochastic modeling of a wireless link in an industrial environment," Telecommunication Networks Group, Technische Universitt Berlin, Tech. Rep., Mar. 2001.

[83] IEEE, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Standard, June 1999.

[84] "Opnet simulation framework," http://www.opnet.com.

[85] D. Gu and J. Zhang, "QoS enhancement in IEEE 802.11 wireless local area networks," in *Proceeding of World Wireless Congress (3G Wireless)*, 2003.

[86] R. Jain, G. Babic, B. Nagendra, and C. Lam, "Fairness, call establishment latency and other performance metrics," ATM Forum/96-1173, Tech. Rep., Aug. 1996.

[87] M. Barry, A. T. Campbell, and A. Veres, "Distributed control algorithms for service differentiation in wireless packet networks," in *Proceedings of IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 582–590.

[88] I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11," in *Proceedings of IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 209–18.

[89] L. Romdhani, Q. Ni, and T. Turletti, "Adaptive EDCF: Enhanced service differentiation for IEEE 802.11 wireless ad hoc networks," in *Proceedings of IEEE WCNC*, Mar. 2003.

[90] A. Banchs and X. Perez, "Providing throughput guarantees in IEEE 802.11 wireless LAN," in *Proceedings of IEEE WCNC*, Orlando, FL, Mar. 2002, pp. 130–8.

[91] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless LAN," in *Proceedings of ACM MobiCom*, Boston, MA, Aug. 2000, pp. 167–78.

[92] B. S. Kim, Y. Fang, and T. F. Wong, "Dynamic fragmentation scheme for rate-adaptive wireless LANs," in *Proceedings of IEEE PIMRC*, Beijing, China, Sept. 2003.

[93] J. Tourrilhes, "Fragment adaptive reduction: Coping with various interferers in radio unlicensed bands," in *Proceedings of IEEE ICC*, Helsinki, Finland, June 2001.

[94] Y. Kwon, Y. Fang, and H. Latchman, "A novel medium access control protocol with fast collision resolution for wireless LANs," in *Proceedings of IEEE INFOCOM*, San Francisco, CA, Apr. 2003.

[95] J. Prado, "Mandatory TSPEC parameters and reference design of a simple scheduler," IEEE 802.11-02/705ar0, Nov. 2002.

[96] T. V. Lakshman, A. Ortega, and A. R. Reibman, "VBR video: Tradeoffs and potentials," in *Proceedings of IEEE*, May 1998, vol. 86, no. 5, pp. 952–973.

[97] M.-T. Sun and A. Reibman, *Compressed Video over Networks*. Marcel Dekker, 2001, ch. 9.

[98] P. Ansel, Q. Ni, and T. Turletti, "An efficient scheduling scheme for IEEE 802.11e," in *Proc. of WiOpt (Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks)*, Mar. 2004.

[99] J.-A. Zhao, B. Li, C.-W. Kok, and I. Ahmad, "MPEG-4 video transmission over wireless networks: A link level performance study," *Wireless Networks*, vol. 10, pp. 133–146, 2004.

[100] C. Taylor and S. Dey, "ORBit: An adaptive data shaping technique for robust wireless video clip communication," in *Asilomar Conference on Signals Systems and Computers*, Pacific Grove, Nov. 2003.

[101] D. Qiao and K. G. Shin, "UMAV: A simple enhancement to the IEEE 802.11 DCF," in *Hawaii International Conference on System Sciences*, Hawaii, Hawaii, january 2003.

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

[103] S. Choi, J. del Prado, S. Shankar, and S. Mangold, "IEEE 802.11e contention-based channel access (EDCF) performance evaluation," in *Proceedings of IEEE ICC*, Anchorage, Alaska, May 2003.

[104] A. Lindgren, A. Almquist, and O. Schelén, "Quality of Service schemes for IEEE 802.11 wireless LANs - an evaluation," *In Special Issue of the Journal on Special Topics in Mobile Networking and Applications (MONET) on Performance Evaluation of Qos Architectures in Mobile Networks*, vol. 8, no. 3, pp. 223–235, June 2003.

[105] A. Grilo and M. Nunes, "Peformance evaluation of IEEE 802.11e," in *Proceedings of IEEE PIMRC*, Lisboa, Portugal, Sept. 2002.

[106] N. Ramos, D. Panigrahi, and S. Dey, "ChaPLeT: Channel-dependent packet level tuning for service differentiation in IEEE 802.11e," in *Proceeding of Intl. Symposium on Wireless Personal Multimedia Communications*, Yokusuka, Japan, Sept. 2003, pp. 86–90.

[107] A. Banchs, X. Perez-Costa, and D. Qiao, "Providing throughput guarantees in IEEE 802.11e Wireless LANs," in *Proceedings of International Teletraffic Congress*, Berlin, Germany, Sept. 2003.

[108] C.-T. Chou, S. Shankar, and K. G. Shin, "Achieving per-stream QoS with distributed airtime allocation and admission control in IEEE 802.11e Wireless LANs," in *Proceedings of IEEE INFOCOM*, Miami, Florida, Mar. 2005, pp. 1584–1585.

[109] A. Grilo, M. Macedo, and M. Nunes, "A scheduling algorithm for QoS support in IEEE 802.11e networks," *IEEE Wireless Communications*, vol. 10, no. 3, pp. 36–43, 2003.

