國立成功大學
資訊工程學系

博士論文

支援服務品質保證與多點傳輸之無線區域網路
Quality of Service Supports in Multihop Wireless Local Area Networks

研究生：陶明宏
指導教授：鄭憲宗 博士

中華民國九十六年六月

博碩士論文授權書

(國科會科學技術資料中心版本 93.2.6)

本授權書所授權之論文為本人在 國立成功 大學(學院) 資訊工程 系所 ______ 組 九十五 學年度第 二 學期取得 博 士學位之論文。

論文名稱： 支援服務品質保證與多點傳輸之無線區域網路

☑同意 □不同意

本人具有著作財產權之論文全文資料，授予行政院國家科學委員會科學技術資料中心（或其改制後之機構）、國家圖書館及本人畢業學校圖書館，得不限地域、時間與次數以微縮、光碟或數位化等各種方式重製後散布發行或上載網路。

本論文為本人向經濟部智慧財產局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：__________，註明文號者請將全文資料延後半年再公開。

☑同意 □不同意

本人具有著作財產權之論文全文資料，授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，不限地域與時間，惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鉤選，本人同意視同授權。

指導教授姓名：鄭憲宗

研究生簽名：陶明宏　　　　學號：P78911202

(親筆正楷)　　　　(務必填寫)

日期:民國 96 年 7 月 2 日

1. 本授權書 (得自http://sticnet.stic.gov.tw/sticweb/html/theses/authorize.html 下載或至 http://www.stic.gov.tw首頁右下方下載) 請以黑筆撰寫並影印裝訂於書名頁之次頁。
2. 授權第一項者，請確認學校是否代收，若無者，請自行寄論文一本至台北市(106-36)和平東路二段 106 號 1702 室 國科會科學技術資料中心 黃善平小姐。(本授權書諮詢電話: 02-27377606 傳真：02-27377689)

國立成功大學

資訊工程學系

博士論文

支援服務品質保證與多點傳輸之無線區域網路

研究生：陶明宏

本論文業經審查及口試合格特此證明

論文考試委員

指導教授：

系(所)主管：

中華民國九十六年六月二十二日

Quality of Service Supports in Multihop Wireless Local Area Networks

By

Ming-Hung Tao

A thesis submitted to the Graduate Division in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY in

Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan, R.O.C.

June 22, 2007

Approved by:

Advisor:

Chairman:

支援服務品質保證與多點傳輸之無線區域網路

陶明宏
國立成功大學資訊工程研究所

中文摘要

近年來，因為無線網路市場對於即時通訊服務(real-time services)的需求日益增加，網路服務品質保證(Quality of Service, QoS)變成了一個相當重要的議題，特別是在網路通訊協定的設計上面。然而，由於無線頻道容易遭遇受到干擾和一些不可預測的狀況，要設計一個好的無線媒介存取控制(Medium Access Control, MAC)協定，是件相當不容易的事。

在本論文中，我們分析了現存的 MAC 協定在處理即時通訊服務方面的缺陷，並且提出了一個完整的解決架構。為了提供無線區域網路(Wireless Local Area Networks, WLANs) QoS 的支援，我們在架構中提出了一個基於排程演算法的中央管理式 MAC 協定；為了使得我們所提出的中央管理式 MAC 協定具備多點傳輸的能力，我們設計了一套多點傳輸(multihop)的機制。另外，由於在多路徑的傳輸過程中容易遺失原本被保證的 QoS 需求，我們發展了一個具備延遲感應(delay-sensitive)能力的公平排隊模組來補強此一缺陷。

我們所提出的中央式 MAC 協定是由一個時槽排程(slot-scheduling)演算法，以及一個增強 QoS 控管的允入控制(admission control)演算法所組成。時槽排程演算法的設計，是為了能有效率的利用網路的頻寬，並且公平的替各種服務作傳輸排程。允入控制演算法的設計，是為了能夠正確的管理頻寬資源以及保障每個服務的 QoS 需求。我們所提出的多點傳輸機制，可以套用在任何的中央式 MAC 協定和任何的網路拓樸上；此機制除了解除中央式 MAC 協定的封閉傳輸限制外，並能改善隱藏終端點(hidden terminal)問題。此外，為了確保所有的服務經過多點轉傳時也能兼顧到 QoS 的需求，我們在易變的 WLAN 上提出了一個具備延遲感應能力的公平排隊模組。這個模組可以延緩即時通訊服務的儲列(queue)成長速度，維持傳輸的公平性，並且保證每個網路節點的總吞吐量(throughput)會在一定程度以上。

我們也對於在論文中所提出的協定、機制、模組，提供了實驗和理論分析的模型，藉由這些模組，可以更容易的驗證這些技術的正確性與可行性。總言之，本論文的成果將有助於解決無線區域網路上 QoS 和多點傳輸方面的難題。

關鍵字：即時通訊服務，服務品質保證，多點傳輸，公平排隊，隱藏終端點。

Quality of Service Supports in Multihop Wireless Local Area Networks

Ming-Hung Tao
Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan.

Abstract

Since the demand for the wireless transmission of real-time services has been growing significantly in recent years, Quality of Service (QoS) becomes a very important issue in the design of network protocols. However, it is not trivial to design a good Medium Access Control (MAC) protocol which provides real-time services large amounts of bandwidth and also supports multihop transmission, since wireless channel suffers much interference and unpredictable problems.

In this dissertation, we analyze the performance of the existing wireless MAC protocols, and propose a framework which provides QoS supports and multihop transmission ability for Wireless Local Area Networks (WLANs). This framework consists of a centralized scheduling-based MAC protocol, a multihop mechanism, and a delay-sensitive fair queueing model.

The scheduling-based MAC protocol contains a slot-scheduling algorithm and a QoS enhanced admission control algorithm. The slot-scheduling algorithm is designed to efficiently utilize the network bandwidth and fairly schedule the transmission for various services while the QoS-enhanced admission control algorithm is designed to manage resources and guarantee the QoS requirements of services. The multihop mechanism that eliminates the restriction on single-hop transmission for centralized MAC protocols and alleviates the hidden terminal problem can be adapted to various protocols and network topologies. Moreover, to ensure that the services in relay stations meet their QoS requirements, the delay-sensitive fair queueing model is proposed for highly variant multihop WLANs. This queueing model slows down the growth of queue length for real-time services, still maintains the property of fairness, and guarantees the throughputs of stations.

This dissertation also provides experimental and analytical models for the proposed protocols, mechanisms and algorithms. With these models, the correctness and feasibility

of our framework can be verified. Based on the achievements in this dissertation, the QoS issue in the multihop WLAN can be easily addressed.

Keywords: real-time services, quality of service, multihop transmission, fair queueing, hidden terminal problem.

誌謝

本論文之完成，首先要感謝我的指導教授 鄭憲宗老師。老師在我的大學、碩士、博士的求學生涯中，一直不吝於給我關心與指導。從基本的資訊學門訓練到論文研究的啟發，老師總是在我最需要的時候給予指引。一直以來，老師以嚴謹的態度與科學的方法指導我作研究，令我受益良多。也感謝老師提供了非常好的研究環境與資源，以及付出相當多的時間與精神修訂我的論文，讓我能夠順利取得博士學位，在此致上最真誠的謝意。

感謝系上許多關心我的老師，因為你們的鼓勵，讓我更加執著於學問的追求。感謝實驗室的學弟妹（伯符、玉池、妍帆、思韻、仕安、哲佑)，你們帶來了實驗室前所未有的歡樂氣氛，陪我度過許多快樂時光，也謝謝你們在我論文口試所提供的幫助。感謝實驗室畢業的學長們，你們的寶貴經驗讓我能更快地適應實驗室的生活。感謝系辦公室的佳蓉，不厭其煩地給予我很多協助。也感謝口試委員們對於本論文提出許多寶貴的建議，使得本論文能更加完善，在此一併答謝。

感謝我的父母，因為你們的支持和鼓勵，讓我能無後顧之憂地攻讀博士，更篤定地完成我的學業。感謝我的女友，這幾年來一直陪著我度過許多的波折，在我不順遂的時候不斷地給我鼓勵，傾聽我的心聲。因為有父母和女友的陪伴，我才能度過這艱辛的求學過程。也感謝所有關心我的親友，你們的關懷給予我不斷前進的動力，讓我受用不盡。

謹將此論文獻給我的家人以及所有關心我的人

請與我一起分享這份榮耀與喜悅

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

WLAN is the fastest growing segment of the communications market. A WLAN is a wireless communication system that allows computers and workstations to communicate data with each other in a local area using radio waves as the transmission medium. WLANs can provide almost all the functionality and high data-transmission rates offered by wired LANs, but without the physical constraints of the wire. Low installation costs, high availability, and mobile data connectivity are the significant advantages of WLANs.

IEEE 802.11 is the first WLAN standard and so far the only one that has secured a market. This standard defines the medium access control and physical layers for a LAN with wireless connectivity, it also defines two topologies and several terminologies. Figure 1.1 illustrates the infrastructure and ad hoc topologies that are the two configurations provided by the IEEE 802.11 standard. In the infrastructure configuration, wireless terminals are connected to a backbone network through Access Points (APs). In the ad hoc configuration, terminals communicate in a peer-to-peer basis. The IEEE 802.11 MAC layer provides two basic access schemes: the Distributed Coordination Function (DCF) and Point Coordination Function (PCF) [CWK+97]. The DCF scheme is designed for use in the infrastructure and ad hoc configurations, while the PCF scheme is designed only for the infrastructure configuration. The DCF is the fundamental access method in IEEE 802.11, which utilizes Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism to support asynchronous data transfer across multiple stations. The PCF is proposed as an optional access method to support time-bounded services for IEEE 802.11 WLANs.

Since IEEE 802.11 WLAN is being accepted widely and rapidly for many different environments, it attracts many interests in some advanced issues. Many research and investigations are proposed to improve the IEEE 802.11 standard by considering real-time support, mobility, energy consumption, channel utilization, and so on. The 802.11 Working Group also established an activity (802.11e) to enhance the original 802.11 MAC protocol to support applications with QoS requirements [IEE03]. Moreover, this dissertation proposes a novel framework to provide priority access with high bandwidth utilization, and multihop transmission in WLANs.

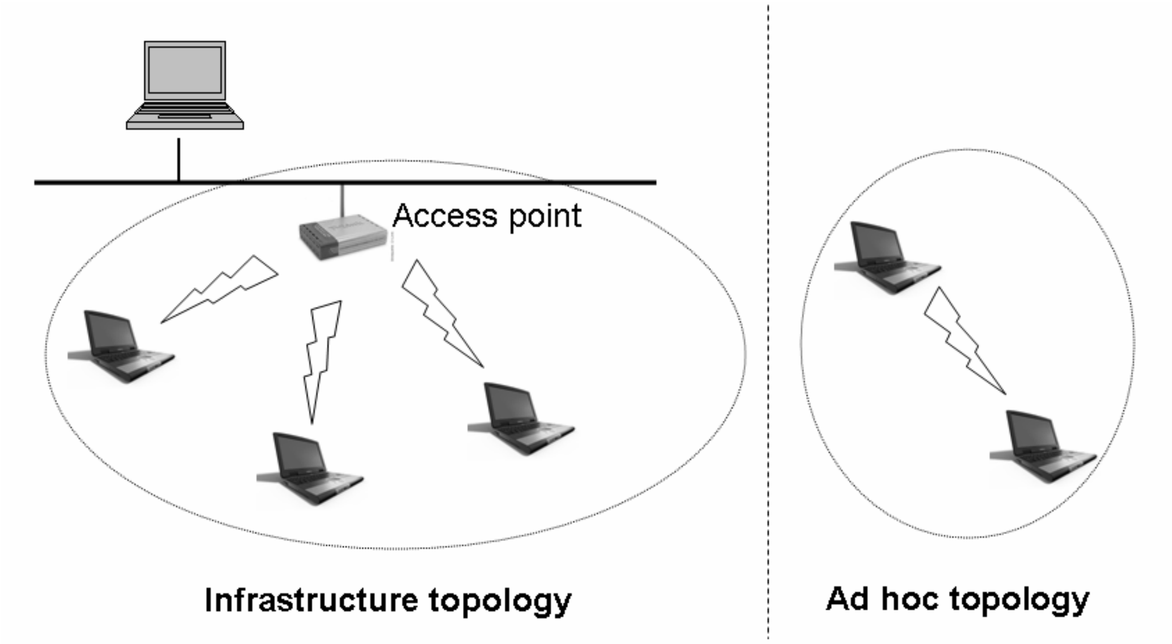

Figure 1.1: The Infrastructure and ad hoc topologies for the IEEE 802.11.

1.1 QoS Supports in WLANs

QoS in wireless networks is a challenging problem due to limited network bandwidth, terminal mobility, low power capability, and security problems. The DCF defined by the IEEE 802.11 Working Group (WG) is a contention-based protocol which uses CSMA/CA mechanism. Since wireless stations must contend to access the wireless medium in the DCF mode, the medium access delay for each station cannot be bounded during high load conditions. Therefore, the DCF can support only the asynchronous data transmission on a best-effort basis. On the other hand, the PCF let stations have priority access to the wireless medium, coordinated by a station called Point Coordinator (PC). Although the legacy PCF is primarily designed to support multimedia delivery in WLAN, it has a very limited QoS support due to polling overheads, null-packet slots, and unpredictable beacon delays [MCM+02][GW91]. As a result, IEEE 802.11 Task Group E defines enhancements to the above-described 802.11 MAC, called 802.11e, which introduces EDCF and HCF.

Although the characteristic of service difference can be observed in IEEE 802.11e, IEEE 802.11e does not provide satisfied performance to the real-time and voice services. The contention nature of EDCF and the polling overheads of HCF increase the access delay of real-time services. In fact, all contention-based MAC protocols have the same problem on supporting real-time services with their QoS requirements. The contention-based MAC protocols cannot exactly guarantee the QoS requirements of real-time services since the contention nature in these protocols causes much uncertainty. A high priority service may lose to a low priority service, and therefore the high priority service misses its delay constraint. Contention-based MAC protocols also confront collisions during the contention procedure. Collisions and the following exponential backoff procedures serious degrade the performance of time-bounded services. In addition, the MAC layer retransmissions of lost or corrupted data can be unnecessary or even harmful for real-time traffic with hard delay and jitter constraints since it introduces extra delay.

Besides the IEEE 802.11e, there are many dissertations [MCM+02, AKC00, ZF02, DC99, SK96, SK99, SS01, SLW+04, VBG00, IEE02, LAS01, HCG01, MS98, NO00] that propose distributed QoS MAC protocols based on contention scheme. These protocols (reviewed in Chapter 2) have the same problems on supporting real-time services as described above. Therefore, a scheduling-based centralized MAC protocol is proposed in this dissertation to precisely guarantee QoS requirements of services, and a multi-channel system is proposed to enhance this centralized MAC with multihop transmission capability. Moreover, to ensure that the services in relay stations meet their QoS requirements, the delay-sensitive fair queueing model is designed in this dissertation. The next section will briefly introduce our motivation in this dissertation.

1.2 Motivation

The MAC layer of the protocol stack significantly affects the performance of network applications and services. The major functions of a MAC protocol are to provide a delivery mechanism for user data, fair access control to the shared medium, and to protect the delivered data. Accordingly, the MAC protocol is a very crucial part in the data communication protocol stack. Moreover, since the demand for the transmission of real-time traffic has been growing significantly in recent years, a MAC protocol which can provide good QoS is urgently required.

This dissertation is motivated by the need to propose a prioritized MAC which supports multi-class services, to precisely guarantee the QoS requirements of services, and to provide excellent channel utilization in WLANs. Such a protocol should comprise the design of centralized framework, service classification, service scheduling, admission control, and request management. A multi-channel architecture is also required to enhance such a centralized MAC with the multihop transmission capability. Furthermore, the fair queueing algorithm inside stations and routers has to be redesigned to adapt these nodes to QoS oriented WLANs.

1.2.1 Scheduling-Based MAC Protocol

To well support multi-class services in WLANs, it is necessary to design a prioritized MAC with central coordination functions. Therefore we propose a scheduling-based MAC protocol in this dissertation. This protocol is a centralized MAC protocol which applies a coordinator node (CN) to coordinate the network. The proposed protocol uses a scheduling-based scheme instead of the polling-based scheme on managing transmissions to avoid the overheads of polling packets. In addition, to exploit the channel utilization, the proposed protocol allows peer-to-peer transmissions under the coordination of CN. Our protocol is composed of a slot-scheduling algorithm called *weighted scheduling* and a QoS enhanced admission control algorithm. The weighted scheduling algorithm is designed to efficiently utilize the network bandwidth and fairly schedule the transmission for various services while the QoS-enhanced admission control algorithm is proposed to manage resources and guarantee the QoS requirements of services. An analytical model and an ns-2 module of the proposed protocol are developed to examine and evaluate the performance of the proposed protocol.

1.2.2 Multihop Mechanism

Since we proposes a centralized MAC protocol for supporting priority access in WLANs, the drawbacks of centralized MAC protocols should be taken into account and resolved. A significant drawback is that the centralized MAC protocol cannot be adopted in ad hoc networks, because it provides only single-hop transmission. Therefore, this dissertation proposes a multihop mechanism for centralized MAC protocols to operate on various network topologies. The proposed mechanism eliminates the restriction on single-hop transmission for centralized MAC protocols, provides excellent throughput for both inter-subnet and intra-subnet links, and alleviates the hidden terminal problem. It also eliminates the need for both complex initialization procedures and synchronization between subnets when initiating a network.

1.2.3 Delay-Sensitive Fair Queueing Algorithm

Fair scheduling algorithms have been proposed to tackle the problem of bursty and location-dependent errors in wireless packet networks. Most of those algorithms ensure the fairness property and guarantee the QoS of all services (sessions) in a large-scale cellular network such as 3G or GPRS. Since these algorithms are not sensitive to the growth of queue length, they cannot adapt to highly variant WLANs directly. To ensure that the services in relay stations meet their QoS requirements, the Weighted-Sacrificing Fair Queueing (WSFQ) model is proposed in this dissertation for delay-sensitive multihop WLANs. WSFQ slows down the growth of queue length for real-time traffic, still maintains the property of fairness, and guarantees the throughputs of the station. Moreover, WSFQ can easily adapt itself to various traffic loads. Since the WSFQ is an abstract model, this dissertation proposes a packet-based scheduling algorithm, the Packetized Weighted Sacrificing Fair Queueing (PWSFQ), to approach the WSFQ in practice. To evaluate the performance of our models, WSFQ and PWSFQ are evaluated by mathematical analysis and simulations in this dissertation.

1.3 Summary of Contributions

The major contributions of this dissertation are summarized in the following.

- We propose a centralized MAC protocol for supporting multi-class services in WLANs. Rather than the polling-based scheme that is widely used in centralized MAC protocols, our protocol uses a scheduling-based medium access scheme to avoid polling overheads. This medium access scheme is able to efficiently utilize the network bandwidth and fairly schedule the transmission for various services.
- We design a QoS-enhanced admission control algorithm based on the exponential

bounded burstiness (EBB) model [YS93]. This algorithm is designed to manage resources, classify services, and guarantee the QoS requirements of services.

- We develop an analytical model and an ns-2 module for the proposed protocol. We use these models to evaluate the performance of the proposed protocol, and to compare the proposed protocol with other QoS-supported MAC protocols.
- We point out the challenges to apply centralized MAC protocols to ad hoc networks.
- We propose a multihop mechanism with a multi-channel architecture for centralized MAC protocols to eliminate the restriction on single-hop transmission. This multihop mechanism can be widely applied to various network topologies without complex configurations and initialization.
- We devise a novel fair queueing model for stations to well support time-bounded services in WLANs. We also construct a Stochastic Petri Net model for performance evaluation.

1.4 Organization

The rest of this dissertation is organized as follows. Chapter 2 reviews the background and related work on the QoS support in WLANs. Chapter 3 proposes a scheduling-based MAC protocol for supporting multi-class services in WLANs. Chapter 4 devises a multihop mechanism with a multi-channel architecture for centralized MAC protocols. Chapter 5 constructs a delay-sensitive fair queueing model for relay stations in WLANs. Chapter 6 concludes this dissertation.

Chapter 2

Related Work

In this chapter, we review the related work reported in the dissertation. We first introduce the IEEE 802.11 family and other MAC protocols with QoS supports in WLANs. We also introduce some multi-channel mechanisms that are designed for multihop transmissions in WLANs. The traditional fair queueing algorithms are briefly reviewed in the end of this chapter.

2.1 IEEE 802.11 Standard

Here we briefly summarize the 802.11 MAC protocol (version b and e) by referring the previous dissertation [MCM+02]. We consider an infrastructure Basic Service Set (BSS) of IEEE 802.11 WLAN, which is composed of an AP and a number of stations associated with the AP. The AP connects its stations with the infrastructure.

2.1.1 IEEE 802.11 DCF

The basic 802.11 MAC protocol is the Distributed Coordination Function (DCF) that works as listen-before-talk scheme, based on the CSMA/CA mechanism. Stations deliver MAC Service Data Units (MSDUs) of arbitrary lengths after detecting that there is no other transmission in progress on the wireless medium. However, if two or more stations detect the channel as idle state and transmit data packet at the same time, collisions occur. The DCF defines a Collision Avoidance (CA) mechanism to reduce the probability of such collisions. As part of CA, before starting a transmission a station should perform a backoff procedure. It has to keep sensing the channel for an additional random time after detecting the channel as idle state for a duration called DCF Interframe Space (DIFS). Only if the channel remains idle for this additional random time period, the station is allowed to initiate its transmission. The duration of this random time is determined as a multiple of a slot time. Each station maintains a so-called Contention Window (CW), which is used to determine the number of slot times a station has to wait before transmission.

For each successful reception of a data frame, the receiving station immediately acknowledges the reception by sending an acknowledgement frame (ACK). The CW size increases when a transmission fails, i.e., the transmitted data frame has not been acknowledged. After any unsuccessful transmission, another backoff is performed with a doubled size of the original CW. This reduces the collision probability in case there are multiple stations attempting to access the channel at the same time. The stations that deferred from channel access during the channel busy period do not select a new random backoff time, but continue to count down the time of the deferred backoff after sensing a channel as being idle again. In this manner, the stations, that deferred from channel access because their random backoff time was larger than the backoff time of other stations, are given a higher priority when they resume the transmission attempt. After each successful transmission, another random backoff is performed by the station that succeeds in transmitting data, even if there is no other pending MSDU to be delivered. This is called post-backoff.

There is one situation when a station is not required to perform the random backoff before starting data transmission. An MSDU arriving at the station from the higher layer may be transmitted immediately without waiting any time, if the last post-backoff has been finished already, i.e., the queue was empty, and additionally the channel has been idle for a minimum duration of DIFS. All the following MSDUs after this MSDU have to be transmitted after random backoff, until the transmission queue is empty again. To reduce the probability of long frames colliding and being transmitted more than once, data frames may also be fragmented. Via fragmentation a large MSDU can be divided into several smaller data frames which can be transmitted sequentially. By this way, a station may retransmit less data when failed transmissions occur.

To reduce the hidden terminal problem inherent in CSMA, 802.11 defines a Request-to-Send/Clear-to-Send (RTS/CTS) mechanism, which can be used optionally. In the RTS/CTS mechanism, a station ready for transmission sends a short RTS frame identifying the source address, destination address, and the length of the data to be transmitted. The destination station will respond with a CTS frame after a Short Interframe Space (SIFS) period. The source station receives the CTS and sends the data after another SIFS period. Other stations hearing RTS/CTS that is not addressed to them will go to the virtual carrier-sensing mode for the entire period identified in the RTS/CTS communication, by setting their Network Allocation Vector (NAV) signal on. Therefore, the source station sends its data frame with no contention. After completion of the transmission, the destination station sends an ACK frame, and the NAV signal is terminated, opening the contention for other stations. Figure 2.1 illustrates an example of the DCF with the RTS/CTS mechanism. It should be noticed that SIFS is shorter than DIFS, which gives the CTS and ACK the highest priority for access to the wireless medium.

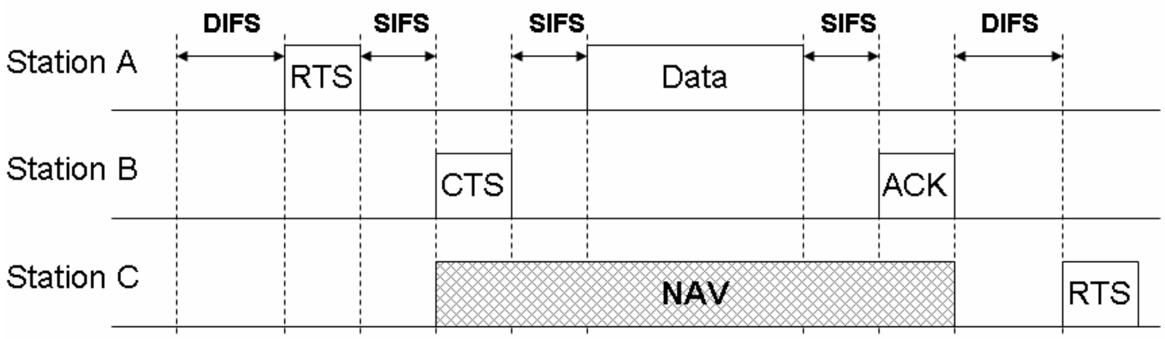

Figure 2.1: DCF timing with the RTS/CTS mechanism.

2.1.2 IEEE 802.11 PCF

To support time-bounded services, the IEEE 802.11 standard defines the PCF to let stations have priority access to the wireless medium, coordinated by a PC. The PCF has higher priority than the DCF, because it may start transmissions after a shorter duration than DIFS; this duration is called PCF Interframe Space (PIFS), which longer than SIFS. Time is always divided into repeated periods, called superframes. With PCF, a Contention Free Period (CFP) and a Contention Period (CP) alternate over time, in which a CFP and the following CP form a superframe. During the CFP, the PCF is used for accessing the medium, while the DCF is used during the CP. It is mandatory that a superframe includes a CP of a minimum length that allows at least one MSDU Delivery under DCF.

A superframe starts with a so-called beacon frame. The beacon frame is a management frame that maintains the synchronization of the local timers in the stations and delivers protocol related parameters. The PC, which is typically collocated with the AP, generates beacon frames at regular beacon frame intervals, thus every station knows when the next beacon frame will arrive; this time is called target beacon transition time (TBTT) and is announced in every beacon frame. Stations are polled to access the medium in CFP. Figure 2.2 illustrates a typical sequence during CFP. The PC polls a station asking for a pending frame. Because the PC itself has pending data for this station, it uses a combined date and poll frame by piggybacking the CF-Poll frame on the data frame. Upon being polled, along with data, the polled station acknowledges the successful reception. If the PC received no response from a polled station after waiting for PIFS, it polls the next station, or ends the CFP. Thus no idle period longer than PIFS occurs during CFP. The PC keeps polling other stations until the CFP expires. A specific control frame, called CF-End, is transmitted by the PC as the last frame within the CFP to signal the end of the CFP.