[110] L. Zhang, Y. Ge, and J. Hou, "Energy-efficient real-time scheduling in IEEE Wireless LANs," in *IEEE International Conference on Distributed Computing Systems*, Providence, Rhode Island, May 2003, pp. 658–673.

[111] W. F. Fan, D. Gao, D. Tsang, and B. Bensaou, "Admission control for variable bit rate traffic in IEEE 802.11e WLANs," in *International Symposium on Multi-Dimensional Mobile Communications*, Aug. 2004, pp. 272–277.

[112] L. Lim, R. Malik, P. Tan, C. Apichaichalermwongse, K. Ando, and Y. Harada, "A QoS scheduler for IEEE 802.11e WLANs," in *Proceedings of IEEE Consumer Communications and Networking Conference*, Las Vegas, Jan. 2004, pp. 199–204.

[113] X. J. Dong, M. Ergen, P. Varaiya, and A. Puri, "Improving the aggregate throughput of access points in IEEE 802.11 wireless LANs," in *IEEE Workshop on Wireless Local Networks*, Bonn, Germany, Oct. 2003.

[114] S. Shankar, Z. Hu, and M. van der Schaar, "Cross-layer optimized transmission of wavelet video over IEEE 802.11a/e WLANs," in *Proceedings of International Packet Video Workshop*, Irvine, CA, Dec. 2004.

[115] M. van der Schaar and S. Shankar, "Cross-layer wireless multimedia transmission: Challenges, principles, and new paradigms," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 50–58, Aug. 2005.

[116] "Wireless universal resource file," http://www.wurfl.sourceforge.net.

[117] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," *In Proceedings of the IEEE*, vol. 93, no. 1, pp. 84–97, Jan. 2005.

[118] N. Kuwahara, T. Hata, T. Nozawa, and A. Vetro, "Network transcoder with seamless switching function," MERL, Tech. Rep., Dec. 2004.

[119] Y. Hong, K. Lee, J. Kim, and W.-K. Cho, "High speed architecture for MPEG-2/H.264 video transcoding," in *International Symposium on Communications and Information Technologies*, Oct. 2006, pp. 674–678.

[120] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, no. 7, pp. 645–656, 2003.

[121] J.-R. Ohm, "Advances in scalable video coding," in *In Proceedings of the IEEE*, vol. 93, no. 1, Jan. 2005, pp. 42–55.

[122] Y.-K. Wang, S. Wenger, and M. M. Hannuksela, "System and transport interface of H.264/AVC scalable extension," in *Proceedings of IEEE ICIP*, 2006, pp. 165–168.

[123] M. van der Scharr, Y. Andreopoulos, and H. Zhiping, "Optimized scalable video streaming over IEEE 802.11a/e HCCA wireless networks under delay constrants," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 755–768, 2006.

[124] G. Liebl, T. Schierl, T. Wiegand, and T. Sctockhammer, "Advanced wireless multiuser video streaming using the scalable video coding extensions of H.264/MPEG4-AVC," in *Proceedings of IEEE ICME*, 2006, pp. 625–628.

[125] X. Sun, W. Feng, L. Shipeng, W. Gao, and Y.-Q. Zhang, "Seamless switching of scalable video bitstreams for efficient streaming," *IEEE Transactions on Multimedia*, vol. 6, pp. 291–303, 2004.

[126] J. Apostolopoulos, "Architectural principles for secure streaming and secure adaptation in the developing scalable video coding standard," in *Proceedings of IEEE ICIP*, 2006, pp. 729–732.

[127] T. Kim and M. Ammar, "Optimal quality adaptation for scalable encoded video," *IEEE Journal on Selected Areas in Communication*, vol. 15, no. 1, pp. 344–356, 2005.

[128] P. de Cuertos and K. Ross, "Adaptive rate control for streaming stored fine grained and scalable video," in *Proceedings of NOSSDAV*, May 2002.

[129] R. Rejaie, D. Estrin, and M. Handley, "Quality adaptation for congestion controlled video playback over the internet," in *Proceedings of SIGCOMM*, Aug. 199, pp. 189–200.

[130] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Scalable video coding and transport over broadband wireless networks," *Proceedings of the IEEE*, vol. 89, no. 1, pp. 6–20, 2001.

[131] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "End-to-end QoS for video delivery over wireless internet," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 123–134, 2005.

[132] D. Niyato and E. Hossain, "A novel analytical framework for integrated cross-layer study of call-level and packet-level QoS in wireless mobile mutlimedia networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 3, pp. 322–334, Mar. 2007.

[133] A. Ksentini and M. Naimi, "Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture," *IEEE Communications Magazine*, pp. 107–114, Jan. 2006.

[134] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive cross-layer protection strategies for robust scalable video transmissions over 802.11 WLANs," *IEEE Transactions on Selected Areas in Communications*, vol. 21, no. 19, pp. 1732–1763, 2003.

[135] A. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 278–290, Apr. 2004.

[136] T. Ahmed, A. Mehaoua, and R. Boutaba, "Adaptive packet video streaming over IP networks: A cross-layer approach," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 385–401, Feb. 2005.