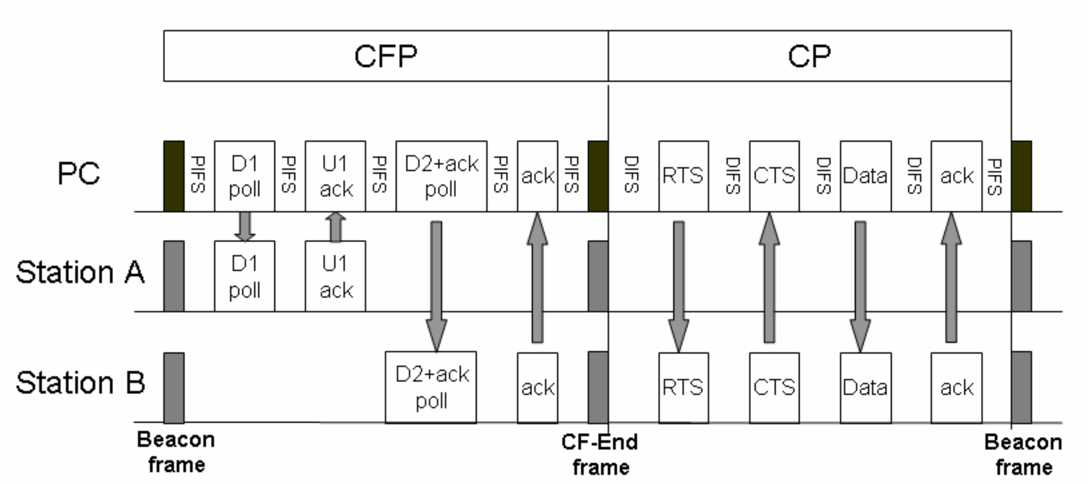

Figure 2.2: Example for the PCF operation.

Although the legacy PCF is primarily designed to support multimedia delivery in WLAN, it has a very limited QoS support due to the following facts: 1) the start of the CF period is not exactly periodic since it can only begin when the medium is sensed as being idle. As a result, the CFP may be forced to end prematurely not serving some members on the polling list; 2) all the CF-pollable stations have the same level of priority; 3) the CFP has to poll all the stations on its polling list, even if there is no traffic to be sent; 4) the data flow between two stations has to pass through the AP. Therefore, the access delay increases during the relay transmission.

2.1.3 IEEE 802.11e EDCF

The IEEE 802.11 TG E defines IEEE 802.11e [IEE03] to enhance the access mechanism of the above-described MAC with QoS supports. Since then, two new modes have been added to the original specifications: an Enhanced version of DCF called EDCF and a Hybrid Coordination Function called HCF. EDCF provides differentiated control of access to the medium. The 802.11e defines eight traffic categories for priority-based traffic and each QoS-capable station marks their packets to indicate a specific service requirement. Stations still contend for the medium but Arbitration Interframe Space (AIFS), minimum contention window (CW_{min}), and maximum contention window (CW_{min}) differ from one Traffic Class (TC) to another (Figure 2.3). The QoS-capable AP can dynamically adjust the contention window parameters as well as the Transmit Opportunity (TxOP) limit for each traffic category.

EDCF provide relative QoS difference among traffic classes but it does not provide any QoS guarantee. In other words, a traffic contract for a connection is only an objective that the wireless network will attempt to honor as often as possible over the lifetime of the connection. EDCF is relatively simple but the performance provided by this scheme is obviously less predictable than a reservation-based method and may also suffer from network congestion. Its contention nature and the fact that a station does not have to go through admission control to get bandwidth, implies that it is nearly impossible to reduce the amount of traffic that one station might send. The AP can only adjust the contention window and TxOP duration for each TC.

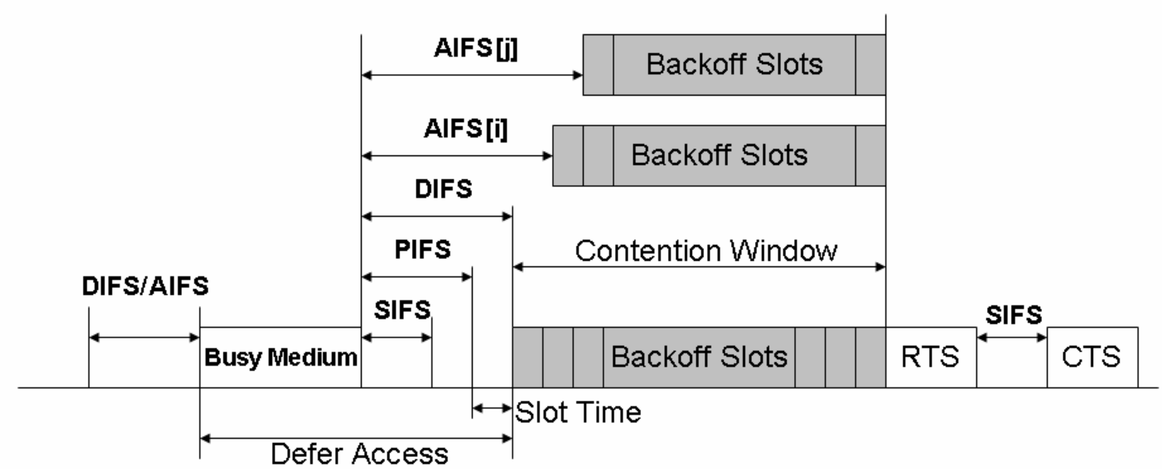

Figure 2.3: Multiple Backoff and Interframe Space in EDCF.

2.1.4 IEEE 802.11e HCF

The HCF is a new form of PCF that also runs above EDCF, using the highest medium access priority (PIFS) to gain access to the medium. The Hybrid Coordinator (HC) allocate TxOPs to stations during both the CP and CFP periods, through centralized scheduling of the radio interface, taking into account the traffic contract and QoS requirements of each active connection. By knowing the amount of pending traffic belonging to different traffic flows, the HC can adjust its scheduling accordingly and provides QoS guarantees. Additionally, a TxOP could also be started by a station after successfully gaining access to the medium. Therefore, 802.11e combines some polling and random access techniques to coordinate packet transmissions.

2.2 QoS MAC Protocol in WLANs

The Soft Reservation Multiple Access with Priority Assignment (SRMA/PA) protocol is proposed by Ahn et al. [AKC00] for supporting integrated services of real-time and non-real-time applications in mobile ad hoc networks. The SRMA/PA protocol adopts a simple frame structure that allows the distributed stations to contend for and reserve for time slots with a RTS/CTS-like handshake and soft reservation mechanism. Soft reservation is a unique concept that allows any urgent station of real-time application to snap the radio resource from other station of the non-real-time application on a demand basis. A slot is divided into six different fields: SYNC, SR (Soft Reservation), RR (Reservation Request), RC (Reservation Confirm), DS (Data Sending), and ACK fields. The SYNC field is used for synchronization on a slot basis. The SR, RR, RC, and ACK fields are intended for the corresponding control packets while the DS field is used for transmission of data packets. For the delay-sensitive voice application, voice stations snap the slots already reserved by data stations via RR field when their priorities are greater than those of data stations.

L. Zhao and C. Fan [ZF02] proposed Modified PCF (M-PCF) protocol to enhance the original PCF protocol with performance issues. In M-PCF, stations access the channel in a hub-poll manner. During CP, if new stations contend for the channel successfully, the PC adds them into its polling list, assigns the polling sequence for them, and broadcast the sequence to all stations. During CFP, the PC polls the first station in the polling list. Then, the second station is allowed to access automatically after the first one without being polled. Moreover, since the real-time service is sensitive to delay but not packet drop probability, and the non-real-time service is sensitive to packet drop probability but not delay, M-PCF does not reply the ACK frame upon the receipt of a real-time packet; it only replies the ACK frame upon the receipt of a non-real-time packet.

Deng and Chang [DC99] proposed a multi-priority scheme for IEEE 802.11 DCF protocol by modifying the original CSMA/CA mechanism. The basic idea of the proposed scheme is that priority access to the wireless medium is controlled through the use of different IFS and backoff time between the transmission frames. The shorter IFS/backoff time a station uses, the higher priority this station will get.

Shobrinho and Krishnakumar [SK96, SK99] proposed a modified CSMA/CA scheme that provides QoS supports for mobile ad hoc networks. The authors elaborate on the black-burst (BB) contention mechanism. With this mechanism, real-time stations contend for access to the common radio channel with pulse of energy (BB). The length of BB is proportional to the time that the station has been waiting for the channel to become idle. After a station transmits its BB, this station waits for a certain period to determine whether any other station is transmitting a longer BB. If the channel is idle, then this station begins to transmit its data. This method gives priority access to real-time traffic and ensures collision-free transmission of real-time packets.

Sheu and Sheu [SS01] proposed the distributed bandwidth allocation/sharing/extension DBASE protocol to support multimedia traffic over ad hoc networks. Both the IFS and contention window of a real-time service is smaller than that of a non-real-time service. By this way, the priority access is achieved. In addition, a station that successfully accesses the channel will join a reservation table and do not need to join the contention again. Moreover, the proposed protocol uses a repeat interval, D_{max}, to specify the smallest maximal tolerance delay of all active real-time services. The real-time stations that stay in the reservation table transmit data frames in turns within the D_{max} period. When the D_{max} period expires, all non-real-time stations can contend for the channel after the channel is idle for a DIFS period.

Sheu et al. [SLW$^+$04] proposed a priority-based MAC protocol that provides multiple priority levels. The priority-based protocol adopts the BB mechanism to separate high priority stations from low priority stations. By this way, the protocol guarantees that high priority packets are always transmitted earlier than low priority packets. An ID initialization mechanism is also used in this protocol to schedule the transmission order of those stations with the same priority. The stations with the same priority then transmit in a round robin manner. The advantage is that the proposed protocol can save bandwidth and time because stations can transmit their data consecutively according to their IDs without involving any contention resolution.

Vaidya et al. [VBG00] proposes an access scheme which utilizes the ideas behind fair queueing in the wireless domain, called Distributed Fair Scheduling (DFS). DFS uses the backoff mechanism of IEEE 802.11 to determine which station should send first. The backoff interval will be longer the lower the weight of the sending station is, so differentiation will be achieved, while fairness is achieved by making the interval proportional to the packet size.

2.3 Multi-Channel Architecture

By exploiting multiple channels, we can achieve a higher network throughput than using one channel, because multiple transmissions can take place without interfering. There are many related papers that study the benefit of using multi-channel architecture. Most of them focus on either the multi-channel design for distributed ad-hoc networks or optimizing the system utilization by parallel transmission using multiple channels in a subnet. Only a few papers concern the multi-channel architecture for multihop transmission in centralized wireless networks. The following briefly introduces these investigations.

Dual Busy Tone Multiple Access [DH98] divides a common channel into two sub-channels, one data channel and one control channel. Busy tones are transmitted on a separate control channel to avoid hidden terminals, while data is transmitted on the data channel. This scheme uses only one data channel and is not intended for increasing throughput using multiple channels.

Hop Reservation Multiple Access [TG99] is a multi-channel protocol for networks using slow frequency hopping spread spectrum (FHSS). The stations hop from one channel to another according to a predefined hopping pattern. When two stations agree to exchange data by an RTS/CTS handshake, they stay in a frequency hop for communication. Other stations continue hopping, and more than one communication can take place on different frequency hops. Receiver Initiated Channel-Hopping with Dual Polling [TG01] takes a similar approach, but the receiver initiates the collision avoidance handshake instead of the sender.

Nasipuri et al. [NZD99] proposed a multi-channel CSMA protocol with soft channel reservation. If there are *n* channels, the protocol assumes that each station can listen to all *n* channels concurrently. A station wanting to transmit a packet searches for an idle channel and transmits on that idle channel. Among the idle channels, the one that was used for the last successful transmission is preferred. In [ND00] the protocol is extended to select the best channel based on signal power observed at the sender. These protocols require *n* transceivers for each station.

Wu et al. [WLT+00] proposed a protocol that assigns channels dynamically, in an on-demand style. In this protocol, called Dynamic Channel Assignment (DCA), they maintain one dedicated channel for control messages and other channels for data. Each station has two transceivers, so that it can listen on the control channel and the data channel simultaneously. RTS/CTS packets are exchanged on the control channel, and data packets are transmitted on the data channel. In RTS packet, the sender includes a list of preferred channels. On receiving the RTS, the receiver decides on a channel and includes the channel information in the CTS packet. Then, DATA and ACK packets are exchanged on the agreed data channel. Since one of the two transceivers is always listening on the control channel, multi-channel hidden terminal problem does not occur. This protocol does not need synchronization and can utilize multiple channels with little control message overhead. Jain and Das [JD01] proposed a protocol that uses a scheme similar to [WLT+00] that has one control channel and *n* data channels, but selects the best channel according to the channel condition at the receiver side. The protocol achieves throughput improvements by intelligently selecting the data channel.

So and Vaidya [SV04] proposed a multi-channel MAC which operates with one transceiver per station. The protocol does not require a dedicated control channel. Instead, it requires clock synchronization among all the stations. At the start of each interval, the protocol requires all stations to listen to a common channel in order to exchange traffic indication messages. During this interval stations do not exchange data packets. So this duration of time is an overhead in this protocol.

Raniwala et al. [RGC04] proposed a multi-channel wireless mesh network architecture in which each station is equipped with multiple IEEE 802.11 interfaces, presented the research issues involved in this architecture, and demonstrated through an extensive simulation study the potential gain in aggregate bandwidth achievable by this architecture. They also developed two channel assignment and bandwidth allocation algorithms for the proposed multi-channel wireless mesh networks. The first algorithm, Neighbor Partitioning Scheme, performs channel assignment based only on network topology. The second algorithm, Load-Aware Channel Assignment, reaps the full potential of the proposed architecture by further exploiting traffic load information. Even with the use of just two NICs per station, the two algorithms improve the network cross-section goodput by factors of up to 3 and 8 respectively.

2.4 Wireless Fair Queueing Algorithm

Packet Fair Queueing (PFQ) algorithms are first proposed in the context of wired networks to approximate the idealized Generalized Processor Sharing (GPS) policy [DKS89, PG93]. GPS has been proven to have two important properties: 1) it can provide an end-to-end bounded delay service to a leaky-bucket constrained session; 2) it can ensure fair allocation of bandwidth among all backlogged sessions regardless of whether or not their traffic is constrained. However, it is inappropriate to directly apply GPS and the corresponding algorithms to wireless networks because these algorithms assume that all transmitted packets are received correctly, which is not true in a wireless network. To provide fairness guarantees similar to that in wired networks while making efficient use of wireless bandwidth, many investigations were proposed in recent years. The following briefly introduces these contributions.

Lu et al. [LBS99] proposed a fair scheduling policy for wireless networks which tries to approximate the WFQ policy [DKS89] of the wired network. To make efficient use of the wireless bandwidth, the scheduling policy defers transmission of packets from sessions with noisy channel. To guarantee fairness, these sessions are later supplemented with additional bandwidth once the quality of their channel improves. To determine how much bandwidth a session is to be supplemented, the scheduler simulates an error-free model in real-time, and compares the actual amount of service received by a session to that it has received in simulation. Sessions which have received more service than in simulation are said to be leading; sessions which have received less service than in simulation are said to be lagging. Leading sessions are forced to give up a part of their guaranteed bandwidth to lagging sessions to ensure long-term fairness among all sessions.

Ng et al. [NSZ98] identified a set of properties, called Channel Condition Independent Fair (CIF), desirable for any PFQ algorithm in a wireless network: 1) delay and throughput guarantees for error-free sessions; 2) long term fairness guarantee among error sessions; 3) short term fairness guarantee among error-free sessions; 4) graceful degradation in quality of service for sessions that have received excess service. They also presented a methodology for adapting PFQ algorithms for wireless networks and applied this methodology to derive a scheduling algorithm called the Channel-condition Independent packet Fair Queueing (CIF-Q) algorithm that achieves the CIF properties. They proved that CIF-Q achieves all the properties of the CIF and showed that it has low implementation complexity.

Jeong et al. [JMA01] proposed Wireless General Processor Sharing (WGPS) as a wireless fair scheduler and Packetized WGPS (PWGPS) as a packetized algorithm of WGPS. WGPS is an extension of GPS for dealing with burst and location-dependent channel errors of wireless networks. WGPS operates different from GPS only when there is a flow suffering channel errors. While a flow is in burst error state, it is excluded from scheduling by WGPS. When the flow’s channel has recovered afterwards, the flow is compensated by the increased service share until the additional service becomes equal to the lost service. In order to prevent the service degradation of error-free flows during compensation, the bandwidth for the increased service shares is pre-allocated for compensation. In WGPS, fairness is also guaranteed among the flows being compensated, since their service shares are increased by the same factor.

Ramanathan et al. [RA98] proposed an approach called Server Based Fairness Approach (SBFA) which can be integrated with any PFQ policy designed for wired networks. The basic idea of SBFA is to compensate the sessions which have received reduced goodput due to poor channel condition. To keep track of the amount of bandwidth a session is to be compensated, SBFA creates one or more special sessions called Long-Term Fairness Servers (LTFSs). The bandwidth allocated to a LTFS is used to compensate lagging sessions in such a way that fairness guarantees are provided to all sessions over a long time period.

Chapter 3

Quality of Service Supports in WLANs

In this chapter, we propose a novel MAC protocol based on scheduling mechanism for providing QoS supports in wireless local area networks. The proposed protocol is a centralized MAC protocol which applies a coordinator node (CN) to coordinate the network. Our protocol is composed of a slot-scheduling algorithm called *weighted scheduling* and a QoS enhanced admission control algorithm. The weighted scheduling algorithm is designed to efficiently utilize the network bandwidth and fairly schedule the transmission for various types of services while the QoS-enhanced admission control algorithm is proposed to manage resources and guarantee the QoS requirements of services. We make mathematical analysis for the protocol parameters and compare our protocol with IEEE 802.11e EDCF and IEEE 802.11 PCF in terms of throughput and delay by conducting simulations. The experimental results show that our protocol has the best throughput and delay performance among the simulated protocols. Moreover, our protocol has much fewer collisions than IEEE 802.11e EDCF has and always keeps on excellent performance even in the high-loaded network.

3.1 Overview

In recent years, major interest has been focused on the design of the wideband communication on the wireless networks [FHF02]. As the volume of traffic over a wireless network increases, so does the need for an efficient and robust QoS technology, which will be essential when multiple services are multiplexed into the same radio access technology. There has been a number of MAC protocols proposed to handle prioritized and parameterized QoS-based traffic in recent years [MCM+02, AKC00, ZF02, DC99, SK96, SK99, SS01, SLW+04, VBG00, IEE97, IEE02, LAS01, HCG01, MS98, NO00]. These MAC protocols can generally be divided into two categories: centralized and distributed.

Centralized access schemes rely on a multiple access mechanism to coordinate the transmission of stations. Examples include time division multiple access (TDMA), frequency division multiple access (FDMA), and code division multiple access (CDMA), where stations must reserve time slots, frequencies, and codes, respectively, to transmit their data. Polling is also a centralized scheme, where one common channel is shared by all stations but a station has right to use the channel only after it is polled by the coordinator. IEEE 802.11 PCF [IEE97], IEEE 802.15.3 [IEE02], and M-PCF [ZF02] are the examples of polling-based protocols. A drawback of these polling-based protocols is that some bandwidth is wasted due to polling overheads. Moreover, most peer-to-peer traffic has to be relayed by the Access Point (AP) to arrive to destinations; this will deteriorate the throughput when lots of peer-to-peer connections exist in the system.

Distributed access schemes, such as IEEE 802.11e EDCF [IEE03], Distributed Fair Scheduling (DFS) [VBG00], SRMA/PA [AKC00] and Blackburst [SK99] are contention-based schemes which use the CSMA/CA protocol. These schemes provide relative QoS differentiation among traffic classes but they do not provide any QoS guarantees. The performance provided by a contention-based scheme is obviously less predictable than a reservation-based scheme and may also suffer from network congestion.

For the purpose of providing prioritized medium access in WLANs, we propose a novel MAC protocol based on centralized access schemes. This novel protocol uses a scheduling-based mechanism instead of the polling-based mechanism on managing transmissions to avoid the overheads of polling packets. In addition, we propose the peer-to-peer communication architecture rather than the relay-based communication architecture to exploit the system bandwidth. Our protocol is composed of a slot-scheduling algorithm called *weighted scheduling* and a QoS enhanced admission control algorithm. The weighted scheduling algorithm aims to efficiently utilize the network bandwidth and fairly schedule the transmission for various types of services while the QoS-enhanced admission control algorithm aims to manage resources and guarantee the QoS requirements of services. Furthermore, our protocol can avoid the starvation of low-priority services which often occurs in contention-based protocols. By the mathematic analysis and experiments described in the later sections, we will show that our protocol provides service difference distinctly for various types of services and achieves excellent performance in high-loaded networks.

The rest of this chapter is organized as follows. In Section 3.2, the randomized initialization procedure (RIP) [NO00] is reviewed. In Section 3.3, a novel MAC protocol with one primary slot-scheduling algorithm and two extended slot-scheduling algorithms is proposed. The QoS-enhanced admission control algorithm of the proposed protocol is presented in section 3.4. The mathematical analysis for the protocol-related parameters is made in section 3.5. Section 3.6 compares the performance of our protocol with the performance of other MAC protocols such as IEEE 802.11e EDCF and IEEE 802.11 PCF by conducting simulations. Conclusion remarks are given in section 3.7.

3.2 Randomized Initialization Protocol

We briefly introduce the Randomized Initialization Protocol [NO00] which is used as the contention-based access technology in our protocol. In the Randomized Initialization Protocol, each node is assumed to have the collision detection (CD) capability, where the node is the source / destination of data streams. By CD capability, a node is able to determine the current channel status: silence, collision (there are multiple transmission), or busy (there is only one transmission). However, in many practical situations, especially in the presence of noisy channels, collision detection is rather hard to perform. Therefore, the Randomized Initialization Protocol elects a leader which informs all nodes (no CD capability) about the channel status at the cost of additional time slots.

In the randomized initialization protocol, all nodes are treated fairly during contest. A binary tree called contention tree is constructed in the protocol. A node can grab a unique ID number corresponding to its position in the contention tree. All nodes have to send their request messages through a common channel. A node successfully grabs an ID when it is allowed to send a request without encountering collision. The *i*th node successfully sending its request grabs an ID number equal to *i*. If the node encounters collision, it will flip a fair coin. In case of head, the node will go into the left subtree belonging to the current tree node and continue to send the request in the next round. In case of tail, the node will go into the right subtree and wait until all processes in the left subtree terminate. Then it can join the contention again.

Figure 3.1 shows an example of the contention tree formed by five nodes, A, B, C, D and E. Initially, all nodes stay in the root. At the first round, all nodes send their request messages concurrently. Because collision occurs, all nodes have to flip a fair coin. At the moment, B, D and E get heads and go into the left subtree; A and C get tails and go into the right subtree. In the second round, B, D and E will continue to send their request messages. Of course the collision occurs in this round. At this stage, B and D get heads, E gets tails. In round 3, B and D get heads again and the collision abides to round 4. Now suppose that B gets a head and D gets a tail. B will successfully send its request in round 5 and obtains an ID equal to 1. D and E will obtain their IDs in round 6 and 8, respectively. Then the subtree rooted by the tree node 2 terminates. It should be noticed that the channel in round 7 is silence since no node goes into the right subtree rooted by node 3. In round 9, the right subtree rooted by the root node begins, which implies A and C will rejoin the contention to send their requests. The process will repeat recursively until all nodes obtain their ID numbers. In summary, a collision increases the depth of the contention tree, while a successful transmission or a silent status terminates a subtree. All nodes know their current positions in the contention tree by referring the information piggybacked in request messages.

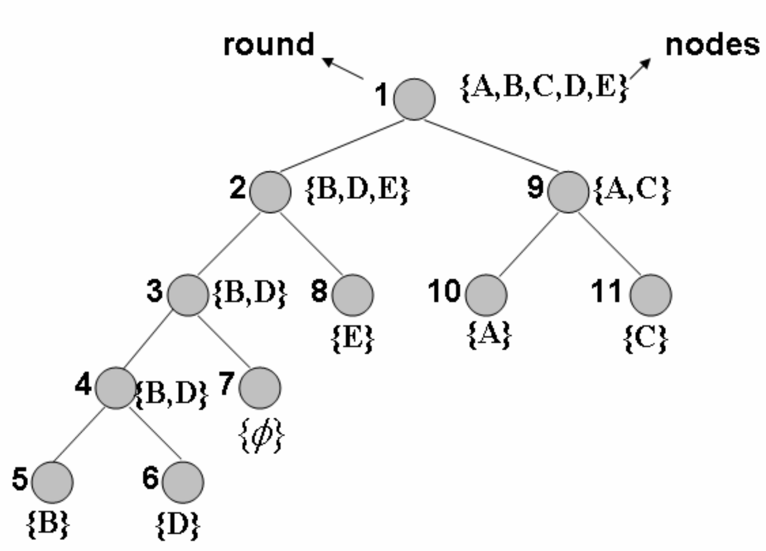

Figure 3.1: A contention tree of 5 nodes.

A node without CD capability cannot distinguish the situation between “the channel is silent” and “collision occurs in the channel”. Therefore, to let the protocol work correctly in the practical situations where nodes have no CD capability, one node is elected as the leader in the network. For the leader, no further action is necessary when the channel is busy. However, when the channel is not busy (silent or collision), the leader and the nodes who transmit at the current time slot (perhaps no node transmits at the current time slot) will transmit together at the next time slot. If the channel is still not busy at the next time slot, the status of the current slot is collision; otherwise, the status of the current slot is silent. After the ID initialization period, the nodes who already received IDs can enter the transmission period. These nodes will transmit their data frames in an ascending order of their ID numbers. These data frames should be separated by a SIFS period, thus each node knows when to transmit by monitoring the network.

3.3 The Medium Access Control Protocol

The network topology considered in this chapter is based on a one-hop centralized piconet, as shown in Figure 3.2. Each node (station) in the topology can directly communicate with other nodes. Home networks and AP-based networks are typical examples of the single-hop topology. In a single-hop (centralized) environment, a central coordinator is proposed to absolutely guarantee the QoS requirements for services. Via the central coordinator, the information gather and admission control can be easily achieved, and the radio resource can be allocated to meet the QoS requirements with packet scheduling algorithms.

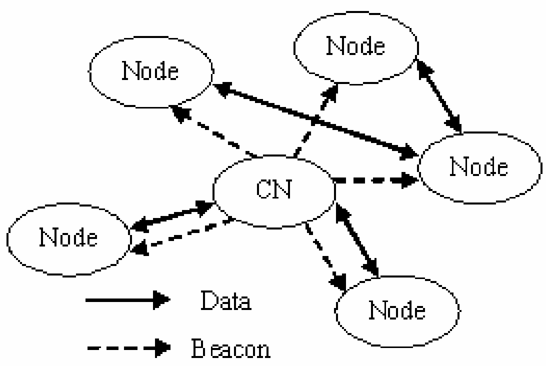

Figure 3.2: The network topology.

Most services in the networks can be categorized into three service types: the real-time, streaming, and background services. The background service is the fundamental service type of the Internet Protocol (IP). This service mode was originated in early Internet research projects [CK74] when the applications were relatively unsophisticated. The real-time service requires that the data transmitted by nodes can be received at the destinations within a certain time. If the data is received late, it is essentially worthless. The main characteristic of real-time services is the fixed maximum delay requirement. The streaming service, such as the service for multimedia streaming applications, is less strict than the real-time service but more important than the background service. This type of service offers an ideal trade-off between the utilization of resources and the provision of QoS guarantees. Occasionally violations to QoS guarantees are acceptable to the flows using this service class.

Our protocol follows the frame-based architecture, the most appropriate architecture for centralized access schemes. The CN coordinates the piconet by broadcasting beacon frames. The frame-based architecture of our protocol in the fully scheduling mode (other modes are optional and are described in section 3.3.2) is shown in Figure 3.3. The superframe beginning with a beacon frame is composed of a *transmission period* and a *request period*. The beacon frame is a management frame which maintains the synchronization of the local timers within the nodes and delivers protocol related parameters. The beacon frames are broadcast periodically at regular beacon frame intervals, thus every node knows when the next beacon frame will start. The transmission period is divided into times slots where the admitted services can transmit their data packets while the request period is used for submitting requests. All data stream transmitted between nodes are not necessary to get across CN; they can be directly transmitted under the coordination of CN.

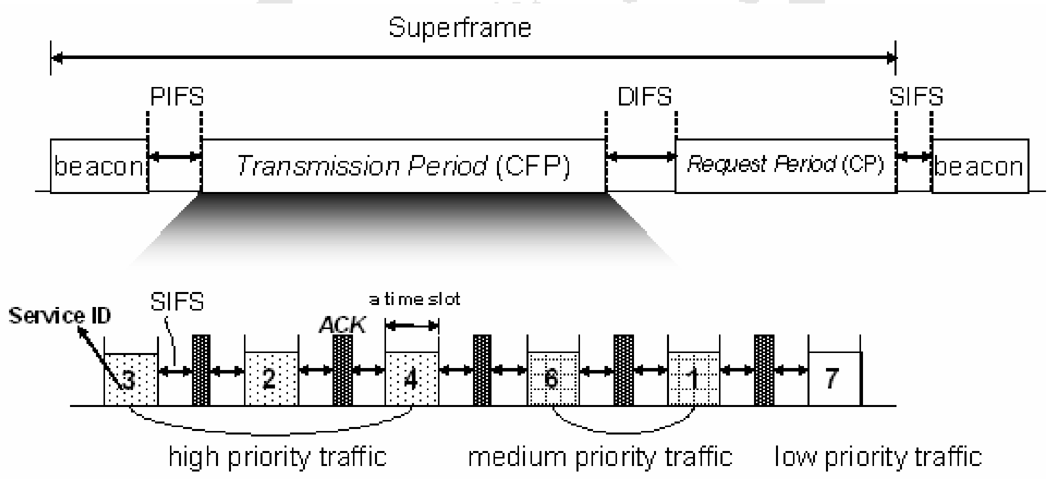

Figure 3.3: The superframe in our protocol operating in fully scheduling mode.

To accept as many requests as possible in the request period and reduce the delay of beacon frames, our protocol employees the ID initialization procedure of the randomized initialization protocol to manage the request behaviors. The contention tree of the randomized initialization protocol is adopted in the request period. An example shown in Figure 3.4 illustrates how services contend with each other to submit their requests by the contention tree. The coordinator has to send the beacon frame as a request when the time for the request period is expired. Every time the coin flipped by CN will be set to ‘1’ when CN wishes to broadcast its beacon frame to coordinate the next superframe. The service *C* in the example shown in Figure 3.4 is not able to submit its request due to the persistence of the beacon frame. In the practical situation, our protocol needs additional one slot for each time slot in Figure 3.4 to let a node without CD capability simulate the CD-capable node by utilizing the leader node (CN is usually elected as the leader node).

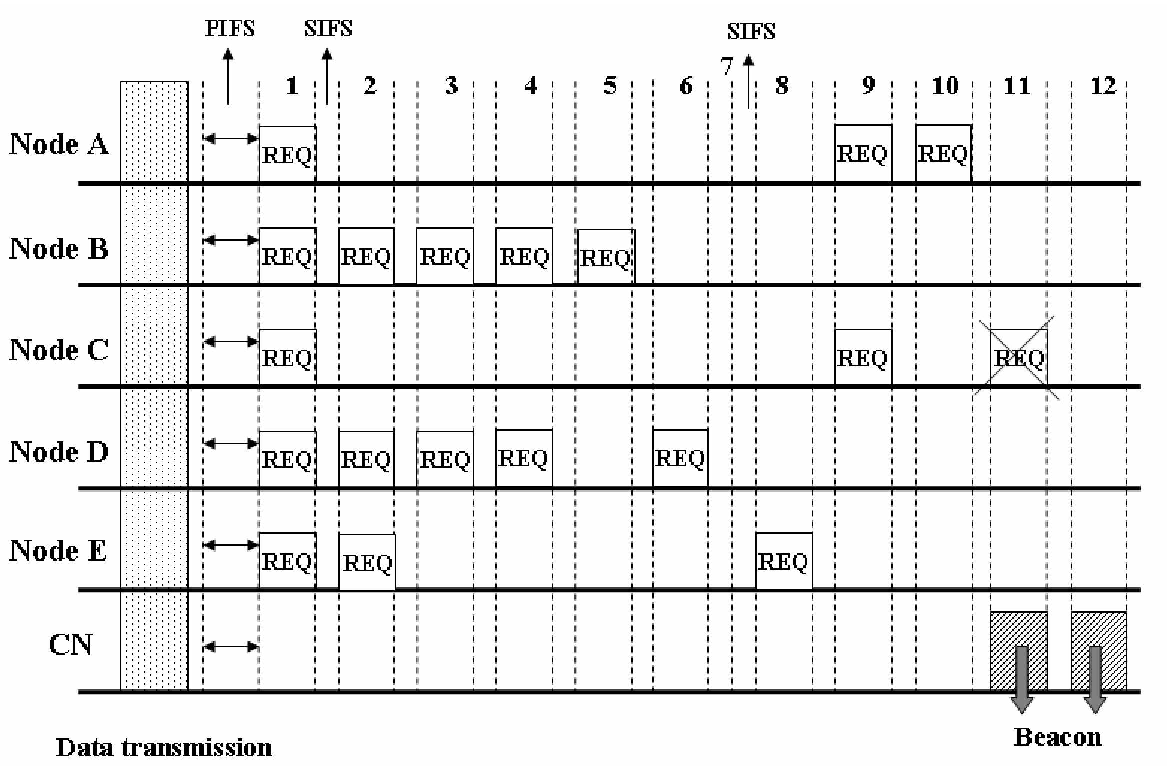

Figure 3.4: The ID initialization procedure based on Figure 3.1.

After the request period, CN decides which services are admitted to access the transmission period according to the QoS requirements of services and reserved bandwidth in the system. The admitted services are then scheduled into time slots by the weighted scheduling algorithm. The results of admission and scheduling are broadcast to each node via the next beacon frame. These admitted services transmit their data packets by following the order announced in the scheduling results while the rejected services keep on submitting their requests. The node receiving the data packet in the transmission period broadcasts an *ACK* frame to announce this reception. The termination of a service is also announced by the *ACK* frame when the receiving node receives the ending packet of a service. It should be noticed that only one packet can be transmitted in a time slot as well as the size of a packet is bounded.

3.3.1 Weighted Scheduling Algorithm

The protocol divides the total available bandwidth into several pieces with different sizes. The services of the same type contend with each other to access a certain piece. In general, the bandwidth reserved for the high-priority service is larger than the bandwidth reserved for the low-priority service. The nodes that attempt to transmit their services should first get admissions through the QoS-Enhanced admission control algorithm located in CN. The admitted services then get their weights (the procedure of weight assignment is described in Section 3.4). The proposed weighted scheduling algorithm is then used to maintain the weights of services and schedule the time slots in the transmission period for those services. The weights of the admitted services may change when a new request is accepted or a service is terminated.

The followings are the key features of the weighted scheduling algorithm.

(i) Spreading: the algorithm generates a slot allocation identical to WFQ [DKS89] or WF^2Q [BZ96].

(ii) Credit adjustment: CN broadcasts the credit information of all services via the beacon frame. Each service has an initial credit value equal to 0 and keeps the credit information of other services. When a service cannot transmit in its slot, the candidate service having the largest credit value is allowed to transmit in this slot. Then, the credit value of the original service is incremented and the credit value of the candidate service is decremented. If several services share the largest credit value, the one having the smallest ID number gets the slot (ID numbers are delivered by beacon frames). Every node should maintain the global credit information by analyzing the broadcast ACK frame.

If a service does not transmit in its slot, e.g. a service has no data on its queue, it has to broadcast a small-size dummy packet to announce this disability. Then a candidate service will utilize this time slot. However, if there is no service available in this slot, this slot will be skipped without utilization.

The following example shows how the weighted scheduling algorithm works in our protocol. Consider three services having weights 2, 2, 1, respectively. The first line of Figure 3.5 illustrates a basic slot allocation for a system with 5-slotted transmission period using the spreading method. Since service 3 has no data to transmit in slot *e*, as shown in the second line of Figure 3.5, service 1 transmits its data packet in slot *e* (because service 1 has the largest credit value). At the end of frame 1, service 3 increases its credit by 1 and service 1 decreases its credit by 1. Then, if service 2 cannot transmit in slot *i* in the next transmission period, service 1 transmits again in slot i (because service 1 has the smallest ID number). At the end of frame 2, the credit for service 2 is increased by 1 and that for service 1 is decreased by 1.

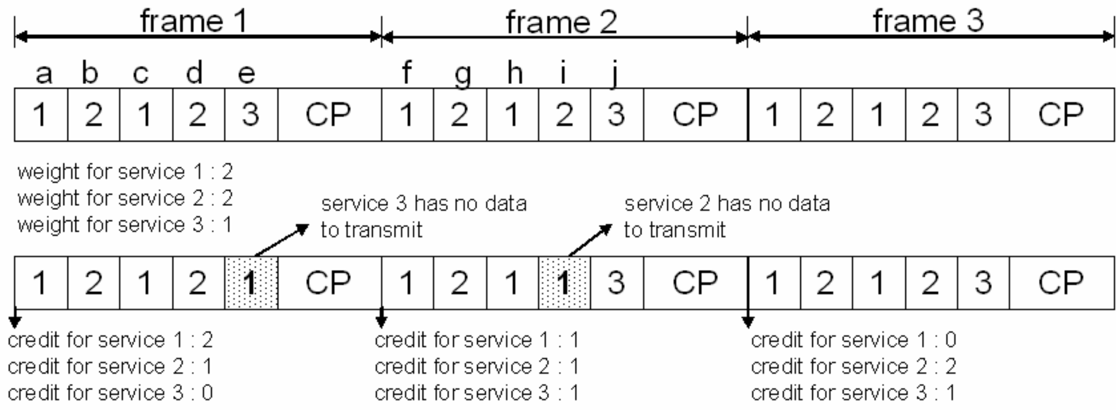

Figure 3.5: A scheduling example.

The weighted scheduling algorithm can adapt to extreme cases such as the case of 99% background traffic and 1% streaming traffic. Initially our protocol allocates a larger bandwidth to the 1% streaming traffic and allocates a smaller bandwidth to the 99% background traffic, the streaming traffic will digest only a few time slots. Then the remaining time slots left by the streaming traffic will be allocated to the best-effort traffic according to the weighted scheduling algorithm.

3.3.2 The Optional Modes

When our protocol operates in the optional mode, the *hybrid mode of scheduling and contention*, the transmission period is reserved for high-priority services only. Low-priority services have to contend with each other to access the request period. The frame-based architecture of this mode is shown in Figure 3.6. High-priority services seek admissions through the QoS-enhanced admission control algorithm. Then the admitted services are scheduled into time slots by the weighted scheduling algorithm.

It should be noticed that T_{TP}, the length of the transmission period in a superframe, is adjustable and is proportional to the traffic load of high-priority services. However, even there are too many high-priority services to be scheduled within a superframe, our protocol bounds the value of T_{TP} by the maximum length, T_{TP_MAX}, to guarantee the minimum amount of throughput for low-priority services. Accordingly, the starvation of the low-priority service can be avoided.

The original request period in the fully scheduling mode is divided into two periods with equal lengths. One is the request period in which high-priority services submit their requests, and the other is the *data transmission period* in which low-priority services transmit their data packets by the CSMA/CA manner. The request period is followed by the data transmission period. The ID initialization procedure is employed to arrange the request period. After sensing an idle period of SIFS in the data transmission period, CN broadcasts the beacon frame when the time for the contention period is expired.

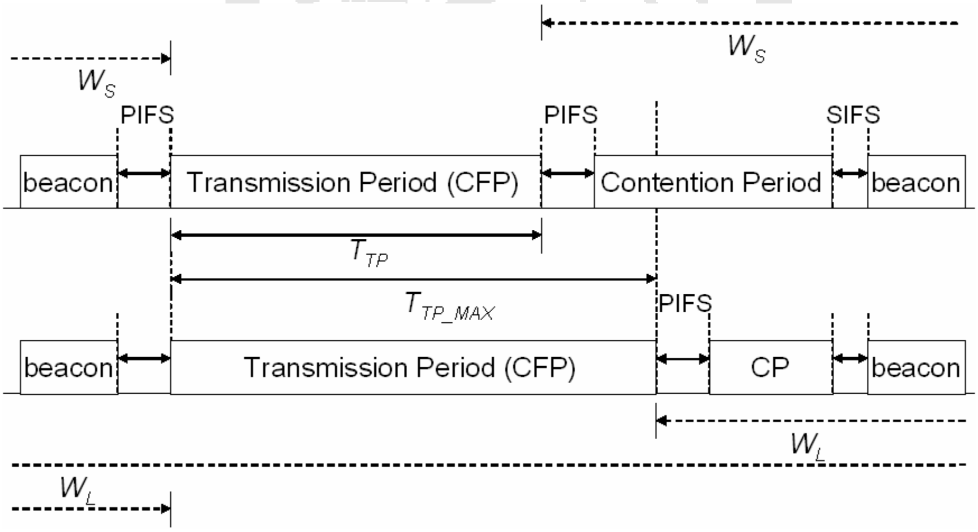

Figure 3.6: The superframe in our protocol operating in hybrid mode.

Another optional mode, the *advanced hybrid mode of scheduling and contention*, is based on the *hybrid mode of scheduling and contention* and is enhanced by the feedback information of services. Some specific low-priority services in this mode can be treated as high-priority services. We call such services the *dual-mode services*. This upgrade is triggered when the service delay of a dual-mode service exceeds the high threshold value H_{delay}. Whenever the service delay of an upgraded dual-mode service is below the low threshold value L_{delay}, this dual-mode service has to contend with other low-priority services in the data transmission period to transmit its data packets.

3.4 QoS-Enhanced Admission Control Algorithm

In our protocol, the transmission period allows services to access the medium without collisions in wireless networks. However, due to the limited bandwidth, only a finite number of services can be served in the transmission period. Thus a reliable and QoS-capable admission control algorithm is required for our protocol to utilize bandwidth and guarantee the QoS requirements of services. To this end, a QoS-enhanced admission control algorithm based on the exponential bounded burstiness (EBB) model is proposed in this section.

Suppose that each data stream of a service is an exponentially bounded burstiness source. The data streams follow the exponentially bounded bursty stochastic process studied by Yaron and Sidi [YS93] and satisfy the following property,

$$P[X[t_1,t_2] > \rho(t_2 - t_1 + 1) + \sigma] \le A\exp\{-\alpha\sigma\}, \tag{3.1}$$

where $X[t_1, t_2]$ represents the amount of data units generated by the source during the interval $[t_1, t_2]$, ρ is the average data rate associated with the source, and σ is the QoS parameter. Both A and α are constant values representing the characteristic of the stochastic process. α indicates the degree of decay in the process while A is used to tune the probability model. Therefore, an exponential bounded burstiness source can be expressed with the parameter (A, α, ρ). If we feed such a source to a constant rate server, the backlog $W[t]$ of the server can be given by

$$W[t] = X[t - b[t] - 1, t - 1] - R(b[t] + 1), \tag{3.2}$$

where $b[t]$, the duration of the current busy period, is the time elapsed since the buffer was last empty. R is the service rate of the server. The exponential bound of $W[t]$ derived by combining (3.1) and (3.2) satisfies

$$P[W[t] > \sigma] \le \frac{A\exp\{-\alpha(R-\rho)\}}{1-\exp\{-\alpha(R-\rho)\}}\exp\{-\alpha\sigma\}. \tag{3.3}$$

A service should provide its EBB parameter (A, α, ρ) to CN when seeking the admission. CN then makes decision whether this service is admitted to the transmission period according to the submitted parameters, the current distributions of the backlog, and the parameters of other admitted sources.

Consider the system in which n sources (S_1, S_2, ….., S_n) are fed to a constant rate server with FCFS mechanism, each source follows the exponential bounded burstiness property with the parameter (A_i, α_i, ρ_i). The sum of the n exponential bounded bursty sources has a backlog which is exponentially bounded with the parameter $(A_{sum}, \alpha_{sum}, \rho_{sum})$; this parameter satisfies:

$$A_{sum} = (A_1 + A_2 + \ldots + A_n)\cdot\frac{\exp\{-\alpha_{sum}(R-\rho_{sum})\}}{1-\exp\{-\alpha_{sum}(R-\rho_{sum})\}}, \tag{3.4}$$

$$\frac{1}{\alpha_{sum}} = \frac{1}{\alpha_1} + \frac{1}{\alpha_2} + \ldots + \frac{1}{\alpha_n}, \tag{3.5}$$

$$\text{and} \quad \rho_{sum} = \rho_1 + \rho_2 + \ldots.\rho_n. \tag{3.6}$$

That is, the backlog of the constant rate server has an exponential bound:

$$P[W[t] > \sigma] \le \frac{A_{sum}\exp\{-\alpha_{sum}(R-\rho_{sum})\}}{1-\exp\{-\alpha_{sum}(R-\rho_{sum})\}}\exp\{-\alpha_{sum}\sigma\}. \tag{3.7}$$

By utilizing the distribution of the backlog, the distribution of the packet-delay seen by an arriving packet can be obtained. The backlog seen by an arriving packet $W_p[t]$ is upper bounded as follows.

$$W_p[t] \le W[t] + X^{MAX}, \tag{3.8}$$

where $W[t]$ is the backlog of time-average and X^{MAX} is the maximum amount of traffic that could have arrived from all the sources prior to the packet. X^{MAX} is given as

$$X^{MAX} = R_{\Sigma} \times T_p . \tag{3.9}$$

The value R_{Σ} s usually the sum of the capacities of the input links. The value T_p is the upper bound of the time delay between the beginning of a frame and the transmission of the last packet belonging to this frame. According to (3.8), the delay seen by a packet can be derived as follows.

$$D_p[t] \le \frac{W[t] + X^{MAX}}{R} . \tag{3.10}$$

The probability that $D_p[t]$ is greater than a given value σ_D is bounded by P_D, which can be calculated by the following equation.

$$P[\sigma_D \le D_p[t]] \le P[\sigma_D \le \frac{W[t] + X^{MAX}}{R}] = P[W[t] \ge \sigma_D \cdot R - X^{MAX}] = P_D . \tag{3.11}$$

Substitute the value of $\sigma_D \cdot R - X^{MAX}$ for σ in (3.7), the upper bound of $P[\sigma_D \le D_p[t]]$ can be obtained (P_D can be obtained). The admission control algorithm makes decisions according to P_D and the delay requirements of all sources. The details of the decision procedure are described as follows.

When a new service i with the EBB parameter (A_i, α_i , ρ_i) seeks admission, it should also provide the delay requirement (σ_{D_i} , P_{D_i}). The admission control algorithm makes sure that admitting this service will not violate the negotiated delay requirements of other services, as well as that the delay requirement of the new service shall be met after the admission. For these considerations, the algorithm merges the EBB parameters of the old accepted services and the new requesting service by (3.4), (3.5), and (3.6) and calculates the delay bound P_D by (3.7) and (3.11). The algorithm checks whether this delay bound violates any delay requirement among the services. If this delay bound passes the delay requirements of all services, the new requesting service is admitted; otherwise, this service is rejected.

The QoS-enhanced admission control algorithm acts accordingly when our protocol operating in different modes. The details are described in the following.

(i) In fully scheduling mode:

All services should get admissions through the admission control algorithm to access medium. Suppose that the system can provide n types of services. Each service type is labeled as an integer m ($1 \le m \le n$) to represent its priority (a bigger number has higher priority than a smaller number has). The quota associated to each service type is set to be x^{m-1}, where x is a constant integer (larger than 1). The admission control algorithm provides each service type the bandwidth proportional to the quota of the service type. For example, there are three types of services in the system: real-time services (priority = 3), streaming services (priority = 2), and background services (priority = 1). Let R_{CFP} denote the bandwidth reserved for the transmission period. By assigning x the value of 3 in this example, the admission control algorithm first makes admissions for real-time services by the bandwidth equal to $9R_{CFP}$ / 13. Then the algorithm makes admissions for streaming services by the bandwidth equal to $3R_{CFP}$ / 13. At last the algorithm makes admissions for background services by the bandwidth equal to R_{CFP} / 13. After that, the admitted service is assigned a weight by multiplying the quota of its service type by the ratio of its data rate to the summation rates of the admitted services of the same type.

(ii) In hybrid mode of scheduling and contention:

In this mode, only high-priority services are allowed to access the transmission period; other low-priority services have to contend with each other to access the request period. Thus the admission control algorithm acts similarly as in fully scheduling mode. However, the bandwidth reserved for the transmission period in this mode is smaller than that in fully scheduling mode to provide more bandwidth for the request period. The parameter Δ upper bounded by $\Delta_{\max}$ is used to determine the ratio of the bandwidth reserved for the transmission period to the total bandwidth.

(iii) In advanced hybrid mode of scheduling and contention:

The difference between *hybrid mode* and *advanced hybrid mode* is that dual-mode services specified in advance in *advanced hybrid mode* can be upgraded to become high-priority services and are able to transmit their data packets during the transmission period. Thus the admission control algorithm should be capable of handling these upgrade behaviors.

When assigning quotas, we may consider the adaptive mechanism in which quotas are initially learned by the central controller based on the particular scenario. By this way, we are able to avoid the scheduling overhead caused by the bad quota assignment.

3.5 Performance Analysis

In this section, we address the issues of setting the protocol parameters in fully scheduling mode and provide mathematical analysis. These parameters are: 1) the length of superframe, 2) the maximum allowable service rate in the transmission period, 3) the maximum allowable requests in the request period, and 4) the system throughput.

(i) The length of superframe.

The length of superframe affects the time required for a service to submit its request when this service attempts to begin its transmission. Let T_{frame} denote the length of superframe, T_{SIFS} denote the length of SIFS, T_{DIFS} denote the length of DIFS, T_{PIFS} denote the length of PIFS, and T_{beacon} denote the length of a beacon frame. Suppose that the arrivals of requests follow the uniform distribution. As shown in Figure 3.6, the longest waiting time W_L that an admitted service may wait to be served (an admitted service is deemed to be served when this service enters the transmission period) in fully scheduling mode is expressed as follows.

$$W_L = (T_{frame} - 2T_{PIFS} - T_{beacon} - T_{SIFS}) \cdot (1-\Delta) + T_{SIFS} + T_{frame} + T_{beacon} + T_{PIFS}. \quad (3.12)$$

The shortest waiting time W_S that an admitted service may wait to be served is expressed as follows.

$$W_S = T_{PIFS} + (T_{frame} - 2T_{PIFS} - T_{beacon} - T_{SIFS}) \cdot (1-\Delta) + T_{SIFS} + T_{beacon} + T_{PIFS}. \quad (3.13)$$

By substituting the variable T_{RP} for the value of $(T_{frame} - 2T_{PIFS} - T_{beacon} - T_{SIFS}) \cdot (1-\Delta)$, we can obtain the expected value of the waiting time $E[W]$ by operating $(W_L+W_S)/2$.

$$E[W] = T_{RP} + T_{SIFS} + T_{beacon} + \frac{3}{2}T_{PIFS} + \frac{1}{2}T_{frame}. \quad (3.14)$$

In *hybrid mode* and *advanced hybrid mode*, the above equation is also held while T_{RP} indicates a different time length ($T_{RP} = \frac{1}{2}(T_{frame} - 2T_{PIFS} - T_{beacon} - T_{SIFS}) \cdot (1 - \Delta)$).

According to (3.14), we know that a small T_{frame} makes the expected waiting time of a request shorter. However, a small T_{frame} causes the high frequency of arising request periods; this is inefficient when the requesting rate is low. Therefore, the length of superframe should be set as large as possible to complete in both respects of bandwidth utilization and requesting delay.

(ii) The maximum allowable service rate in the transmission period.

Let T_{com_i} denote the transmission time for the service i with data rate r_i within a superframe. T_{com_i} satisfies

$$\begin{aligned} T_{com\_i} &= n_i \cdot (2T_{SIFS} + \frac{H_{PH} + H_{MAC} + D_{\max}}{R_c} + T_{ack}) + 2T_{SIFS} \\ &+ (\frac{H_{PH} + H_{MAC} + T_{frame} \cdot r_i - n_i \cdot D_{\max}}{R_c}) + T_{ack} \end{aligned}, \quad (3.15)$$

where $n_i = \left\lfloor \frac{T_{frame} \cdot r_i}{D_{\max}} \right\rfloor$, R_c is the channel bit rate, D_{max} is the maximum packet size in a transmission, and T_{ack} is the length of a broadcast *ACK* in the transmission period . H_{PH} and H_{MAC} are used to denote the Physical layer and MAC layer headers. Suppose there are m connections accepted to be served in the transmission period. The summation of the service rates of these connections is denoted with R_{total}. The upper bound of R_{total} can be obtained by the following inequality.

$$\begin{aligned} &(T_{frame} - 2T_{PIFS} - T_{SIFS} - T_{beacon}) \cdot \Delta \\ &\quad \geq \sum_{i=0}^{m} T_{com\_i} \approx (n_{total} + 1) \cdot (2T_{SIFS} + \frac{H_{PH} + H_{MAC} + D_{\max}}{R_C} + T_{ack}), \end{aligned} \quad (3.16)$$

where $n_{total} = \left\lfloor \frac{T_{frame} \cdot R_{total}}{D_{\max}} \right\rfloor$. We simplify (3.16) by replacing

$(T_{frame} - 2T_{PIFS} - T_{beacon} - T_{SIFS}) \cdot \Delta$ with the variable T_{TP} to obtain the following inequality.

$$n_{total} + 1 = \left\lfloor \frac{T_{frame} \cdot R_{total}}{D_{\max}} \right\rfloor + 1 \le \left(\frac{T_{TP}}{2T_{SIFS} + \frac{H_{PH} + H_{MAC} + D_{\max}}{R_c} + T_{ack}} \right). \qquad (3.17)$$

Since $\frac{T_{frame} \cdot R_{total}}{D_{\max}} < \left\lfloor \frac{T_{frame} \cdot R_{total}}{D_{\max}} \right\rfloor + 1$, the upper bound of R_{total} can be derived as follows.

$$R_{total} \le \left(\frac{T_{TP} \cdot D_{\max}}{\frac{2T_{SIFS} + T_{ack}}{T_{frame}} + \frac{H_{PH} + H_{MAC} + D_{\max}}{R_c \cdot T_{frame}}} \right) = \left(\frac{R_c \cdot T_{frame} \cdot T_{TP} \cdot D_{\max}}{2R_c \cdot T_{SIFS} + R_c \cdot T_{ack} + H_{PH} + H_{MAC} + D_{\max}} \right) = R_{CFP}. \qquad (3.18)$$

This upper bound is further used in the QoS-Enhanced admission control algorithm described in Section 3.4.

(iii) The maximum number of allowable requests in the request period.

The maximum number of allowable requests in the request period is proportional to the length of the request period. According to the investigations by K. Nakano et al [NO00], $5.67n + O(\sqrt{n \ln n})$ time slots are required for randomized initialization protocol to initialize an n-station wireless network (with no-CD). Thus in a given request period T_{RP}, the following inequality can be obtained.

$$5.67n + k\sqrt{n \ln n} \cdot D_{req} \le T_{RP}, \qquad (3.19)$$

where both k and D_{req} are constant. D_{req} denotes the time required for transmitting a request packet. Then we obtain the following inequality according to (3.19).

$$5.67n + k\sqrt{n\ln 2}\cdot D_{req} \le 5.67n + k\sqrt{n\ln n}\cdot D_{req} \le T_{RP}\text{, when } n \ge 2. \tag{3.20}$$

Therefore, the upper bound of n can be derived from (3.20) as follows.

$$n \le \frac{100}{321489}(567T_{RP} + 50D_{req}{}^2k^2\cdot\ln 2 + 10\sqrt{567D_{req}{}^2k^2T_{RP}\cdot\ln 2 + 25D_{req}{}^4k^4\cdot(\ln 2)^2}. \tag{3.21}$$

According to (3.21), if the number of requests in the request period exceeds the value of $\left\lfloor \frac{100}{321489}(567T_{RP} + 50D_{req}{}^2k^2\cdot\ln 2 + 10\sqrt{567D_{req}{}^2k^2T_{RP}\cdot\ln 2 + 25D_{req}{}^4k^4\cdot(\ln 2)^2}\right\rfloor$, the request period will be too short to accept all requests.

(iv) The system throughput.

We analyze the throughput performance when the system operates under saturation conditions, i.e., each node always has a packet available for transmission. To derive the expression of the normalized system throughput, we first derive the bandwidth utilization U_{TP} on the transmission period. We assume that the packet size in a transmission follows the uniform distribution between the maximum value D_{max} and the minimum value D_{min}. Let D_p be the mean packet size equal to $(D_{max}+D_{min})/2$. Let n_{TP} denote the number of transmissions that can be performed during one transmission period; then we obtain the following equation.

$$n_{TP} = \frac{T_{TP}}{\frac{D_p}{R_c} + 2T_{SIFS} + T_{ACK}}. \tag{3.22}$$

By substituting the variable n_{TP}, U_{TP} can be derived as follows.

$$\begin{aligned} U_{TP} &= \frac{n_{TP}\cdot(D_p - H_{PH} - H_{MAC})}{T_{TP}\cdot R_c} = \frac{T_{TP}\cdot(D_p - H_{PH} - H_{MAC})}{(\frac{D_p}{R_c} + 2T_{SIFS} + T_{ACK})\cdot T_{TP}\cdot R_c} \\ &= \frac{D_p - H_{PH} - H_{MAC}}{D_p + R_c\cdot(2T_{SIFS} + T_{ACK})} \end{aligned}. \tag{3.23}$$

Therefore, the normalized system throughput U_{sys} is obtained as follows.

$$U_{sys} = U_{TP} \cdot \frac{T_{TP}}{T_{frame}}. \tag{3.24}$$

3.6 Experimentations

The goal of the experiments presented in this section is to demonstrate the performance of our protocol in terms of throughput, delay, and collision. We use the simulator ns-2 as the simulation tool to compare our protocol operating in fully scheduling mode with IEEE 802.11e EDCF and IEEE 802.11 PCF protocols.

We implement most of the functions in our protocol except the admission control algorithm. The bandwidth required in the request period is reserved but the admission control algorithm is simplified to have weight-assignment functions only. This simplification will not affect the comparison result since we assume all services in the experiments have no delay requirement (these services still have their priority-levels). With this assumption, the admission control is not necessary because it just accepts all services as well as what EDCF and PCF do. TKN IEEE 802.11e EDCF and CFB simulation model [TKN03] is used to simulate the EDCF protocol in the experiments. We make some modifications to this model to realize RTS/CTS mechanism. Lindgred's IEEE 802.11 PCF model [LAS01] is used to simulate the PCF protocol in the experiments. We also make some modifications to enable Lindgred's model to take over the traffic between two mobile nodes in PCF mode and to reduce the delay of the beacon arrival.

A piconet with one base station (CN) and twelve wireless nodes (node_1 to node_12) is simulated as the network topology. Each node is able to hear all other nodes and restore at most 50 packets in its queue. The total bandwidth is configured to be 2Mbps (we can easily load either high-loaded or low-loaded traffic patterns to the network by this ns-2 default setting). Three types of traffic including real-time, streaming, and background traffic are provided in this piconet. In terms of priority, the real-time traffic is superior to the streaming traffic while the streaming traffic is superior to the background traffic. Three real-time services generated by *Constant Bit Rate* (CBR) sources are simulated in the experiments, each of them has 200 Kbps bit rate (the total offered load of real-time traffic is 600 Kbps). Three streaming services generated by Exponential On-Off applications have the average burst time equal to 50 ms and average idle time equal to 10 ms, each of them has 240 Kbps in burst periods (the total offered load of streaming traffic is 600 Kbps).

Three FTP links are created to generate the background traffic, each link has 200 Kbps bit rate (the total offered load of background traffic is 600 Kbps). The source and destination nodes of each service are randomly chosen from the twelve nodes. The duration of a superframe in our protocol is 40 *ms* (the inter-arrival time of beacon frames is 40 *ms*), and the maximum duration of the transmission period is 40 * 0.9 = 36 *ms* (i.e., the minimum duration of the request period is 4 *ms*). Our protocol assigns the quotas equal to 1, 3, and 9 to the real-time, streaming, and background traffic respectively. Similarly, the duration of a superframe in PCF mechanism is 40 *ms*, and the maximum duration of the transmission period is 40 * 0.9 = 36 *ms* (i.e., the minimum duration of CP is 4 *ms*). The polling sequence in PCF mechanism follows the round-robin mechanism, the most widespread manner applied in polling algorithms. The configurations for the EDCF protocol and other common settings in the simulation are listed in Table 3.1.

Table 3.1: The EDCF settings and other common settings.

Priority (0>1>2>3)	PF	AIFS	CW_{min}	CW_{max}	TXOP_limit
Priority 0 (real-time)	2	16 *us*	7 slots	15 slots	0.003008
Priority 1 (streaming)	2	16 *us*	15 slots	31 slots	0.006016
Priority 2 (background)	2	24 *us*	31 slots	1023 slots	0
Slot time = 16 μs , SIFS time = 8 μs , PIFS time = 12 μs , DIFS time = 16 μs					

3.6.1 Throughput Performance

For each protocol, we run the simulation 20 times, each of 60 seconds. We record the mean value of the cumulative throughput (end-to-end throughput) every 5 seconds. We start all traffic at 3.0 s since ns-2 requires the first 1 to 2 seconds for network initialization and message passing. The cumulative throughput of the system when applying different protocols is shown in Figure 3.7. We find that our protocol (denoted as “WS” in figures) has the highest throughput while IEEE 802.11 PCF protocol has the worst performance. Our protocol is superior to EDCF because neither collision nor backoff procedure takes place in our protocol during the transmission period, while EDCF suffers from collisions and backoff procedures all the time. The PCF protocol has low throughput due to the polling overhead. Moreover, since a packet in PCF has to be transmitted twice to reach the destination (relayed by CN), the PCF protocol requires more bandwidth and a larger queue in CN to achieve the same performance that WS and EDCF provide.

Figure 3.8 illustrates the throughput performance service by service when applying different protocols. We find that the service difference in EDCF is most distinct while no service difference can be found in PCF due to the round-robin polling mechanism. Moreover, we find that our protocol is much superior to EDCF when considering low priority services, while our protocol only leads EDCF a little when considering high priority services. It is because, in EDCF, low priority services always have large backoff time and AIFSs. As a consequence, low-priority services are easy to be starved. However, in our protocol, even a service type is assigned with a low quota, the services belonging to this service type can be guaranteed to transmit a certain amount of data according to the quota. This guarantee lets our protocol be much superior to EDCF from the view point of low priority services.

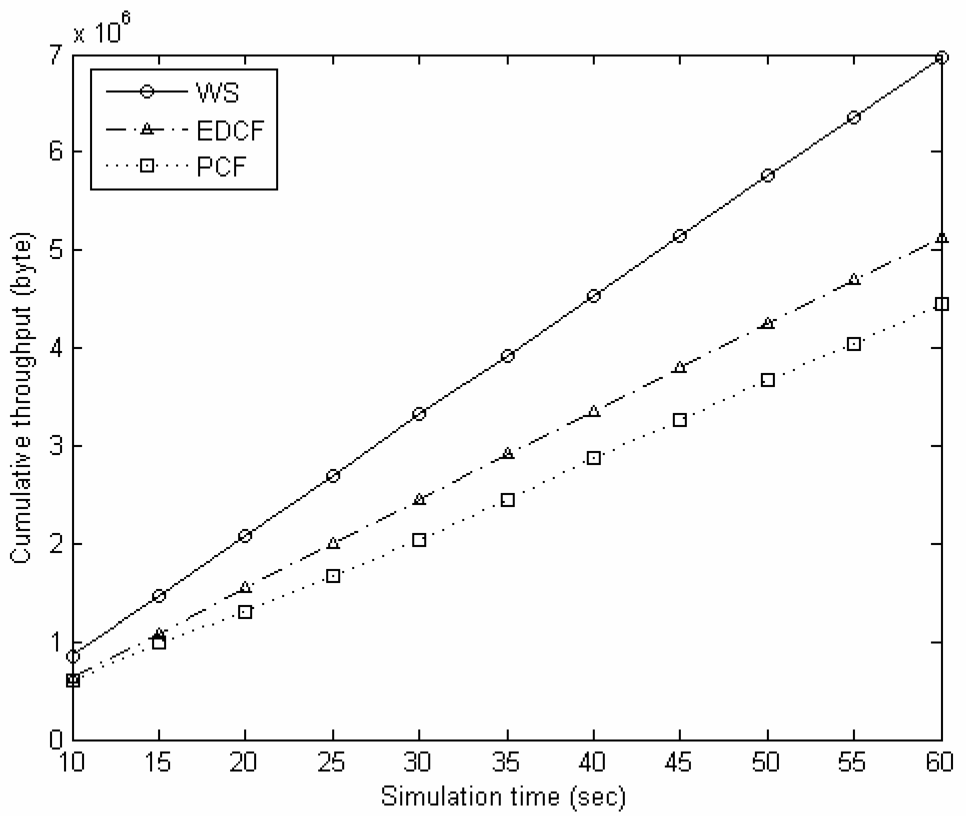

Figure 3.7: The total throughput (cumulative) under different protocols.

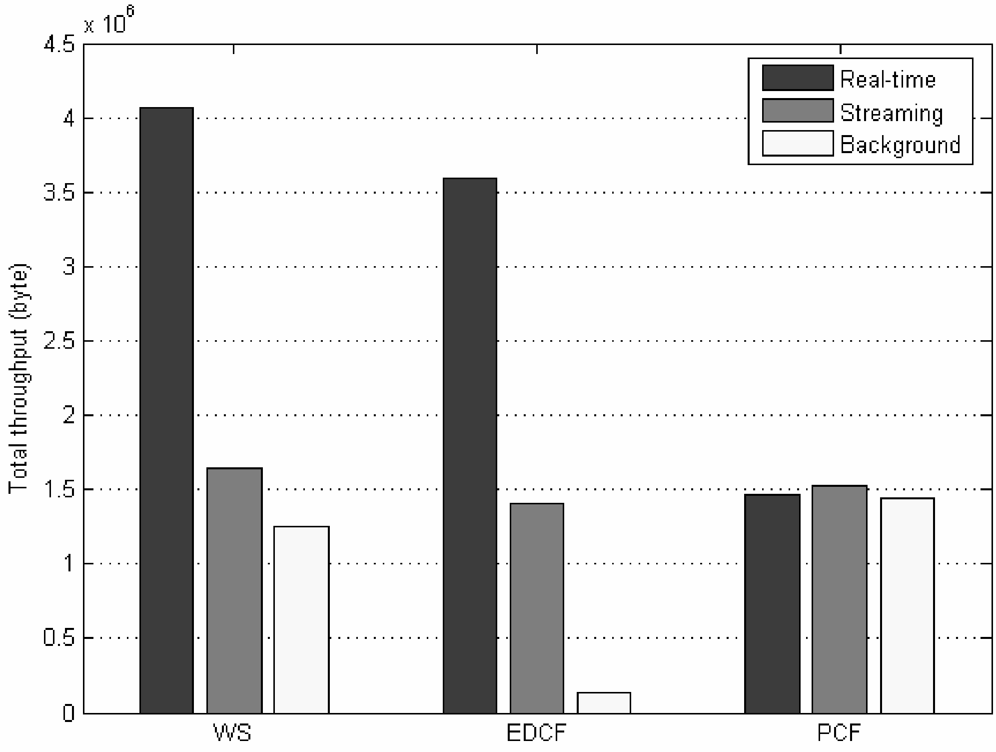

Figure 3.8: The throughput (cumulative) of each service type in different protocols.

3.6.2 Average Access Delay

In this subsection, we feed the network with the original traffic pattern applied in the pervious subsection and a low-loaded traffic pattern to observe the average access delay for each service type when applying different protocols and different patterns. These two patterns have the same services, but the low-loaded traffic pattern has half the offered load of the original traffic pattern (the offered load provided by the original pattern is 1800 Kbps while the offered load provided by the low-loaded pattern is 900 Kbps). Figure 3.9 shows the delay performance when applying the low-loaded pattern while Figure 3.10 shows the delay performance when applying the original pattern. For each traffic pattern, we run the simulation 20 times, each of 60 seconds. The access delay is from the time when the sending node generates a packet to the time when the *MAC* layer of the receiving node receives this packet.

The services in PCF have equal and high access delay when loading either the original pattern or the low-loaded pattern due to the relay transmission and round-robin polling. Concerning the real-time and streaming service, it is hard for us to distinguish the superiority between WS and EDCF when using the low traffic pattern. However, the delay in WS is slightly less than that in EDCF when using the original traffic pattern. It is because the centralized access scheme can adapt to high-loaded networks, but the distributed access scheme suffers from more and more collisions when the traffic load increases. As for the background service, our protocol entirely outperforms the EDCF protocol since our protocol avoids starvation for low priority services. Higher access delay implies higher probability of rejecting a service or dropping a packet if we take the delay requirements of services into account.

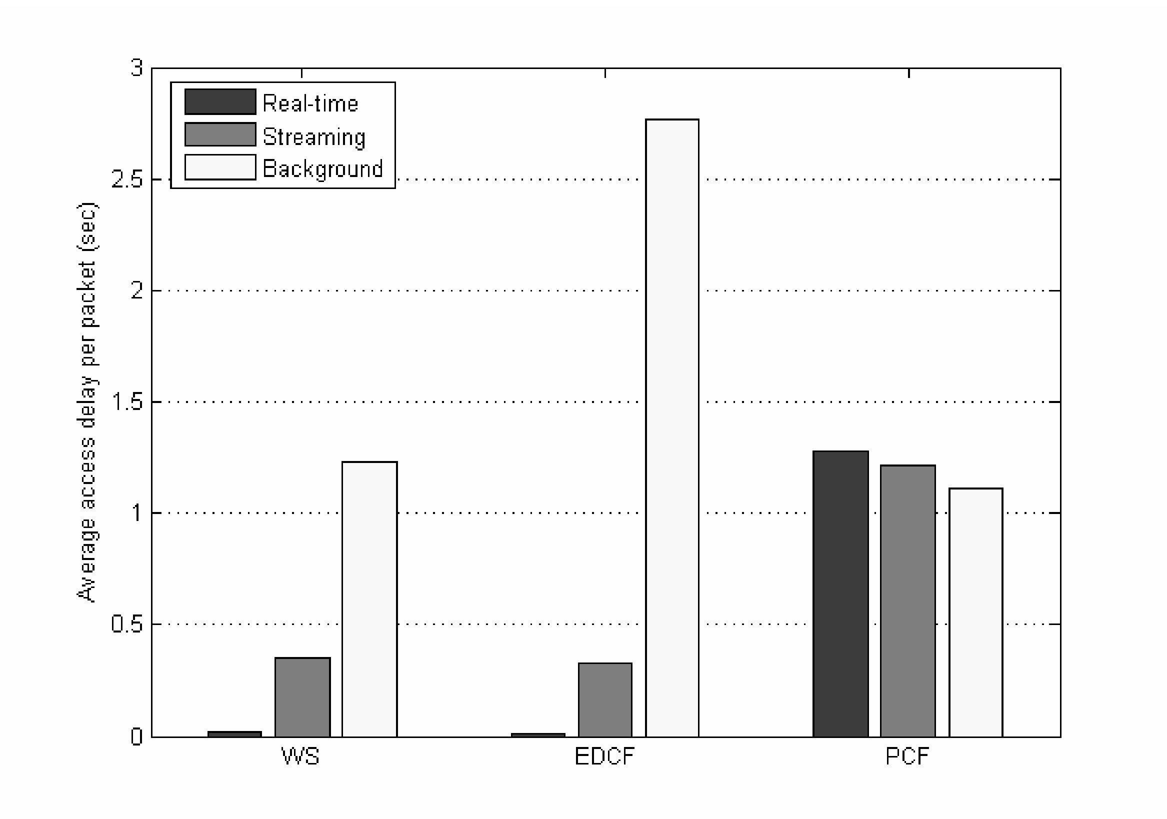

Figure 3.9: The delay performance in the low-loaded network.

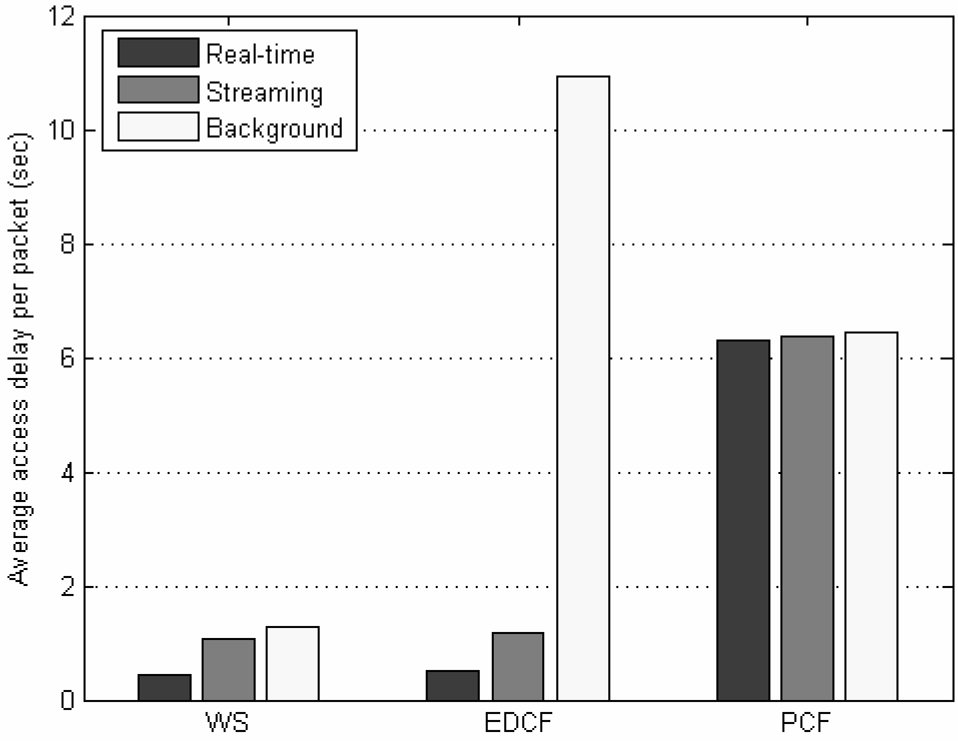

Figure 3.10: The delay performance in the high-loaded network.

3.6.3 Collision Performance

In this subsection, we feed the network with the original traffic pattern and the low-loaded traffic pattern to observe the collision performance when applying different protocols and different patterns. For each traffic pattern, we run the simulation 20 times, each of 60 seconds. Collisions occur when two or more nodes transmit their packets at the same time.

Figure 3.11 illustrates the number of collision packets for each protocol under different traffic patterns. The EDCF protocol has lots of collision packets while PCF and our protocol have much fewer collision packets. From this observation, we know that the distributed access scheme always has more collisions than the centralized access scheme has. In addition, although the contention periods in PCF and in WS have the same length, PCF has more collisions than WS has because PCF has lots of transmission on relaying packets during the contention period.

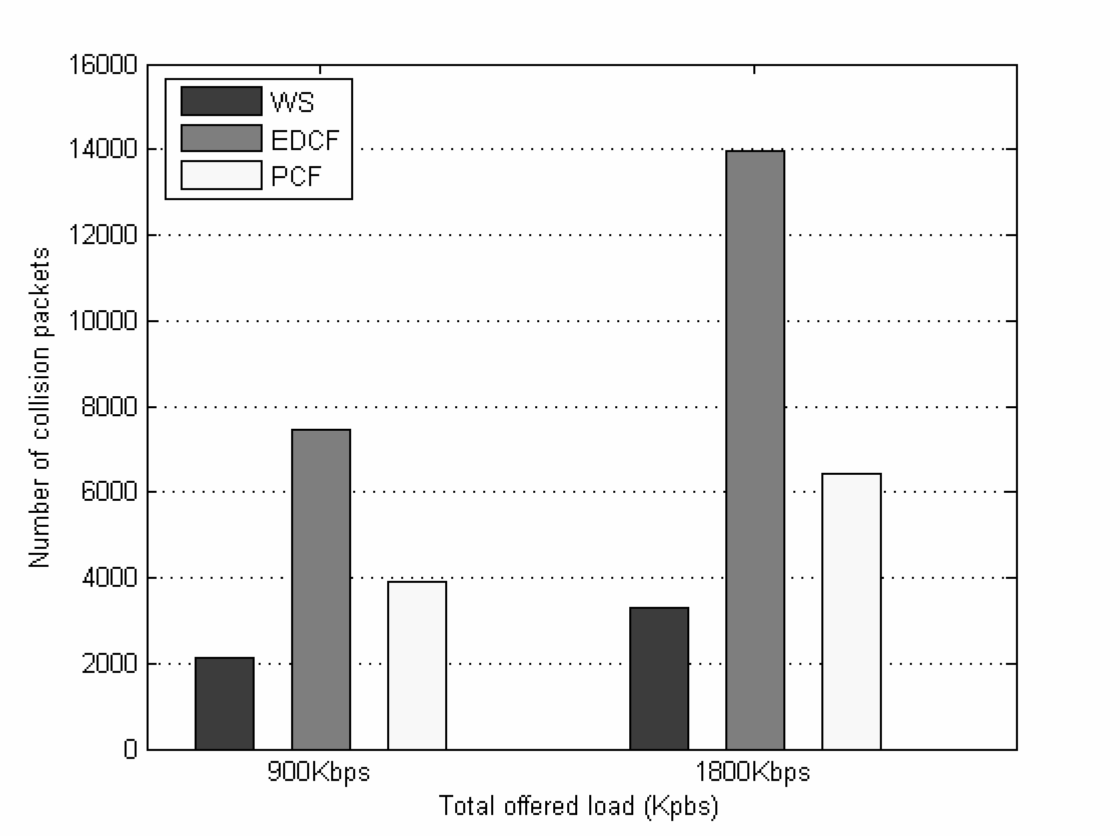

Figure 3.11: The collision packets vs. traffic load.

3.7 Summary

In the chapter, a novel MAC protocol for supporting prioritized medium access in wireless networks was proposed. The weighted scheduling algorithm in the proposed protocol was designed to efficiently utilize network bandwidth and to fairly schedule the transmissions for various types of services. Moreover, a QoS-Enhanced admission control algorithm was proposed to manage resources and guarantee the QoS requirements of services. Four protocol parameters including the length of superframe, the maximum allowable service rate in the transmission period, the maximum allowable requests in the request period, and the system throughput were evaluated by mathematical analysis. IEEE

802.11e EDCF and IEEE 802.11 PCF protocols were compared with our protocol by conducting simulations using ns-2. The simulation results showed that our protocol had the best throughput and delay performance among the simulated protocols. Moreover, it had much fewer collisions than IEEE 802.11e EDFC had and always kept on excellent performance even in high-loaded networks.

Chapter 4

Adaptive Channel Switching Mechanism

Centralized MAC protocols, such as the protocol presented in the previous chapter, are superior to distributed MAC protocols on QoS supports, channel utilization, and congestion control. However, centralized MAC protocols cannot be adopted in ad hoc networks, because they provide only single-hop transmission. This is the significant reason why MAC protocols are not all-pervading in the communication market. Therefore, this chapter proposes a multihop mechanism for centralized MAC protocols to operate on various network topologies. The proposed mechanism provides excellent throughput for both inter-subnet and intra-subnet links, and alleviates the hidden terminal problem. Experimental results reveal that the optimal configuration on the proposed mechanism and the comparison between our mechanism and other multihop forwarding mechanisms. The results demonstrate that the proposed mechanism outperforms other multihop forwarding mechanisms in terms of throughput.

4.1 Overview

Conventional MAC protocols in wireless networks are categorized as distributed MAC protocols or centralized (infrastructure based) MAC protocols. Distributed MAC protocols realize the multihop wireless networks with good mobility, but do not precisely guarantee the QoS of a time-bounded service, since they adopt the contention-based access scheme. Conversely, centralized MAC protocols arrange the transmission in single hop wireless networks to achieve the QoS requirements of the time-bounded service. Centralized MAC protocols have a higher system throughput than distributed MAC protocols, but do not supply the multihop transmission; the origin of this restriction is the hidden terminal problem resulting from the discordance between subnets.

An efficient method for centralized MAC protocols to solve the hidden terminal problem between subnets provides multiple channels or frequencies for transmission. However, most investigations on multi-channel systems focus on either the multi-channel design for distributed ad-hoc networks [NZD99, SV04, NSZ98, MG98, HLG02, CG00, SV03] or optimizing the system utilization by parallel transmission using multiple channels in a subnet [LH00, LYG04, PWS+04]. Only a few investigations on multi-channel systems concern the multi-channel architecture for multihop transmission in centralized wireless networks [MKS02, Pee01a, Pee01b]. K. Mizuno [MKS02] et al. provides a feasible solution to the hidden terminal problem and developed a multihop relaying scheme for centralized MAC protocols. It also realizes an end-to-end QoS guarantee for the time-bounded service. However, in their protocol, initializing a WLAN is complex, and registering a new wireless terminal (WT) is time-consuming (chain topologies are preferred in this protocol). Moreover, lots of channels are required to achieve good throughput for the system, implying that a WT needs many transceivers. Thus, the implementation cost of this protocol, and the power consumption of each WT are both high. The multiple-frequency forwarding mechanism proposed by J. Peetz [Pee01a] eliminates the restriction of the one hop configuration in HiperLAN/2 by employing Multiple Frequency Forwarder Wireless Terminals (MF-WTs). An MF-WT needs to be located in an area with two or more overlapping subnets, and join these overlapping subnets asynchronously by switching the frequency between these subnets. The inter-subnet links are therefore created and the multihop functionality is achieved through these inter-subnet links. However, because the MF-WT must be located in an area with overlapping subnets, many network topologies have no MF-WT available. Additionally, since the throughput of the inter-subnet link depends on the number of MF-WTs and the synchronization between these MF-WTs, the inter-subnet links cannot easily have high throughput.

This chapter proposes a multihop mechanism named *adaptive channel switching* (ACS) for centralized MAC protocols. The ACS mechanism efficiently utilizes the bandwidth by avoiding channel divisions between subnets for centralized protocols. It enables multihop transmission across subnets, and alleviates the hidden terminal problem using a three-channel architecture. Furthermore, ACS can be adapted to various network topologies without complicated initialization procedures or synchronization between subnets.

The rest of this chapter is organized as follows. Section 4.2 reviews pertinent literature that gives centralized MAC protocols the ability of multihop transmission by adopting multi-channel architectures. Section 4.3 then describes the proposed ACS mechanism and its analytical model. Section 4.4 demonstrates the performance results of the ACS mechanism and other multihop mechanisms by conducting simulations. Conclusions are finally drawn in Section 4.5.

4.2 Related Work

This section briefly introduces two multi-channel mechanisms that provide the centralized MAC protocol the ability of multihop transmission.

4.2.1 KMH Mechanism

The KMH mechanism was proposed by K. Mizuno et al. [MKS02], and is named after the authors. The KMH mechanism adopts a PCF based polling scheme in a multihop wireless network with multiple channels, where the PC and WT each utilizes two or more transceivers. The station in the KMH mechanism has three modes for each channel: master mode, slave mode, and silent mode as depicted in Figure 4.1. In master mode, the station acts as a PC in the channel. In slave mode, the station acts as a WT. In silent mode, the station sets NAV, and is not permitted to send packets during the CFP. After NAV resetting, a station in silent mode can send packets until the next beacon frame is sent. In this manner, three modes for each channel enable communication by all wireless links using PCF, and offer QoS guarantees from end-to-end. KMH addresses some issues, such as associating a new station to a network and guaranteeing the QoS, which are not described for considerations of space.

Figure 4.1: The channels and modes in the KMH mechanism.

Analytical results indicate that the KMH mechanism suffers from some inefficient problems. First, initializing a network, as well as associating a new station to an existing network, is complex and time-consuming. The complexity of initializing a network and associating a new station to it is proportional to the number of channels used in the KMH mechanism. Moreover, the KMH mechanism performs well when all stations are distributed in a line, but performs badly when the connection dimension is large, i.e., each station is connected by many stations, since it requires many channels and takes a long time to construct a network.

4.2.2 Multi-Frequency Forwarding Mechanism

The multi-frequency forwarding mechanism was proposed by J. Peetz [Pee01a]. This mechanism enables inter-subnet links, and extends the one-hop connectivity to a multihop ad hoc connectivity for the HiperLAN/2 standards [Joh99]. Each subnet in HiperLAN/2 determines its operation frequency channel based on interference minimization based on the Dynamic Frequency Selection (DFS). Figure 4.2 shows an example for a corresponding multihop network configuration consisting of two interconnected subnets. Both MF-WT1 and MF-WT2 are within the coverage range of *Central Controller* (CC) 1 and 2, where the MF-WT is the WT with the forwarding functionality. Therefore, increasing the number of MF-WT capable terminals increases the number of stable inter-subnet links.

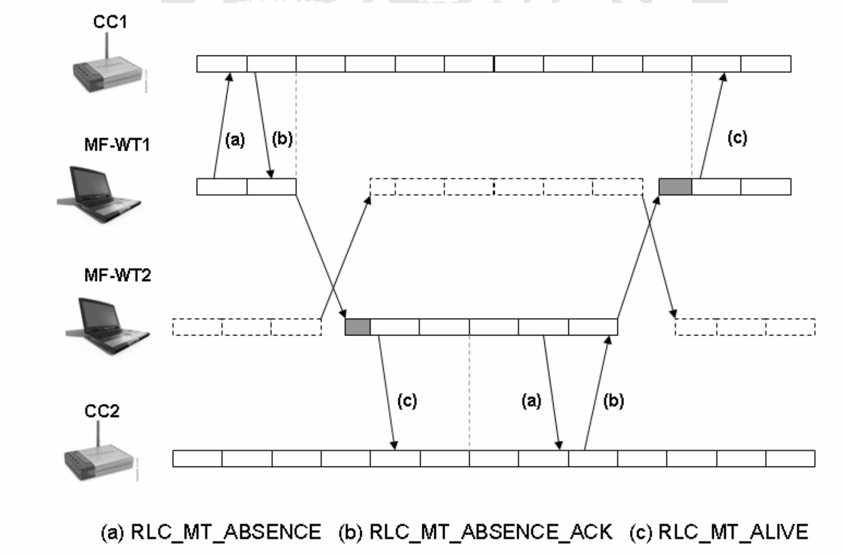

Figure 4.2: MF-WT operation in subnet 1 and 2.

With the MT_Absence function, the H/2 RLC standard enables WT to withdraw from communication. The WT transmits the message *RLC_MT_ABSENCE* to inform the CC that it is unavailable for a time interval of $0 \le mt - absence - time \le 63$ MAC frames. When the CC responds with *RLC_MT_ABSENCE_ACK*, the WT changes to the absent state, and the absence timer is started. The communication between WT and CC is continued immediately as soon as the absence timer expires. MT_Absence is applied for the novel interconnection concept to facilitate WTs to hold connection to more than one CC. The aim of the MT_Alive procedure is to check whether a CC and WT can communicate with each other. The MF_Alive function may be used to indicate the presence of an MF-WT to the CC by sending an *RLC_MT_ALIVE* message after switching and synchronizing to the new frequency channel.

The multi-frequency forwarding mechanism is founded on an intermittent presence of forwarding WTs at each subnet to be interconnected. Therefore, the MF-WT periodically withdraws from a current transmission for a certain number of *mt-absence-time* MAC frames by using the RLC functions MT_Absence and MT_Alive. Figure 2 shows the operation of two MF-WTs successfully associated with the CCs of two subnets. Assume that an MF-WT is alternating between CC1 and CC2. To leave the current CC, for example CC1, it sends the *RLC_MT_ABSENCE* message containing the *mt-absence-time* parameter. When the MF-WT receives the acknowledgement from CC1, the radio connection to CC1 is intermitted, and the absence period counter is started from the following MAC frame. The Broadcast CHannel (BCH) transmitted by CC2 then has to be detected and decoded by the MF-WT for synchronization.

According to our results, the multi-frequency forwarding mechanism cannot work on some networks, e.g., the network in which no overlapped coverage area of CCs exists, or no WT exists in the overlapped coverage area (the MF-WT is not available). Moreover, the throughput of the inter-subnet link depends on the number of MF-WTs and the synchronization between these MF-WTs; the insufficiency of MF-WTs and the uneven synchronization seriously degrades the throughput of inter-subnet links.

4.3 Adaptive Channel Switching

4.3.1 System Description

This section proposes a multihop mechanism called adaptive channel switching for centralized MAC protocols. The ACS mechanism has the following features: 1) it avoids channel divisions between subnets, allowing the system to use the bandwidth efficiently; 2) it enables the multihop transmission across subnets, and alleviates the hidden terminal problem; 3) it eliminates the need for complex initialization and synchronization between subnets; 4) it can be adapted to various network topologies, and 5) it uses the smallest possible number of transceivers to realize these goals.

Since this chapter describes the ACS mechanism based on enhancing the IEEE 802.11 PCF, the following description uses the term PC instead of AP, and uses the term WT to indicate a non-PC station. The ACS mechanism divides the total bandwidth into three channels, namely the *Control Channel* (C-channel), the *Data Channel* (D-channel), and the *Relay Channel* (R-channel). C-channel is adopted for the exchange of control signals such as beacon frames and CF-End; D-channel is adopted for the transmission of data packets, which occupies the most system bandwidth, and R-channel is mainly used by the boundary stations in a subnet to relay packets to adjacent subnets. D-channel can be accessed in contention or contention-free ways, while R-channel can only be accessed with contention. Notably, each station can access only one channel through a particular transceiver in it, so three transceivers are required for each station. A station can operate in either the *Free-Mode* (F-mode) or the *Restricted Mode* (R-Mode). When operating in F-mode, stations send control signals via the C-channel and transmit data packets via the D-channel, they also use the R-channel to communicate with other stations in R-mode. When operating in R-mode, stations are restricted to using only the R-channel to transmit data packets based on the CSMA/CA and RTS/CTS mechanisms. Although the R-mode stations can send packets through only the R-channel, they can hear the data packets sent in other channels. Therefore, the boundary WT of a subnet can enter R-mode to participate in the CFP of its subnet, and relay the *outgoing packets* at the same time, where the outgoing packets are the packets belonging to other subnets. Besides the data packets, the R-mode station sends RTS, CTS, ACK and polling response via R-channel; the station that receives the packets from the R-channel should respond to the sending station through R-channel if needed. The example in Figure 4.3 provides a good understanding of the channels and modes defined in the ACS mechanism.

All stations are initially in F-mode. The PC of a subnet broadcasts the beacon frame or CF-End message through the C-channel to announce the beginning or the end of a CFP.

The WTs that receive the beacon frame sequentially send *jamming packets*, indicating the length of CFP to their neighbors through the C-channel as depicted in Figure 4.3. The sequence of the transmitting jamming packets is included in the beacon frame. The stations that receive the jamming packets then begin the *passive restriction procedure*, and switch to R-mode. An R-mode station entering the passive restriction procedure continues recording all incoming jamming packets (coming from other subnets), and returns to F-mode after the duration of the latest recorded CFP expires. If a PC intends to start a CFP, but is not allowed to send the beacon frame, i.e., it is in R-mode, then it immediately transmits the beacon frame to its WTs after switching to F-mode. If a WT returning from R-mode to F-mode finds that its subnet is current in CFP, then it immediately transmits a jamming packet to its neighbors. The flowchart of the passive restriction procedure is presented in Figure 4.4 (please ignore the dotted portion of the figure at this stage).

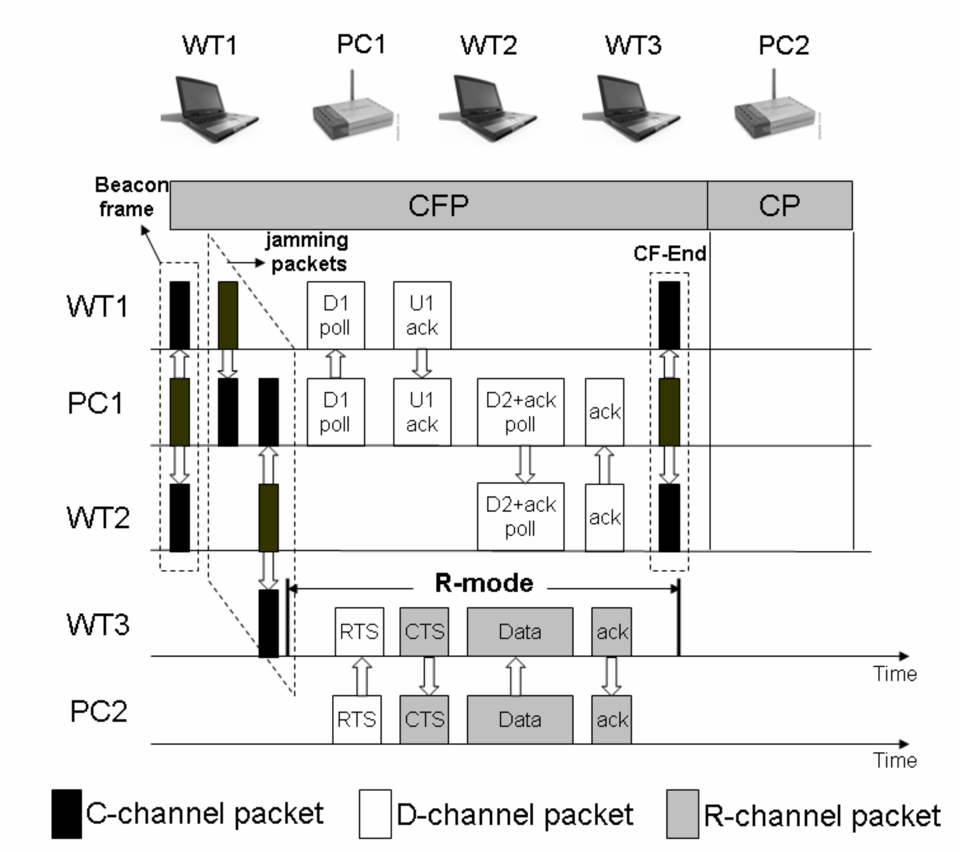

Figure 4.3: An example illustrates the packet exchange in the ACS mechanism.

The other condition (denoted as *self-restriction procedure* in the following) for a WT to operate in R-mode is shown in Figure 4.5. An F-mode WT with more than P_{out} packets that are destined to other subnets in its next-hop switches to R-mode if it receives the beacon frame from its PC, where P_{out} denotes the threshold of the outgoing packets (PC is not allowed to switch to R-mode by the self-restriction procedure). This R-mode WT automatically returns to F-mode if it receives the CF-End frame from its PC. Notably, after the R-mode duration of a WT restricted by the passive restriction procedure expires, there is still a chance for the WT to remain in the R-mode. The chance is when the WT has more than P_{out} outgoing packets to send and finds that the subnet is currently in CFP (please refer to the dotted portion in Figure 4.4). The R-mode duration is then extended to the end of the CFP.

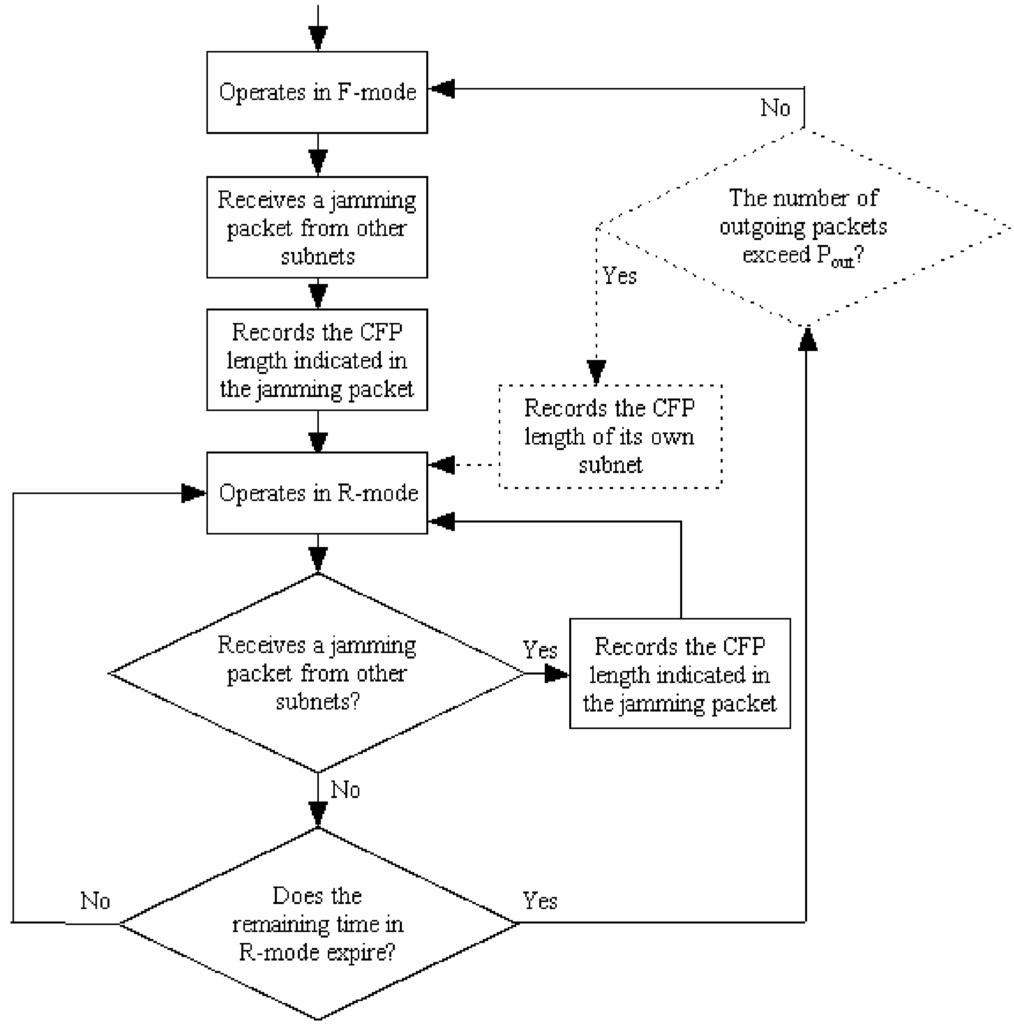

Figure 4.4: The flowchart of the passive restriction procedure including part of the self-restriction procedure.

The self-restriction procedure complementing the passive restriction procedure can increase the throughput of the inter-subnet link. The improvement of throughput due to the self-restriction procedure can be revealed by considering an example in Figure 4.3, where a data flow is sent from PC1 to PC2. This data flow is blocked in halfway when PC1 starts the CFP, and WT2 operates in F-mode. This block occurs frequently, and reduces the throughput of the inter-subnet link if the self-restriction procedure is not introduced. The self-restriction procedure can force WT2 to operate in R-mode (WT3 operates in F-mode since it does not receive the jamming packet from the R-mode WT2), and facilitate the use of R-channel between WT2 and WT3.

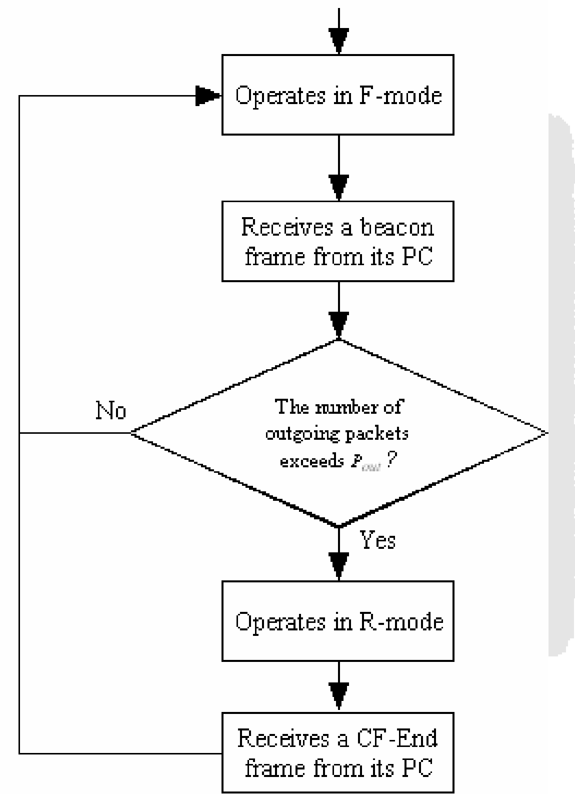

Figure 4.5: The flowchart of the self-restriction procedure in the ACS mechanism.

4.3.2 Analytical Model

This subsection presents an analytical model for evaluating the relay performance of the ACS mechanism under different configurations and topologies. The analytical model applies the ACS to two topologies shown in Figure 4.6: the first topology, denoted as “case 1”, consists of two subnets in which the transmission range of the PCs do not overlap, while the second topology denoted as “case 2” comprises two subnets in which the transmission range of the PCs do overlap. Since the performance of relay transmission is the most important factor in evaluating a multihop mechanism, the proposed model focuses on analyzing the queue length of relay WTs (the gray nodes in Figure 4.6).

(i) Analysis of the ACS mechanism in case 1 without self-restriction procedure.

To evaluate the queue length of the relay WTs (the gray nodes in subnet A) in case 1, the mean arrival and service rates of the relay WTs are derived according to several assumptions mentioned below, by using the *M*/*M*/1 Markovian Birth-Death Queueing Model. The CP length is assumed to be much shorter than the CFP length, allowing all behaviors in CP to be negligible. Additionally, the relay WTs in subnet A (or subnet B) are assumed to spend half of their lifetime in R-mode, and the other half in F-mode. Let n_A and n_B denote the number of WTs in subnets A and B; m_A and m_B denote the number of relay WTs in subnets A and B; B_T denote the system bandwidth (Mbps), and r_D denote the ratio of the bandwidth reserved for D-channel to the bandwidth reserved for R-channel. Additionally, each WT in subnet A is assumed to send data streams with data rate b to PC B (the packet inter-arrival time follows the exponential distribution). The relay WTs in subnet A are regarded as one aggregated node (denoted as R_A in the following), which is analyzed for the mean queue length in different modes.

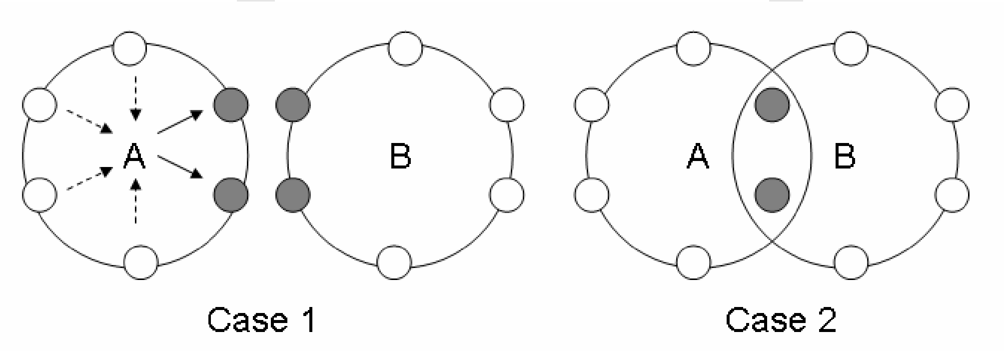

Figure 4.6: The illustration of the topologies presented in the analytical model.

The mean arrival rate of R_A in F-mode equals n_A*b, and should be bounded by $m_A b + \frac{1}{2} B_T r_D U_{CFP}$, where U_{CFP} denotes the channel utilization of CFP, and is formulated as follows.

$$U_{CFP} = \frac{D_p(1-p_{null})}{(D_p + D_H)(1-p_{null}) + D_p p_{null} + PIFS}, \tag{4.1}$$

where D_p denotes the payload size of a packet; D_H denotes the packet header, and p_{null} denotes the probability that a station has no packets to transmit when polled. The mean service rate of R_A equals 0, since R_A only communicates with the stations within subnet A. Therefore, $E[Q_{RA\_F}]$, the mean queue length of R_A measured at each end of F-mode can be calculated as:

$$E[Q_{RA\_F}] = \begin{cases} n_a b \cdot T_{frame} / D_p & if\ (n_a - m_a)b \le \frac{1}{2} B_T r_D U_{CFP} \\ (m_a b + \frac{1}{2} B_T r_D U_{CFP}) \cdot T_{frame} / D_p & otherwise \end{cases}, \tag{4.2}$$

where T_{frame} denotes the duration of a superframe.

The mean arrival rate (λ) when R_A is in R-mode is the same as that when it is in F-mode; the service time follows the exponential distribution with the mean (μ) equal to $1/B_T(1-r_D)U_{CP}(m_A)$, where $U_{CP}(x)$ denotes the channel utilization of CP with x contending stations. $U_{CP}(x)$ is discussed by Bianchi [Bia00], and is formulated as follows.

$$U_{CP}(x) = \frac{p_s(x) p_{tr}(x) D_P}{(1-p_{tr}(x))\sigma + p_{tr}(x) p_s(x) T_s + p_{tr}(x)(1-p_s(x)) T_c}. \tag{4.3}$$

Here, without RTS/CTS, T_s denotes the average time that the channel is sensed busy due to a successful transmission; T_c denotes the average time that the channel is sensed busy by each station during a collision; σ denotes the duration of an empty slot time; $p_{tr}(x)$ denotes the probability that at least one transmission occurs among x stations in the considered time slot, $p_s(x)$ is the conditional probability that exact one station transmits on the channel given that there is at least one transmission among x stations in the considered time slot. These Bianchi's parameters are given in (4.4)–(4.7) (where δ denotes the propagation delay, and $CW_{\min}$ denotes the minimum contention window).

$$\begin{cases} T_s = D_H + D_P + SIFS + \delta + ACK + DIFS + \delta \\ T_c = D_H + D_P + DIFS + \delta \end{cases}. \tag{4.4}$$

$$p_{tr}(x) = 1 - (1-\tau)^x. \tag{4.5}$$

$$p_s(x) = \frac{n\tau(1-\tau)^{x-1}}{p_{tr}(x)} = \frac{n\tau(1-\tau)^{x-1}}{1-(1-\tau)^x}. \tag{4.6}$$

$$\tau = 2/(CW_{\min}+1). \tag{4.7}$$

For simplicity, (4.7) is calculated under the assumption that no exponential backoff is considered. According to the *M*/*M*/1 queueing model, the mean queue length of R_A measured at each end of R-mode, given by $E[Q_{RA\_R}]$, can be obtained as:

$$E[Q_{RA\_R}] = \frac{\lambda^2}{\mu(\mu-\lambda)}. \tag{4.8}$$

Based on the assumption that R_A stays equally in F-mode and in R-mode, $E[Q_{RA}]$, the mean queue length of R_A measured at each end of superframe, can be calculated as:

$$E[Q_{RA}] = (E[Q_{RA\_F}] + E[Q_{RA\_R}])/2. \tag{4.9}$$

(ii) Analysis of the ACS mechanism in case 1 with self-restriction procedure.

Because of the two assumptions: "R_A stays equally in F-mode and R-mode", "CP can be neglected compared to the duration of CFP", we can say that R_A is sure to enter F-mode after its R-mode duration expires, and vice versa. However, when the self-restriction procedure is applied, R_A probably remains in R-mode after its current R-mode duration expires. That is, the time that R_A stays in R-mode is longer than the time it stays in F-mode. The following equation expresses the mean queue length of R_A.

$$E[Q_{RA}] = x_1 E[Q_{RA\_F}] + x_2 E[Q_{RA\_R}], \tag{4.10}$$

where x_1 and x_2 denote the limiting state probabilities of state A_1 (R_A stays in F-mode) and A_2 (R_A stays in R-mode) in the two-state Markov chain shown in Figure 4.7. The values of x_1 and x_2 can be calculated by solving the following equation,

$$\begin{cases} x_1 + x_2 = 1 \\ x_1 p_{12} = x_2 p_{21} \end{cases}, \tag{4.11}$$

where p_{11}, p_{12}, p_{21} and p_{22} denote the transition probabilities of $A_1 \rightarrow A_1$, $A_1 \rightarrow A_2$, $A_2 \rightarrow A_1$, and $A_2 \rightarrow A_2$, respectively. Since R_A is sure to enter F-mode after its R-mode duration expires, $p_{12} = 1$ and $p_{11} = 0$ ($p_{11} + p_{12} = 1$). p_{22} denotes the probability that R_A remains in R-mode after its current R-mode duration expires. In other words, p_{22} denotes the probability that the number of packets in R_A's queue exceeds P_{out}, which denotes the threshold of the outgoing packets for the aggregated relay node. If $\rho = \lambda / \mu$, then the probability that there are n packets in R_A's queue (denoted as p_n) has the following expression,

$$p_n = (1-\rho)\rho^n. \tag{4.12}$$

Therefore, p_{22} can be calculated as follows.

$$p_{22} = \sum_{n=P_{out}}^{\infty} p_n = (1-\rho)\rho^{P_{out}} + (1-\rho)\rho^{P_{out}+1} + ... = \frac{(1-\rho)\rho^{P_{out}}}{(1-\rho)} = \rho^{P_{out}} \tag{4.13}$$

Consequently, $p_{21} = 1 - p_{22} = 1 - \rho^{P_{out}}$. By substituting $p_{12} = 1$ and $p_{21} = 1 - \rho^{P_{out}}$ into (4.11), the limiting state probabilities can be obtained:

$$x_1 = \frac{1-\rho^{P_{out}}}{2-\rho^{P_{out}}}, x_2 = \frac{1}{2-\rho^{P_{out}}}. \tag{4.14}$$

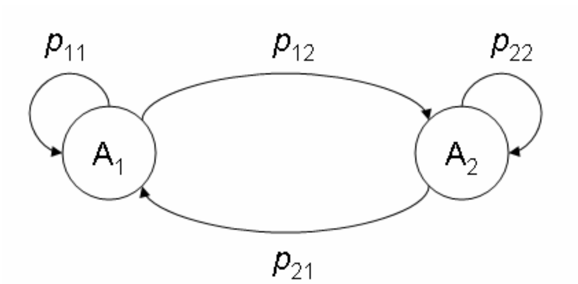

Figure 4.7: The two-state Markov chain illustrating the state of R_A with the self-restriction procedure.

(iii) Analysis of the ACS mechanism in case 2.

In this case, the common relay nodes of subnet A and subnet B depicted in the right hand side of Figure 4.6 are analyzed for their queue length. During the analysis, these common relay nodes are regarded as an aggregated node denoted as R_{AB}. Subnets A and B operate in CFP in turn (nodes A and B enter R-mode in turn) according to the ACS mechanism. To evaluate the queue length of R_{AB}, its mean arrival rate and mean service rate must be derived based on which subnet is currently in CFP. Each node is assumed to be able to hear from other nodes within the same subnet.

When subnet A operates in CFP, all nodes in subnet B except the common relay nodes are sure to be in R-mode. Thus the mean arrival rate of R_{AB} is equal to $n_A \times b$, and should be bounded by $m_{AB} b + \frac{1}{2} B_T r_D U_{CFP}$, where m_{ab} denotes the number of the common relay nodes; the mean service rate of R_{AB} equals 0. Consequently, $E[Q_{RAB\_F}]$, which denotes the mean queue length of R_{AB} measured at each end of F-mode, can be calculated as:

$$E\left[Q_{RAB\_F}\right] = \begin{cases} n_a b \cdot T_{frame} / D_p & if\ (n_a - m_{ab}) b \le \frac{1}{2} B_T r_D U_{CFP} \\ (m_{ab} b + \frac{1}{2} B_T r_D U_{CFP}) \cdot T_{frame} / D_p & otherwise \end{cases} . \tag{4.15}$$

All nodes in subnet A except the common relay nodes are sure to be in R-mode when subnet B operates in CFP. Thus, the mean arrival rate of R_{AB} (denoted as λ') equals $n_A \times b$, and should be bounded by $m_{ab} b + B_T (1 - r_D) \cdot U_{CP} (n_a - m_{ab})$; the mean service rate of R_{AB} (denoted as μ') equals $B_T r_D U_{CFP} / 2$. According to the $M/M/1$ queueing model, $E[Q_{RAB\_R}]$, which denotes the mean queue length of R_{AB} measured at each end of R-mode, can be calculated as:

$$E[Q_{RAB\_R}] = \frac{\lambda'^2}{\mu'(\mu' - \lambda')}. \tag{4.16}$$

The mean queue length of R_{AB} measured at each end of superframe (denoted as $E[Q_{RAB}]$) can then be obtained as:

$$E[Q_{RAB}] = (E[Q_{RAB\_F}] + E[Q_{RAB\_R}]) / 2. \tag{4.17}$$

4.3.3 Model Verification and Evaluation

To verify the analytical model described in the previous subsections, the model was examined by comparing them with using a simulation program in C++. Additionally, the relay performance of the ACS mechanism was evaluated using different topologies, and both with and without the self-restriction procedure. Table 4.1 lists all parameters and configurations for the analytical model and simulations. Figure 4.8 shows the analytical and simulation results. Both the analytical and simulation results indicate that "case 1 with self-restriction" outperforms "case 1 without self-restriction" while "case 2" outperforms "case 1 with self-restriction" in terms of the congestion degree of relay nodes. The difference between the analysis and simulation is primarily affected by the analytical assumption "R_A stays equally in F-mode and R-mode", and is slightly affected by the assumption "CP can be neglected compared to the duration of CFP".

Table 4.1: The parameters and configurations applied in the model verification and evaluation.

Parameter	Value	Parameter	Value
n_a	6	D_P	8000 bits
n_b	6	D_H	400 bits
m_a	2	p_{null}	0
m_b	2	SIFS	28 μs
m_{ab}	2	PIFS	64 μs
r_D	0.7	DIFS	128 μs
B_T	11.0 Mbps	σ	50 μs
T_{frame}	20 ms	δ	1 μs
ACK	240 bits	CW_{min}	16
P_{out}	4 packets	Simulation time	60 s

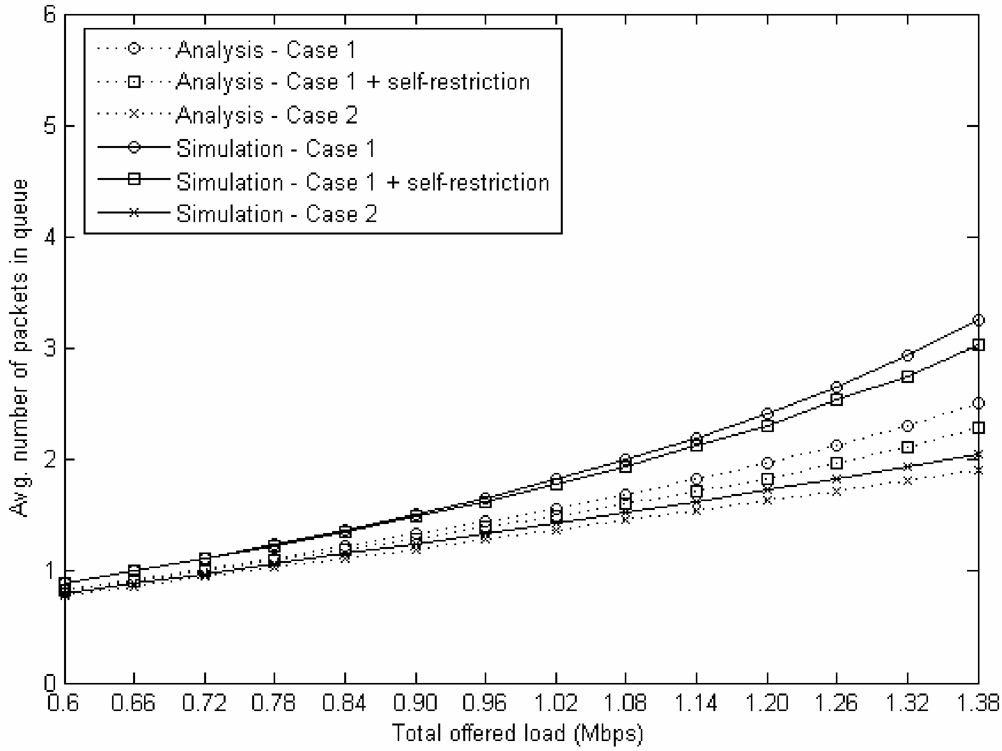

Figure 4.8: The number of packets in relay nodes' queue vs. total offered load.

4.4 Experimentations

The ACS mechanism was compared with the multi-frequency forwarding mechanism, the KMH mechanism, and IEEE 802.11 DCF, based on enhancing IEEE 802.11 PCF. Since the ACS mechanism can adapt to all network topologies, while other mechanisms can only adapt to some specific topologies, three simulation scenarios with corresponding topologies that are suitable to these compared mechanisms were designed. The scenario described in Section 4.4.1 was suitable for the DCF mechanism; the scenario described in Section 4.4.2 was suitable for the multi-frequency forwarding mechanism, and the scenario described in Section 4.4.3 was suitable for the KMH mechanism. In each scenario, the corresponding mechanism was compared with the ACS mechanism in terms of throughput. These scenarios were programmed in C++ and configured by the following settings. The total bandwidth of the system was 11Mbps, and the simulation time was 60seconds. The transmission and interference ranges of a station were 40m and 45m, respectively. Each station had at most 50 packets in its queue, and routed packets according to the DSR [JMB01] routing protocol. All traffic was generated from CBR sources generating fixed size packets (1000 bytes). The details of these scenarios are described separately in the following subsections.

4.4.1 The Transmission Range of Each PC is not Overlapped

Figure 4.9 shows the network topology in this scenario. PC1 coordinated the left subnet, and PC2 coordinated the right subnet; the transmission range of each PC did not overlap. Dataflow 1 transmitted from WT1 to WT8 was simulated as the inter-subnet traffic. Dataflow 2 was sent from WT2 to WT3, and dataflow 3 was transmitted from WT6 to WT7; both were simulated as the intra-subnet traffic. The ACS mechanism with different configurations was first applied on this topology to find the optimal configuration. The length of a superframe was fixed at 40ms, and the load of each data flow was set at 1500 Kbps. The maximal queue size, the value of P_{out}, and the ratio of the bandwidth allocated to D-channel to the bandwidth allocated to R-channel were varied.

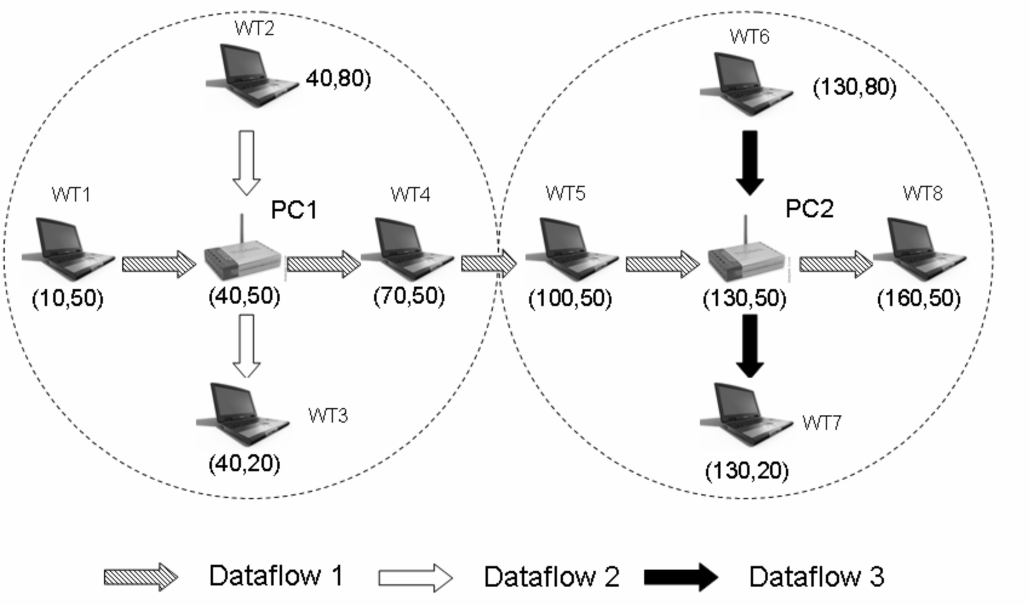

Figure 4.9: The topology in the scenario 1.

The maximal queue size in a station was varied from 10 to 100 packets to observe the cumulative throughput for each dataflow. In the process, P_{out} was fixed to be 20 packets, and the ratio of the bandwidth allocated to D-channel to the bandwidth allocated to R-channel was fixed at 8/3. Simulation results indicated that the cumulative throughput on each dataflow was not affected by the maximum queue size. That is, no benefit can be obtained by increasing the maximal queue size. This result was not illustrated to save space. Figure 4.10 illustrates the throughput performance for each dataflow with P_{out} varying from 10 to 100 packets. In the figure, the maximum queue size was set to be 50 packets, and the ratio of the bandwidth allocated to the D-channel to the bandwidth allocated to the R-channel was fixed at 8/3. Figure 4.10 indicates that the inter-subnet traffic (dataflow 1) achieved the highest throughput when P_{out} was equal to 14 or 16 packets, while the intra-subnet traffic (dataflow 2 and dataflow 3) was not affected by the variation of P_{out}. Additionally, Figure 4.11 shows the throughput performance obtained by varying the ratio of the bandwidth allocated to D-channel to the bandwidth allocated to R-channel. In the figure, the maximum queue size was set to be 50 packets, and P_{out} was fixed at 20 packets. The inter-subnet traffic was found to achieve the highest throughput when the ratio was 7/4; the throughput of the intra-subnet traffic began to drop when the bandwidth allocated for D-channel was less than 7 Mbps.

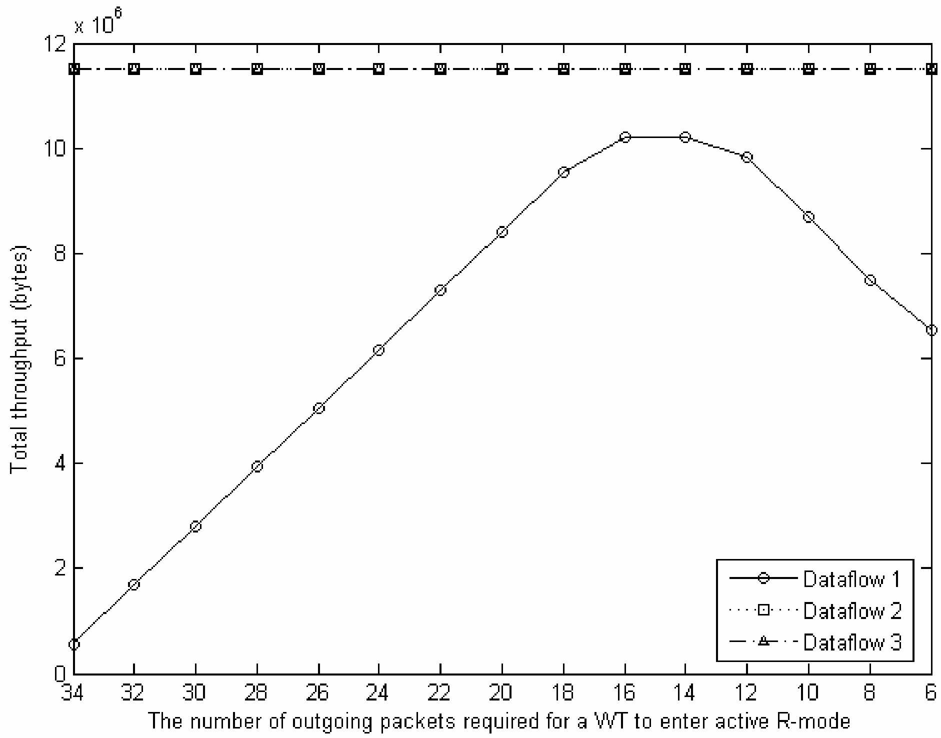

Figure 4.10: The cumulative throughput in the ACS mechanism when varying the P_{out}.

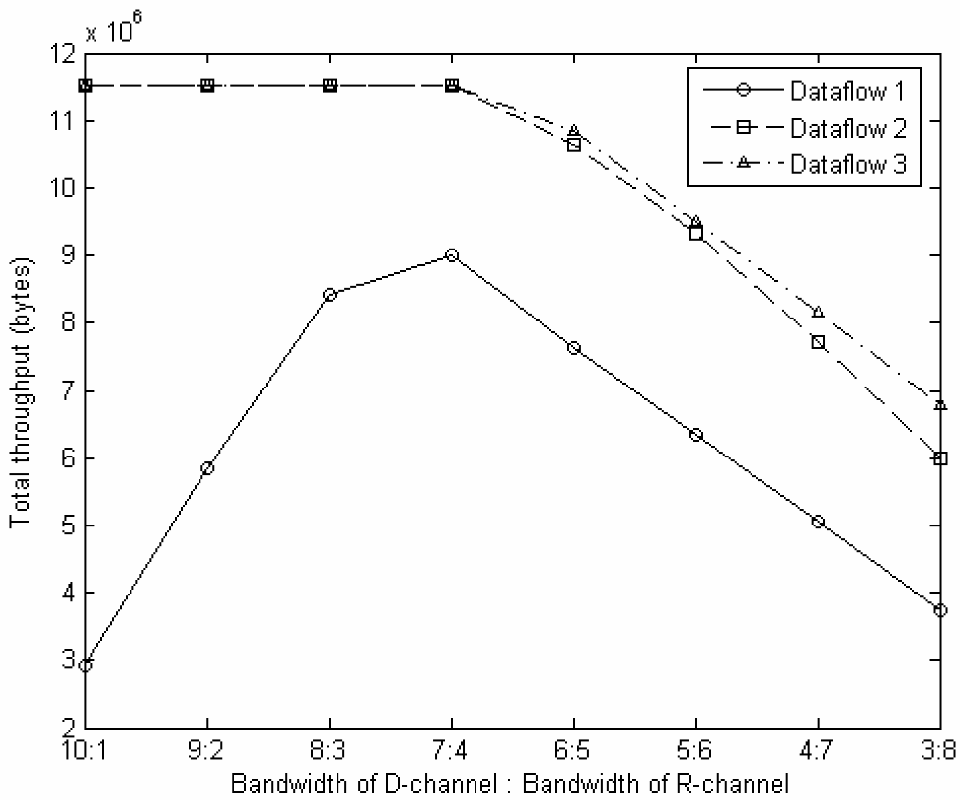

Figure 4.11: The cumulative throughput in the ACS mechanism when varying the bandwidth allocation.

The remaining experiments were performed with the ACS mechanism by the following configuration: maximum queue size of 50 packets, P_{out} = 16 packets, and 7/4 as the ratio of the bandwidth allocated to D-channel to the bandwidth allocated to R-channel.

In the topology depicted in Figure 4.9, the ACS mechanism was compared with IEEE 802.11 DCF by varying the load of each dataflow from 300Kbps to 1500Kbps. Figure 4.12 illustrates the variation of the cumulative throughput on each dataflow. These results indicated that the ACS mechanism strongly outperformed the DCF mechanism in terms of inter-subnet traffic. For instance, in the case of 1500 Kbps load, the ACS had the throughput of $9.0 * 10^6$ bytes while the DCF only got to $9.1 * 10^3$ bytes. This is because the routing path of dataflow 1 is much longer than these of dataflow 2 and dataflow 3. In DCF, if a flow has a long routing path, then the packets belonging to this flow frequently contend with each other for channel access on successive links, reducing end-to-end throughput. As for the intra-subnet traffic, the traffic in the ACS mechanism still had a higher throughput than the traffic in the DCF. It is because the polling-based access scheme always has fewer collisions and retransmissions than the contention-based access scheme in highly congested networks.

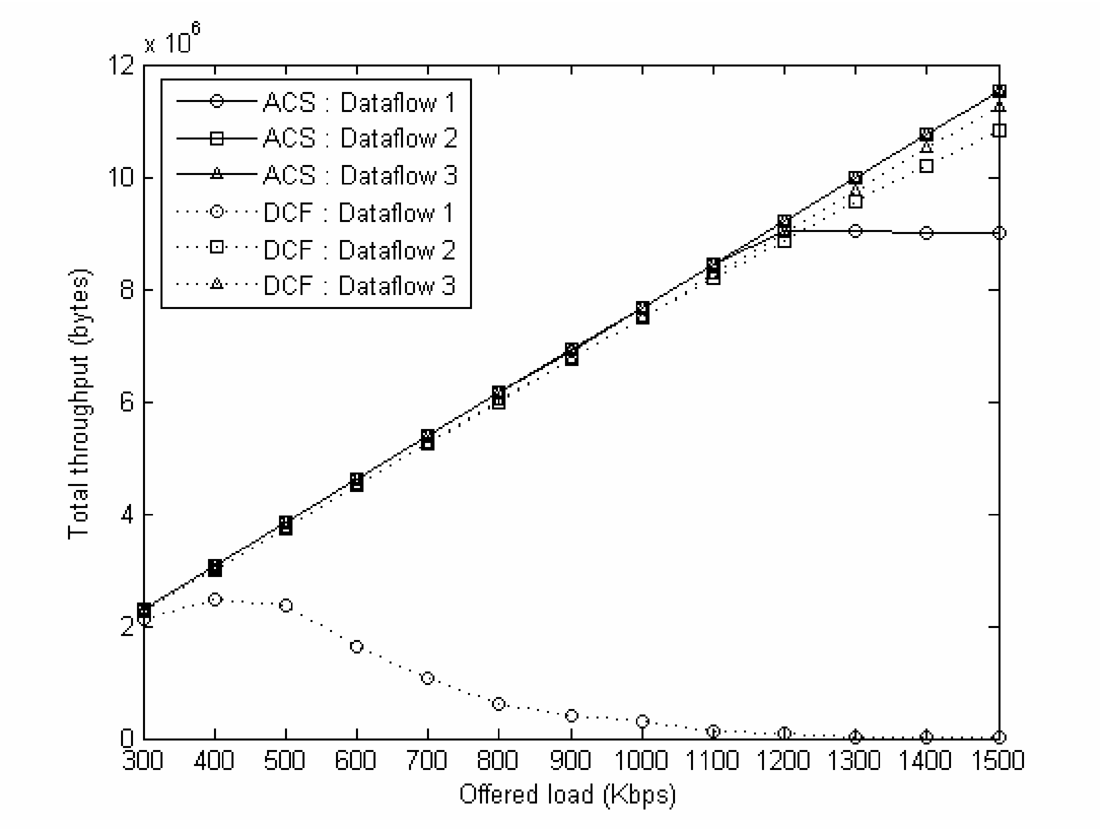

Figure 4.12: The comparison between the ACS and DCF about the cumulative throughput.

4.4.2 The transmission range of each PC is overlapped

This topology was almost the same as the topology shown in Figure 4.9, but the transmission range of each PC was overlapped (the distance between PC1 and PC2 was 60m). WT4 and WT5 in Figure 9 were replaced by an MF-WT in the overlapped area of PC1 and PC2. The ACS mechanism was compared with the multi-frequency forwarding mechanism in this scenario by assigning each dataflow an load equal to 1500Kbps. As for the configurations on the multi-frequency forwarding mechanism, the frequency band of the system was divided into two equal portions, with each subnet occupying one portion of the frequency band (the bandwidth in each subnet was 5.5 Mbps). The MF-WT switched the frequency once every 80ms. A special channel shared by these two subnets was created, and used by the MF-WT to send the request to PCs for asking switch and to receive the acknowledgement from PCs. The configuration on the ACS mechanism was the same as that in previous scenario, since it approached the optimal configuration in this scenario. Figure 4.13 shows the cumulative throughput on each dataflow. For instance, in the case of dataflow 1, ACS had the throughput of $7 * 10^6$ bytes while the multi-frequency forwarding only got to $3.36 * 10^6$ bytes. A gain of $3.64 * 10^6$ bytes was achieved in this case. Overall experimental results indicated that the ACS mechanism outperformed the multi-frequency forwarding mechanism when considering either the inter-subnet traffic or the intra-subnet traffic.

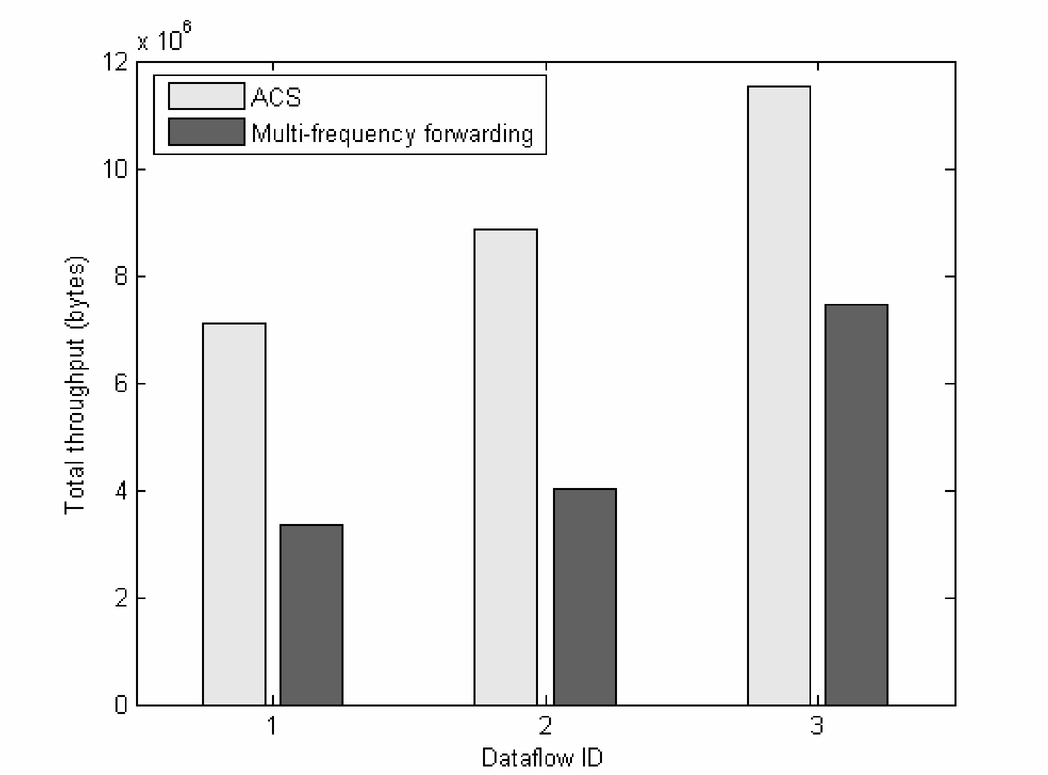

Figure 4.13: The comparison between the ACS and multi-frequency forwarding mechanism about the cumulative throughput.

4.4.3 The WTs and PCs are distributed in a line

Figure 4.14 shows the network topology in this scenario. The topology comprised three PCs and one dataflow going from WT1 to WT4 in this topology. The ACS mechanism was compared with the KMH mechanism in terms of throughput. When applying the KMH mechanism, the complex initialization procedures were ignored, the modes for each station were set as shown in Figure 4.1. Moreover, the experiment assumed that only the station operating in master mode could send data packets to its neighbors in the KMH mechanism. The configuration on the ACS mechanism was the same as that in the first scenario, since it approached the optimal configuration in this scenario. Figure 4.15 illustrates the comparison results which indicated that the ACS had $1.8 * 10^4$ dropped packets and $1.42 * 10^7$ throughput (bytes) while the KMH had $2.2 * 10^4$ dropped packets and $1.03 * 10^7$ throughput (bytes). Accordingly, the dataflow has fewer dropped packets and higher throughput when applying the ACS mechanism than when applying the KMH mechanism.

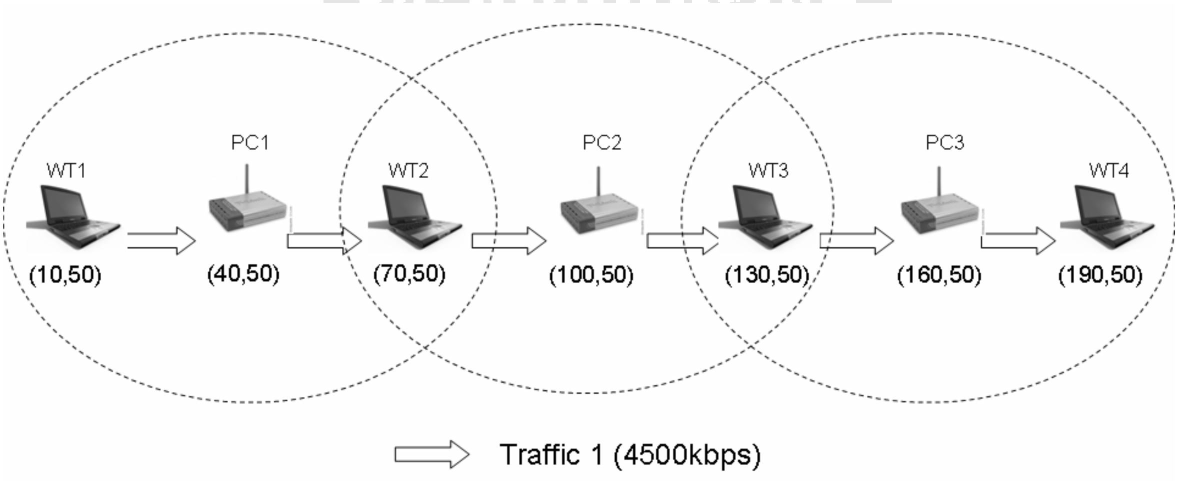

Figure 4.14: The topology in the scenario 3.

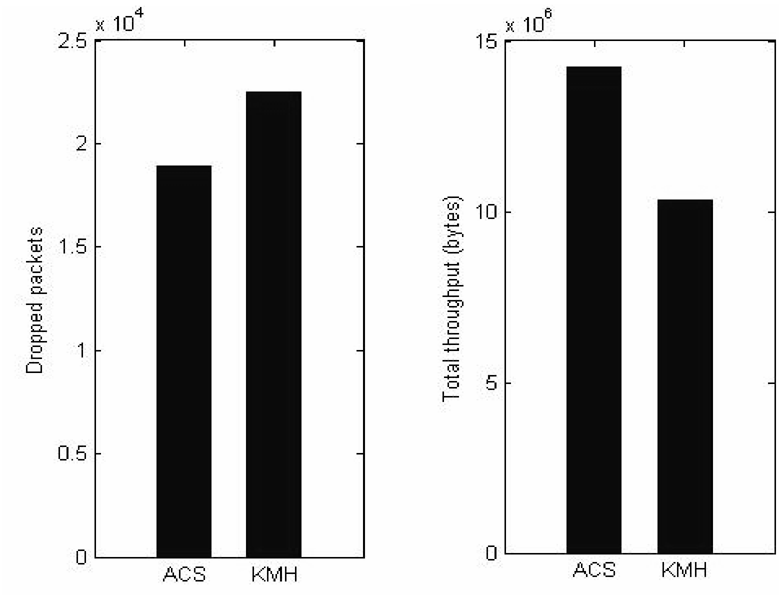

Figure 4.15: The comparison between the ACS and KMH mechanism about the dropped packets and cumulative throughput.

4.5 Summary

This chapter proposed the ACS mechanism for centralized MAC protocols to eliminate the restriction on single-hop transmission. The ACS mechanism that allowed the centralized MAC protocol to transmit data flows across subnets and alleviated the hidden terminal problem can be adapted to various network topologies. The ACS mechanism also eliminated the need for both complex initialization procedures and synchronization between subnets. Experimental results indicated the optimal configuration on the ACS mechanism, and the comparison between the ACS mechanism and other multihop forwarding mechanisms. Table 4.2 summarizes the experimental results and our conclusions.

Table 4.2: The comparison between the ACS mechanism and other multihop forwarding mechanisms (degree: 1 > 2 > 3).

	ACS	KMH	Multi-frequency forwarding
Adaptability to network topologies	1	2	3
Throughput for inter-subnet links	1	2	3
Throughput for intra-subnet links	1	2	3
The number of transceivers required	2	1	3
Initialization complexity	3	1	2

Chapter 5

Delay-Sensitive Fair Queueing Algorithm for WLANs

Fair queueing is useful for providing QoS in an integrated services network, since it provides fairness and bounded delay access. Many wireless fair queueing algorithms have been proposed to tackle the problem of bursty and location-dependent errors, but these algorithms are mainly designed for the large-scale network such as GPRS or 3G. Since these algorithms are not sensitive to the growth of queue length, they cannot adapt to highly variant WLANs directly. To ensure that the services in relay stations meet their QoS requirements, the Weighted-Sacrificing Fair Queueing (WSFQ) model is proposed in this chapter for delay-sensitive multihop WLANs. WSFQ slows down the growth of queue length for real-time traffic, still maintains the property of fairness, and guarantees the throughputs of the station. Moreover, WSFQ can easily adapt itself to various traffic loads. Since the WSFQ is an abstract model, we design a packet-based scheduling algorithm, the Packetized Weighted Sacrificing Fair Queueing (PWSFQ), to approach the WSFQ in practice. To evaluate the performance of our models, WSFQ and PWSFQ are evaluated by mathematical analysis and simulations in the end of this chapter.

5.1 Overview

Many Packet Fair Queueing algorithms (PFQ) have been developed for providing fairness and bounded delay access in wireline networks. PFQ algorithms are first proposed in the context of wired networks to approximate the idealized Generalized Processor Sharing (GPS) policy [DKS89, PG93]. GPS has been proven to have two important properties:

(i) It provides end-to-end delay-bounded services to leaky-bucket constrained sessions; and

(ii) It ensures fair allocation of bandwidth among all backlogged sessions regardless of whether or not their traffic is constrained.

While GPS is a fluid model that cannot be implemented, various packet approximation algorithms are designed to provide services that are nearly identical to GPS.

However, it is inappropriate to directly apply GPS and the corresponding algorithms to wireless networks because of bursty channel errors and location-dependent channel capacity and errors. Bursty channel errors make the host unable to receive continuous services. Location-dependent channel capacity and errors make other error-free sessions receive more services. Lu et al. [LBS99] and Ng et al. [NSZ98] noticed the unfairness problem and presented the Idealized Wireless Fair-Queueing (IWFQ) and the Channel-condition Independent Packet Fair Queueing (CIF-Q) solutions respectively. IWFS and CIF-Q tried to solve the problem by compensating the error-prone flows by using the service share of error-free flows. Thus, if the network becomes heavily loaded when a flow is being compensated, the services of error-free flows may be deteriorated. In order to prevent the service degradation of error-free flows during compensation, Jeong et al. [JMA01] presented the Packetized Wireless General Processor Sharing (PWGPS) algorithm which used pre-allocated service shares for compensation.

In most wireless PFQ algorithms, the amount of compensation services from a leading session is proportional to its weight (i.e. the service share). If the traffic is not heavy and is steady, most of the wireless PFQ algorithms work correctly and efficiently. However, in a highly variant WLAN, the traffic is heavy and varied, the queue length of each session that has limited buffer size grows rapidly and hence the packet loss occurs frequently, especially for a high weighted real-time session. Thus the adaptability to the traffic load becomes a new requirement of wireless PFQ algorithms. In this chapter we develop a Weighted-Sacrificing Fair Queueing (WSFQ) algorithm. It dynamically adjusts the sacrificing rates of leading sessions according to their weights and the traffic load. In WSFQ, the ability of adaptation to network traffic is observed. It achieves better performance in packet loss rate and queueing delay.

The rest of the chapter is organized as follows. In Section 5.2, we introduce WSFQ, examine its fairness, and compare WSFQ with other PFQ policies in terms of the queue length of high weighted leading sessions. Section 5.3 implements PWSFQ that approaches WSFQ. Section 5.4 discusses the simulation results of PWSFQ and CIF-Q. Section 5.5 concludes this chapter.

5.2 Weighted Sacrificing Fair Queueing

5.2.1 Model Definitions

In order to tackle the high variance in traffic, a fair scheduling model with Weighted Sacrificing (WS) is proposed for wireless network in this chapter. WSFQ performs well in terms of packet loss, and still retains the following properties that are addressed by the well-known CIF-Q:

(i) Delay bound and throughput guarantees,
(ii) Long-term fairness,
(iii) Short-term fairness, and
(iv) Graceful degradation in quality of services.

In order to measure fairness, Weighted Sacrificing Fair Queueing (WSFQ) associates to each system S a reference error-free system S_r. WSFQ checks the states of all sessions by S_r at the *checking points* for every T_c time units, where T_c is the *checking period* of the WSFQ and assigned a constant value. The period between the checking point i-1 and i is called the *steady period i* (Figure 5.1). At the *checking points*, a session is *leading* if it has received more services in S than it would have received in S_r, *lagging* if it has received less, and *synchronizing* if it has received the same amount of services. The reference system of WSFQ satisfies the Start-time Fair Queueing (SFQ) model [GVC96] and is described by referring to the work of Jeong et al. [JMA01]. The details are shown as follows.

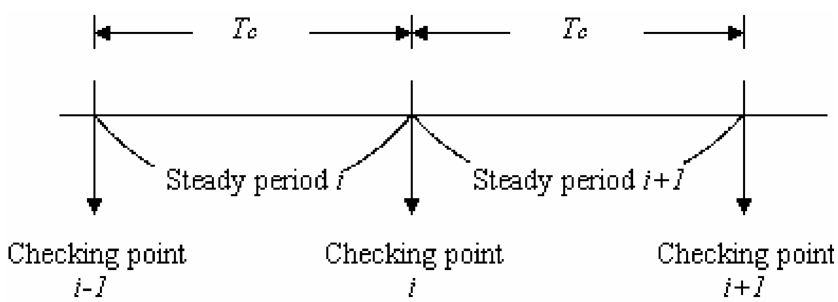

Figure 5.1: Illustration of the *checking points* and the *steady period*.

Let $F_i(t_0, t)$ be the amount of services for session *i* in S_r in a time interval $(t_0, t]$. $F_i(t_0, t)$ is given by

$$\frac{\partial F_i(t_0, t)}{\partial t} = r_i \cdot \frac{\partial V}{\partial t}, \tag{5.1}$$

in which r_i is the given weight (as well as the service share in GPS) of session *i*, and *V* is the virtual time of S_r , which is defined as follows.

$$\frac{\partial V}{\partial t} = \begin{cases} \dfrac{A}{\sum_j r_j} & if \sum_j r_j \neq 0 \\ 0 & otherwise \end{cases}, \tag{5.2}$$

in which *A* is the total service rate of the system.

In WSFQ, the lagging services caused by the channel errors are compensated by the weights of all sessions. But it is important that not only the fairness properties need to be satisfied, but also the problems of crowded queue caused by high traffic load to each session need to be overcome, especially for high weighted sessions. Thus, a leading session should be postponed by the normalized amount of services according to its weight, compensation services should be distributed to lagging sessions in proportion to the weights of lagging sessions, and the services from suspended sessions should be distributed to all the available sessions in proportion to their weights.

For the reasons described above, WSFQ uses time-varying weight $v(t)$ instead of constant weight *r* to make services allocation for each session. The time-varying weight $v_i(t)$ of session *i* is defined as follows.

$$v_i(t) = \begin{cases} r_i & if \quad i \in C(t) \\ r_i \cdot (1 - \Phi_i') & if \quad i \in N(t), \\ 0 & Otherwise \end{cases} \tag{5.3}$$

in which $C(t)$ is the set of sessions that are backlogged and in the lagging state, $N(t)$ is the set of sessions that are backlogged and not in the lagging state (i.e. leading or synchronizing). Let n $= |N(t)|$. Define Φ_i' to be the sacrificing ratio of session *i*. The value of Φ'_i is set according to session i's weight, i.e.

$$\Phi'_i = \begin{cases} \Phi_{\max}, if \quad \Phi + ((\sum_{j \in N(t)} r_j)/n - r_i) \cdot \Delta > \Phi_{\max} \\ 0, \quad if \quad \Phi + ((\sum_{j \in N(t)} r_j)/n - r_i) \cdot \Delta < 0 \\ \Phi + ((\sum_{j \in N(t)} r_j)/n - r_i) \cdot \Delta, \quad otherwise \end{cases}, \tag{5.4}$$

in which Φ is the default ratio of sacrificing services and is set within the range of 0 to $\Phi_{\max}$, where $\Phi_{\max}$ is the maximum sacrificing ratio. Δ in (5.4) is the degree of the influence of the weight. In other words, the larger the value of Δ is, the more influence r_i has on Φ'_i. Therefore WSFQ can be adapted to different traffic density by controlling the value of Δ. For example, if the traffic is heavy, the value of Δ needs to be increased; if the traffic is low, the value of Δ needs to be decreased.

With the time-varying weight $v(t)$, the services served for session i in WSFQ in a time interval $(t_0, t]$ can be defined as follows (the state for each session in WSFQ is not changed in $(t_0, t]$).

$$\frac{\partial W_i(t_0, t)}{\partial t} = v_i(t) \cdot \frac{\partial V'}{\partial t}, \tag{5.5}$$

where V' is the virtual time of WSFQ system, and satisfies

$$\frac{\partial V'}{\partial t} = \begin{cases} \dfrac{A}{\sum_{j \in G(t)} v_j(t)} & if \ \sum_{j \in G(t)} v_j(t) \neq 0 \\ 0 & otherwise \end{cases}. \tag{5.6}$$

Each session in WSFQ is associated with a parameter *delay* to measure the difference between the services that a session should receive in a referenced error-free network and the services it has received in the WSFQ. If the value of $delay_i$ is positive, then we say session i is in *lagging* state. If the value is negative, then session i is in *leading* state. If the value is exactly zero, then the session is in *synchronizing* state.

In WSFQ, if a non-lagging session i has a high weight (higher than the average weight of all non-lagging sessions), the value of $((\sum_{j \in N(t)} r_j)/n - r_i) \cdot \Delta$ is negative, thus the value of Φ_i' is smaller than the default ratio Φ. On the other hand, if a non-lagging session has a low weight (lower than the average weight of all non-lagging sessions), the value of Φ_i' is larger than the default ratio Φ. We can easily find that the sacrificing ratio of a high weighted non-lagging session is lower than that of a low weighted non-lagging session. Thus a high weighted non-lagging session has better service rate at the low cost of less service rate associated to low weighted sessions. The problem of crowded queue in high weighted sessions is improved with little influence on other sessions. Moreover, since the sacrificing ratio is limited to be smaller than $\Phi_{\max}$, all non-lagging sessions have graceful degradation in quality of services while it is sacrificed for compensation.

5.2.2 Fairness and Delay Analysis

In this subsection, we investigate the fairness property of WSFQ in Theorem 5.1 by proving the bounded difference in the services served to arbitrary two sessions. The arbitrary two sessions can be both in non-lagging states or one is in lagging state and the other is not. In Theorem 5.2 we show that within this fairness property, high weighted sessions have shorter queue length and smaller queueing delay in WSFQ than in other PFQ algorithms. At last in Theorem 5.3, the growth speed of queue length in WSFQ is proved to be slower than that in other PFQ algorithms.

Theorem 5.1 Let t_0 and t_1 be any two instants in the same *steady period* and $t_0 < t_1$. Assume that session *i* and session *j* are two servable sessions of the WSFQ. $\forall t \in (t_0, t_1)$, the value of $\left| \frac{W_i(t_0,t)}{r_i} - \frac{W_j(t_0,t)}{r_j} \right|$ satisfies:

(i) If $i \in C(t), j \in N(t)$ or $i \in N(t), j \in C(t)$, then

$$\left| \frac{W_i(t_0,t)}{r_i} - \frac{W_j(t_0,t)}{r_j} \right| \leq \Phi_{\max} \cdot (V(t_1) - V(t_0)). \tag{5.7}$$

(ii) If $i, j \in N(t)$, then

$$\left|\frac{W_i(t_0,t)}{r_i} - \frac{W_j(t_0,t)}{r_j}\right| \le \left|\Delta \cdot (r_i - r_j)\right| \cdot (V(t_1) - V(t_0)). \tag{5.8}$$

(iii) If $i, j \in C(t)$, then

$$\left|\frac{W_i(t_0,t)}{r_i} - \frac{W_j(t_0,t)}{r_j}\right| = 0. \tag{5.9}$$

Proof If $i \in C(t), j \in N(t)$, the time-varying weight $v_i(t) = r_i$, and $v_j(t) = r_j(1 - \Phi'_j)$. By applying the time-varying weight into (5.5),

$$\begin{cases} W_i(t_0,t) = r_i \cdot (V(t) - V(t_0)) \\ W_j(t_0,t) = r_j \cdot (1 - \Phi'_j)(V(t) - V(t_0)) \end{cases}. \tag{5.10}$$

Normalizing W_i and W_j by the life-time weight r_i and r_j, we have

$$\begin{cases} \dfrac{W_i(t_0,t)}{r_i} = (V(t) - V(t_0)) \\ \dfrac{W_j(t_0,t)}{r_j} = (1 - \Phi'_j)(V(t) - V(t_0)) \end{cases}. \tag{5.11}$$

Subtracting $W_j(t_0,t)/r_j$ from $W_i(t_0,t)/r_i$ then we have

$$\begin{aligned} &\frac{W_i(t_0,t)}{r_i} - \frac{W_j(t_0,t)}{r_j} = \Phi'_j \cdot (V(t) - V(t_0)) = \\ &(\Phi + ((\sum_{m \in N(t)} r_m)/n - r_j) \cdot \Delta) \cdot (V(t) - V(t_0)) \le \\ &\Phi_{\max} \cdot (V(t) - V(t_0)) \le \Phi_{\max} \cdot (V(t_1) - V(t_0)). \end{aligned} \tag{5.12}$$

Considering the case of $i \in N(t), j \in C(t)$, we get (5.7). If $i, j \in N(t)$, the (5.12) is modified as follows:

$$\left|\frac{W_i(t_0,t)}{r_i}-\frac{W_j(t_0,t)}{r_j}\right| = \left|\Phi'_j - \Phi'_i\right| \cdot (V(t)-V(t_0)) = \\ \left|r_i - r_j\right| \cdot \Delta \cdot (V(t)-V(t_0)) \le \left|r_i - r_j\right| \cdot \Delta \cdot (V(t_1)-V(t_0)). \tag{5.13}$$

If $i, j \in C(t)$, there is no difference between $W_j(t_0,t)/r_j$ and $W_i(t_0,t)/r_i$, therefore we get (5.9). □

Theorem 5.1 shows that WSFQ is fair between two sessions if both of them are being compensated. If both of the sessions are in the non-lagging state, or only a session is being compensated, the services difference is bounded in (5.7) and (5.8) respectively.

Theorem 5.2 In WSFQ, if a non-lagging session whose weight is higher than the average weight of all non-lagging sessions, then it has shorter queue length and smaller queueing delay compared to CIF-Q-like queueing policies.

Proof Each session is considered to be a single server with individual arrival rate and service rate. All sessions connect to a main server. The arrival rate of the main server is equal to the summation of all sessions' service rates, and the service rate of the main server is equal to the system capacity. The total arrival rates and system capacity are limited to satisfy

$$\sum_i \lambda_i = \lambda_{total} \le A, \tag{5.14}$$

in which λ_i is the arrival rate of session *i*, A is the system capacity (total service rate).

In an error-free system, neither compensation nor sacrificing takes place. Thus the service rate of each session depends on its own weight. That is, if the system capacity is presented as *A*, the service rate of session *i* is equal to $A \cdot r_i / \sum_j r_j$. However, in an error-prone system, the lagging session *k* is compensated by increasing its service rate to be more than $A \cdot r_k / \sum_j r_j$ when it becomes servable, and the non-lagging session *l* is sacrificed by decreasing its service rate to be less than $A \cdot r_l / \sum_j r_j$. Assume that the service rate of the non-lagging session l is $(A \cdot r_l / \sum_j r_j) - \alpha_l$. The value of α_l is determined according to the PFQ algorithm employed.

In IWFQ and CIF-Q, due to the bound of leading services of each session, the value of α_l of non-lagging session l is proportional to its weight. Thus α_l is presented as follows in the CIF-Q or in the IWFQ.

$$\alpha_l = r_l \cdot \alpha_l' \qquad (\alpha_l' < A \cdot / \sum_i r_i). \tag{5.15}$$

In the WSFQ, the denotation of α_l is different from (5.15) and is expressed as follows.

$$\alpha_l = r_l \cdot (\alpha_l' + \beta_l), \tag{5.16}$$

where β_l is expressed as follows:

$$\beta_l = ((\sum_{i \in N(t)} r_i) / n - r_l) \cdot \Delta. \quad . \tag{5.17}$$

If a non-lagging session whose weight is higher than the average weight of all non-lagging sessions ($r_l > (\sum_{j \in N(t)} r_j) / n$), the value of β_l is negative. By comparing $r_l \cdot \alpha_l'$ with $r_l \cdot (\alpha_l' + \beta_l)$, we find that $r_l \cdot (\alpha_l' + \beta_l) < r_l \cdot \alpha_l'$. Thus the following relationship is derived.

$$(A \cdot r_l / \sum_i r_i) - r_l \cdot (\alpha_l' + \beta_l) > (A \cdot r_l / \sum_i r_i) - r_l \cdot \alpha_l'. \tag{5.18}$$

Let μ_l^w be the service rate of session l in WSFQ and μ_l^c be the service rate of session l in CIF-Q, we have

$$\mu_l^w > \mu_l^c. \tag{5.19}$$

Considering the session l as a single server with M/D/1 queueing model, its arrival rate is λ_l. Let ρ_l^w and ρ_l^c be the *traffic intensity* in WSFQ and in CIF-Q such that $\rho_l^w = \lambda_l / \mu_l^w$ and $\rho_l^c = \lambda_l / \mu_l^c$. By applying (5.19), we have

$$\rho_l^w < \rho_l^c. \tag{5.20}$$

Notice that the expected value of queue length for M/D/1 queue is given by

$$E[N] = \rho + \frac{\rho^2}{2(1-\rho)}. \tag{5.21}$$

Therefore, let $E_l^w[N]$ be the expected queue length of session l in WSFQ and $E_l^c[N]$ be the expected queue length of session l in CIF-Q. Then $E_l^w[N]$ and $E_l^c[N]$ satisfy:

$$\begin{cases} E_l^w[N] = \rho_l^w + \dfrac{\rho_l^{w\,2}}{2(1-\rho_l^w)} \\ E_l^c[N] = \rho_l^c + \dfrac{\rho_l^{c\,2}}{2(1-\rho_l^c)} \end{cases}. \tag{5.22}$$

By applying (5.20) into (5.22), we have

$$E_l^w[N] < E_l^c[N]. \tag{5.23}$$

Thus queue length of session l (high weighted session) in WSFQ is shorter than that in CIF-Q. According to *Little's Formulas*, if the mean of the queue length in WSFQ is smaller than that in CIF-Q, the mean of the queueing delay in WSFQ is also smaller than which in CIF-Q. □

Theorem 5.3 In WSFQ, if a non-lagging session whose weight is higher than the average weight of all non-lagging sessions, then it grows more gently in queue length comparing to CIF-Q-like queueing policies while increasing the arrival rate of this session.

Proof The derivative of $E[N]$ *is* presented as $E'[N]$, which represents the growth-gradient of the queue length in a queue and is derived in (5.24).

$$E'[N] = \frac{dE[N]}{d\rho} = 1 + \rho(2-\rho)(1-\rho)^{-2}. \tag{5.24}$$

Let $E_l'^w[N]$ and $E_l'^c[N]$ be the growth-gradient of the queue length in the session *l* by applying WSFQ and CIF-Q respectively. They are denoted as follows.

$$\begin{cases} E_l^w[N] = \rho_l^w + \dfrac{\rho_l^{w^2}}{2(1-\rho_l^w)} \\ E_l^c[N] = \rho_l^c + \dfrac{\rho_l^{c^2}}{2(1-\rho_l^c)} \end{cases}. \tag{5.25}$$

In order to get the relationship between $E_l'^w[N]$ and $E_l'^c[N]$, we denote with the derivative of $E'[N]$ with $E''[N]$ in (5.26).

$$E''[N] = \frac{dE'[N]}{d\rho} = (2-2\rho)(1-\rho)^{-2} + 2(2\rho-\rho^2)(1-\rho)^{-3}. \tag{5.26}$$

Since the value of ρ is less than 1 ($0 \le \rho \le 1$), the value of $E''[N]$ is always positive ($E''[N]$>0). It implies that $E'[N]$ is an incremental function. Since $E'[N]$is an incremental function and $\rho_l^w < \rho_l^c$ (which is given by (5.20)), we find the relationship that $E_l'^w[N]$<$E_l'^c[N]$. Which means WSFQ grows gently in queue length by comparing that in CIF-Q while the arrival rate of the high weighted sessions are increased.

In addition**,** the service time (1/*A*) may either be a constant value or obey the exponential distribution. If it follows the exponential distribution, replace the equation in (5.21) with the equation (5.27) of the M/M/1 model. Then the theorem 5.2 and 5.3 can be proved by the proof steps described above.

$$E[N] = \frac{\rho^2}{1-\rho} \quad . \tag{5.27}$$

□

Theorem 5.1 shows the property of short-term fairness between all sessions in the WSFQ. Moreover, Theorem 5.2 and 5.3 illustrate that the high weighted sessions have better performance in WSFQ than in other PFQ algorithms in terms of queue length. These theorems claim the significance of WSFQ while performing scheduling in WLAN stations since the packet loss is mainly caused by the overflow of the high weighted sessions.

5.3 Packized WSFQ Algorithm

In this section, we propose a packetized weighted sacrificing fair queueing algorithm (PWSFQ) that realized WSFQ by practical packet-by-packet scheduling. The procedure of the PWSFQ algorithm is described as follows:

(i) Initially, all sessions are synchronized. But soon some of them change their states to become leading, lagging, or synchronizing due to the channel failures.

(ii) The SFQ scheduler that is implemented by the reference system S_r schedules the packets from each session and outputs the sequence of the packets. This procedure is shown in Figure 5.2.

(iii) With the inputs of the sequence of backlogged sessions and the information of the states of all sessions, PWSFQ performs the fair scheduling. The sequence of backlogged sessions is decided by the sequence of the packets in the previous step.

(iv) The fair scheduling of the PWSFQ includes the following procedures: the procedure of non-lagging sessions, the procedure of compensation, the procedure of extra services, and the procedure of lagging sessions. The details are described in Section 5.3.1 and 5.3.2.

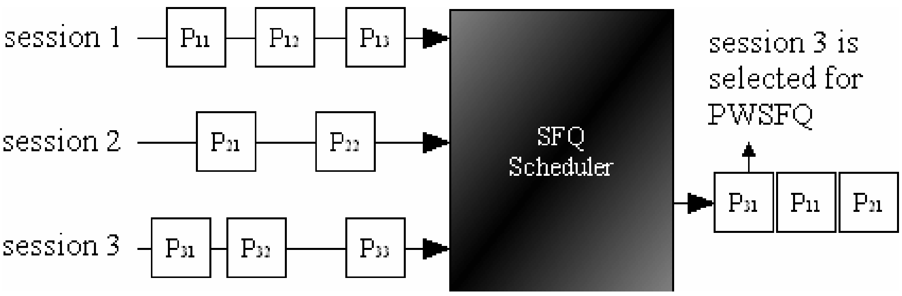

Figure 5.2: Illustration of the SFQ scheduler.

In order to account for the service lost or gained by a session due to errors, we associate with PWSFQ a reference error-free system S_r that has been described in the previous section. PWSFQ determines the states of all sessions according to the service difference of each session between the real system and the reference system. All parameters used in PWSFQ algorithm are summarized in Table 5.1.

Table 5.1: Parameters used in PWSFQ.

Parameter	Definition
R_i	Service count of a leading session *i*.
C_i	Sacrificing testing of a non-lagging session *i*.
C_{min}	Lower bound of C_i
$lead_i$	Amount of leading services of leading session *i*.
lag_i	Amount of lagging services of lagging session *i*.
O_i	The priority of a lagging session *i*.
O_{max}	Upper bound of O_i
L_i	The packet length of the head-of-line packet in a session *i*.

The parameter $lead_i$ represents the amount of services that a leading session *i* leads by comparing it to S_r. The parameter lag_i represents the amount of services that a lagging session *i* lags by comparing it to S_r. At any time, if the value of $lead_i$ becomes a negative integer, session *i* becomes lagging and the parameter lag_i is assigned the value of $|lead_i|$. On the other hand, if the value of lag_i becomes a negative integer, session *i* becomes leading and the parameter $lead_i$ is assigned the value of $|lag_i|$. Parameters C_i and R_i are both configured for leading sessions. The parameter C_i is the checking flag to determine the time when a non-lagging session *i* is about to do the sacrificing; R_i is a count which keeps the amount of services that the session *i* has already received. If the value of R_i exceeds C_i, the sacrificing process takes place and the value of R_i is reset to be zero. The C_i is configured according to

$$C_i = k_1 + (r_i - \sum_{j \in N(t)} r_j / n) \cdot k_2 . \qquad (5.28)$$

C_i is bounded by C_{min}, k_1 and k_2 are constant values that implement the values of Φ and

Δ in (5.4).

For a lagging session i, we use parameter O_i to indicate its priority to receive compensation services. Let $O_i = Lag_i \times r_i$. The value of O_i is bounded by O_{max}.

5.3.1 Non-Lagging Sessions

The procedure that tackles the packets of a non-lagging session is shown in Figure 5.3. At first the parameter R_i is increased by the value of L_i, then the value of R_i is checked to see if it exceeds C_i. If it exceeds, R_i is reset to be zero, $Lead_i$ is subtracted by L_i, and the procedure of compensation is performed. If it does not exceed, this packet is allowed to transmit. However, if the packet cannot be sent because of the channel errors, PWSFQ subtracts L_i from $Lead_i$ and the procedure of compensation is performed. Since the value of C_i is bounded by C_{min}, even if a session leads by a large amount of services, the penalty it would pay is bounded. With the upper bound, the property of graceful degradation is observed.

Figure 5.4 depicts the procedure of compensation. At first the PWSFQ checks the numbers of lagging sessions in the system. If there is at least one lagging session, PWSFQ continues with the current procedure. After that PWSFQ picks the session k with the maximum O_k among all lagging sessions and checks whether the channel state is good enough for this session. If it is good, the head-of-line (HOL) packet of the session is transferred and Lag_k is subtracted by L_k, but if the session k is not servable due to the channel errors, PWSFQ picks the next lagging session that has maximum priority except the session k until there is no lagging session. This procedure terminates and another procedure called “the procedure of extra services” starts if no lagging session passes the check.

The procedure of extra services is almost the same as the procedure of compensation. This time PWSFQ picks the session m whose $Lead_m$ is the minimum among all available sessions, transfers the packet of session m, and adds the value of L_m to $Lead_m$. Figure 5.5 depicts the whole procedure of extra services.

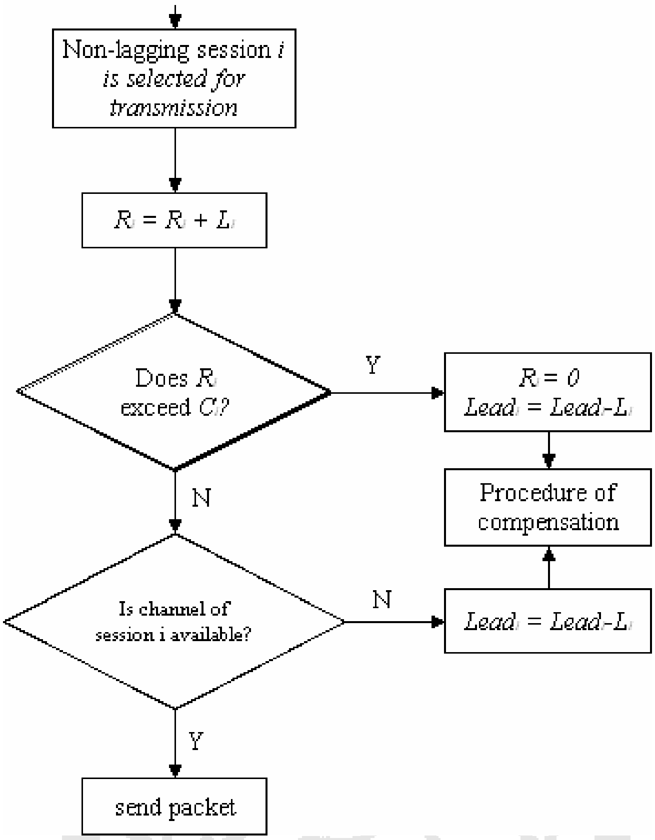

Figure 5.3: The procedure of non-lagging sessions.

5.3.2 Lagging Sessions

The procedure of lagging sessions is depicted in Figure 5.6. PWSFQ only needs to check whether the current session *i* is servable. If it is, the PWSFQ transfers the HOL packet of the session *i*; if it is not, PWSFQ adds Lag_i the value of L_i and starts the procedure of extra services.

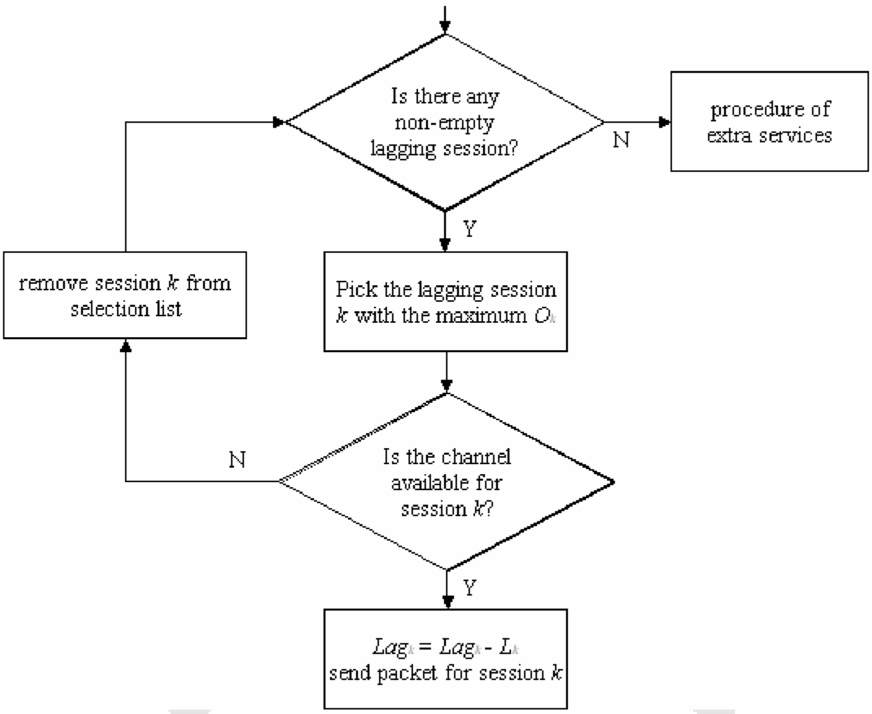

Figure 5.4: The procedure of compensation.

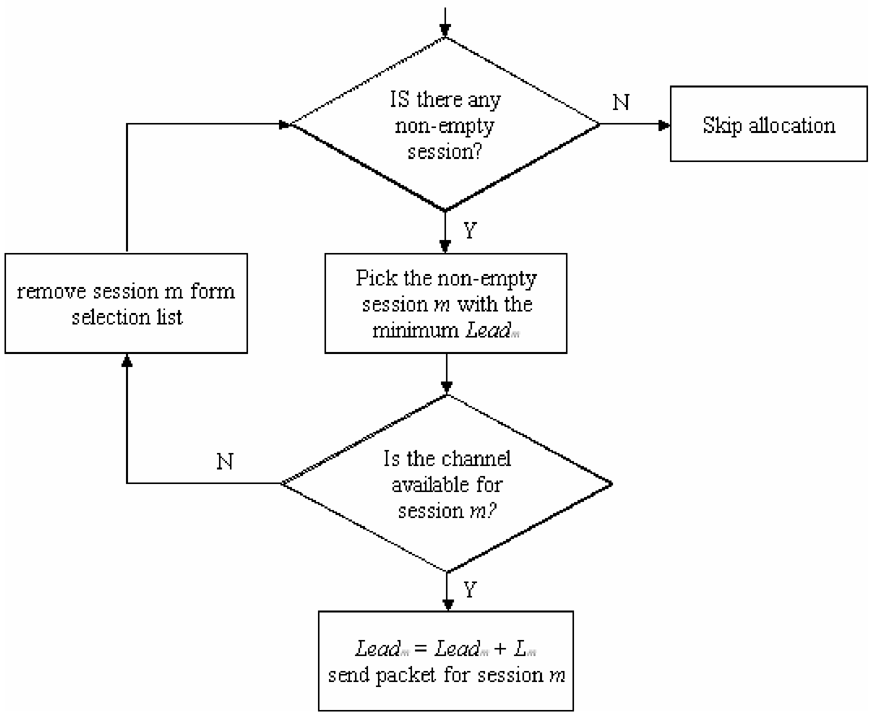

Figure 5.5: The procedure of extra services.

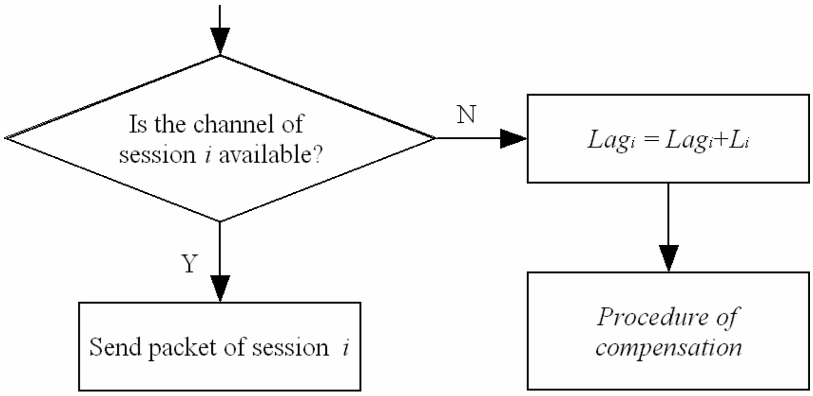

Figure 5.6: The procedure of lagging sessions.

5.4 Performance Analysis

In this section, we present the simulation results obtained by the Stochastic Petri Net (SPN) to demonstrate the queueing behaviors of various kinds of sessions thorough PWSFQ and CIF-Q. We discuss two scenarios of the simulation in the following. One is the full-length simulation and the other one is focusing on the period of an error-free circumstance in which all sessions are backlogged.

5.4.1 Full-Length Simulation

Consider the scenario with three sessions: one text session and two image sessions in the system. The default properties of each session are listed in Table 5.2 and the details of all parameters used in scenario A are listed in the Table 5.3.

Table 5.2: Properties of three sessions in scenario A.

	Weight	Session model	Error
Text session	1	Poisson	None
Image session A	2	Poisson	None
Image session B	2	Poisson	30% of channel error

Table 5.3: Parameters used in scenario A.

Parameter	Description	Default value
total_service_rate	Total service rate of the system (Poisson)	2.5
r_1	Weight of image session A	2
r_2	Weight of text session	1
r_3	Weighted of image session B	2
r_total	Summation the weight of all sessions	r_1+r_2+r_3
a_1	Arrival rate of image session A	0.5
a_2	Arrival rate of text session	0.25
a_3	Arrival rate of image session B	0.5
s_1	Pure service rate of image session A	total_service_rate* (r_1/r_total)
s_2	Pure service rate of text session	total_service_rate* (r_2/r_total)
s_3	Pure service rate of image session B	total_service_rate* (r_3/r_total)
sac_ratio	The default sacrificing ratio of leading session (k_1 in (5.28))	0.5
k_2(5.28)	The parameter for the implementation of adaptability	0.6666666….
err	The probability of channel error of image session B	30%

We assign the text session with the weight of 1 and the image sessions with the weight of 2. This implies that the amount of services needs to be served in the image sessions is twice as much as that in the text session. For the purpose of simulation, we assume the channel of the image session B has the probability of 30% to have errors and the other channels have no error, and the packet sizes of all packets are identical. The SPN model of each session in this scenario is depicted in Figure 5.7. In scenario A, three sessions are considered. Therefore, three SPN models are required. Notice that the three models are identical except the identifiers of the places and transitions. The complete legends of places and transitions in three sessions are given in Table 5.4, and the firing rates of all transitions are given in the Table 5.5. The multiplicity inhibitor arc from p_1 to t_0 defines the buffer capacity of a session. The value *m* is the maximum numbers of packets that a buffer can restore, if the numbers of packets exceed the value of *m*, the packet loss occurs.

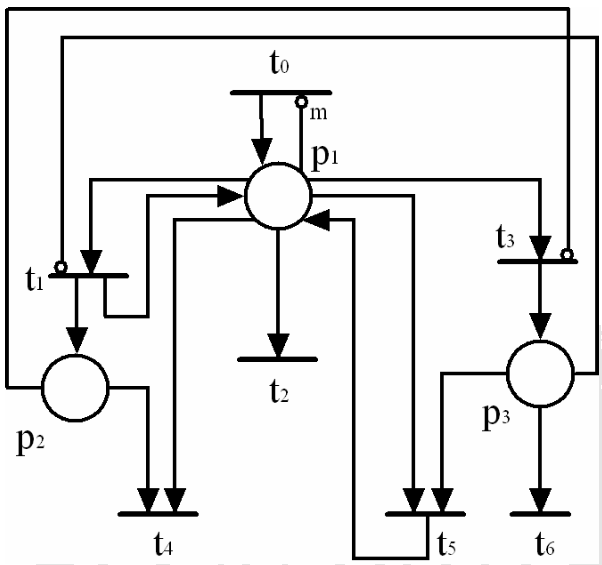

Figure 5.7: SPN model of each session in scenario A.

Table 5.4: Definitions of places and transitions.

Image A	Text	Image B	Descriptions
p_1	p_4	p_7	Numbers of packets queued in buffer.
p_2	p_5	p_8	Lagging count.
p_3	p_6	p_9	Leading count.
t_0	t_7	t_{14}	Arrival of a packet.
t_1	t_8	t_{15}	An occurrence of channel errors.
t_2	t_9	t_{16}	Transmitting a packet.
t_3	t_{10}	t_{17}	Transmitting a packet by the procedure of extra services.
t_4	t_{11}	t_{18}	Transmitting a packet by the procedure of compensation.
t_5	t_{12}	t_{19}	Sacrificing a transmission for compensation.
t_6	t_{13}	t_{20}	An occurrence of channel errors.

Table 5.5: Firing rate and enable functions of transitions.

Transition	Firing rate	Enabling function
t_0, t_7, t_{14}	t_0:=a_1, t_7:= a_2, t_{14}:= a_3,	None
t_2 (PWSFQ)	If $p_3 = 0$, then t_2:=s_1, else t_2 := s_1*(1-(sac_ratio+ (AvgWeightofLeadingSession()-r_1)* k_2))	None
t_2 (CIF-Q)	If $p_3 = 0$, then t_2:=s_1, else t_2 := s_1*(1-(sac_ratio))	None
t_3 (PWSFQ)	If there is no leading session, then t_3:=s_3*err*(s_1/(s_1+s_2)), else t_3:= (s_3+ t_5+ t_{12})*err*(p_6/(p_6+p_3))	p_6!=0 ||(p_6=0&& p_3=0)
t_3 (CIF-Q)	If there is no leading session, then t_3:=s_3*err*1/2, else t_3:= (s_3+ t_5+ t_{12})*err*(p_6/(p_6+p_3))	p_6!=0 ||(p_6=0&& p_3=0)
t_5	s_1- t_2	None
t_9 (PWSFQ)	If $p_6 = 0$, then t_9:=s_2, else t_9 := s_2*(1-(sac_ratio+ (AvgWeightofLeadingSession()-r_2)* k_2))	None
t_9 (CIF-Q)	If $p_6 = 0$, then t_9:=s_2, else t_9 := s_2*(1-(sac_ratio))	None
t_{10} (PWSFQ)	If there is no leading session, then t_{10}:=s_3*err*(s_2/(s_1+s_2)), else t_{10}:= (s_3+ t_5+ t_{12})*err*(p_3/(p_6+p_3))	p_3!=0 ||(p_3=0&& p_6=0)
t_{10} (CIF-Q)	If there is no leading session, then t_{10}:=s_3*err*1/2, else t_{10}:= (s_3+ t_5+ t_{12})*err*(p_3/(p_6+p_3))	p_3!=0 ||(p_3=0&& p_6=0)
t_{12}	s_2- t_9	None
t_{15}	err * (s_3)	None
t_{16}	(1-err) * (s_3)	None
t_{18}	(t_5+t_{12})*(1.0-err)	p_3!=0 || p_6!=0

To compare the performance of PWSFQ with that of CIF-Q, we approximate the CIF-Q algorithm by using the same SPN model. The major difference is the firing rates of the transitions t_2, t_3, and t_5. Thus we reassign these faring rates according to the policy of CIF-Q. The details of assignment are shown in the Table 5.5. Moreover, the value of the system parameter α in full version CIF-Q is set to be 0.5 in order to approximate the configurations of the PWSFQ.

We run the SPN to solve the corresponding Markov chain with the minimum precision of 10^{-6}, and analyze the transient results of the simulation. The performance of queue length and packet loss (for $m=6$) for each session under different packet arrival rates is shown from Figure 5.8 to Figure 5.11. Here the packet loss is caused only by the limited buffer size. The performance of queue length of all sessions in the PWSFQ when the network-adaptive parameter k_2 (in the Table 5.3) is assigned with the value from 0.1 to 0.7 is shown in Figure 5.12. Our main conclusions from the simulation results are described as follows.

By Comparing PWSFQ with approximated CIF-Q by the increasing arrival rate of the image session A, the PWSFQ performs better on the queue length and the packet loss than the approximated CIF-Q does. Oppositely, the approximated CIF-Q has shorter queue length and smaller packet-loss rate than the PWSFQ does in the case of text session. However, the difference between PWSFQ and CIFQ of the high weighted sessions (Image A) is more obvious than the difference of the low weighted sessions (the Text session). This is because that the traffic circumstance has more influence on high weighted sessions than on low weighted sessions, and PWSFQ has the agency for adapting the heavy traffic, especially for high weighted sessions.

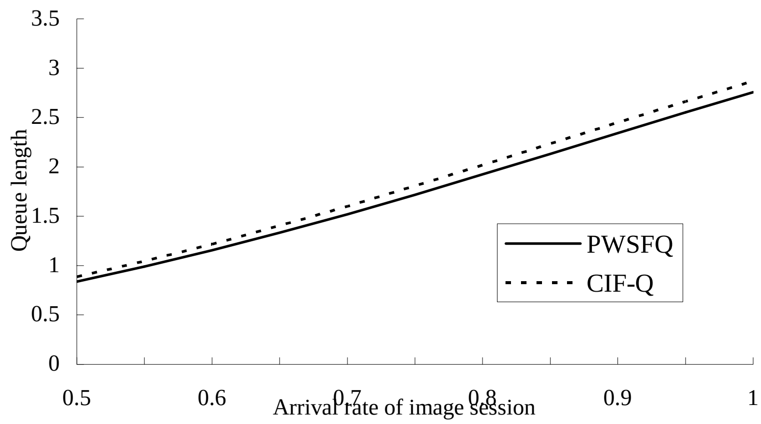

Figure 5.8: Queue length of Image Session A.

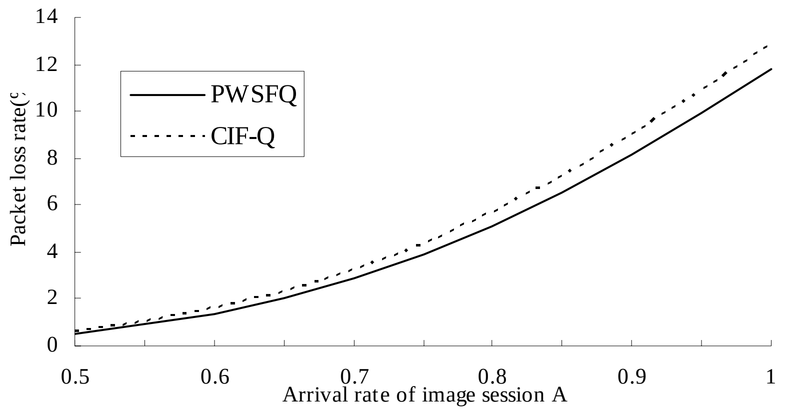

Figure 5.9: Packet loss of Image session A.

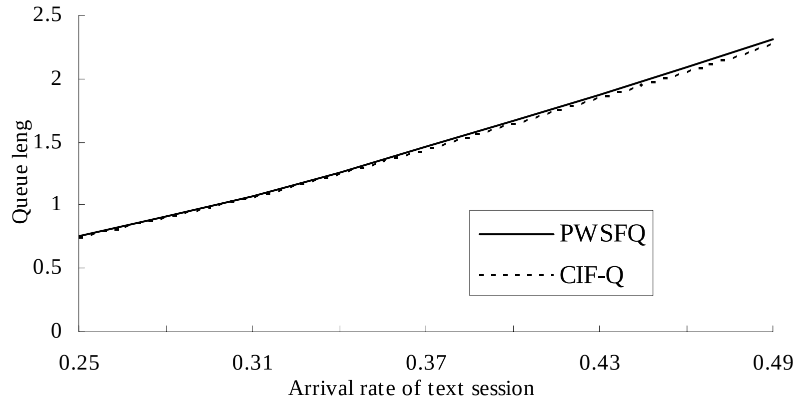

Figure 5.10: Queue length of the Text session.

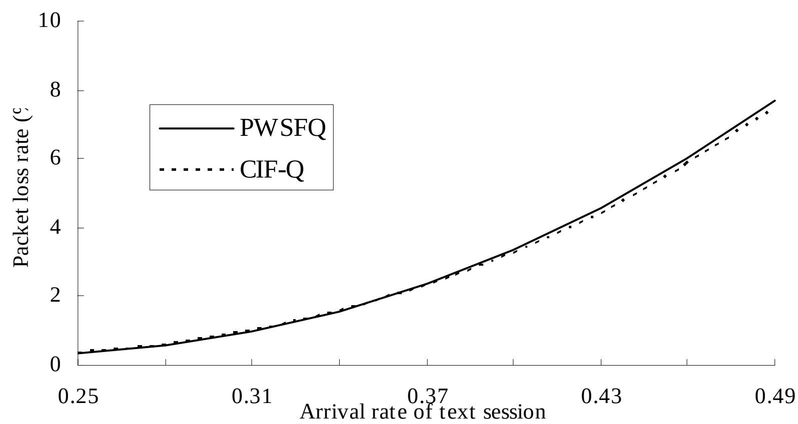

Figure 5.11: Packet loss of the Text session.

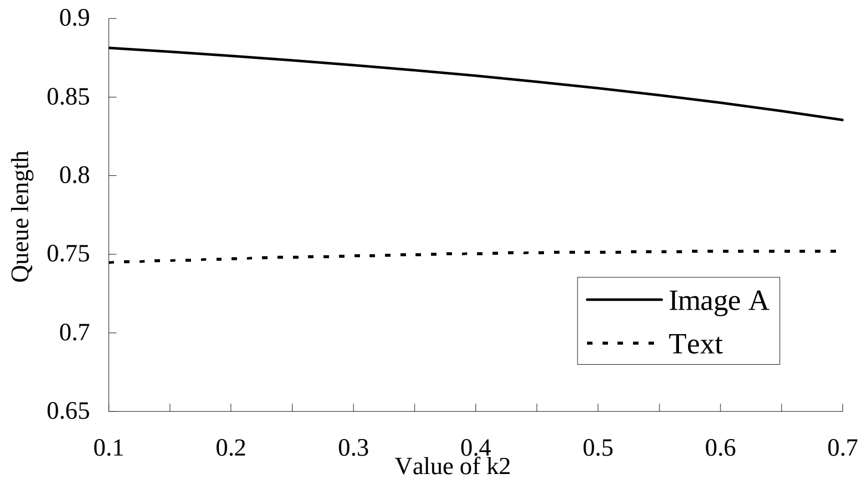

Figure 5.12: Queue length performance of PWSFQ with various values of k_2.

In the case of lagging sessions, the performance of queue length and packet loss in CIF-Q is better than that in PWSFQ. This is because that there is only one low weighted session in the scenario. We will discuss this issue father at the next scenario.

5.4.2 Scenario B: Simulation of an Error-Free Period

In this scenario, we would like to focus on the period in which no channel error occurs to any session, and to analyze the simulation results about the queueing behaviors of the leading sessions. Seven sessions are considered for simulation: four are leading sessions and the others are lagging sessions. Among these leading sessions, one is an image session and the other three are text sessions. Among lagging sessions, two are image sessions and one is a text session. The detailed information of parameters used in the scenario is listed in the Table 5.6.

The SPN model of each leading session in this scenario is depicted in Figure 5.13. In scenario B, four leading sessions are considered. Therefore, four SPN models are required. The complete legends of places and transitions in the four leading sessions are given in Table 5.7, and the firing rates of all transitions are given in the Table 5.8 and Table 5.9. The multiplicity inhibitor arc from p_1 to t_0 defines the buffer capacity of a session.

Table 5.6: Parameters used in Scenario B.

Parameter	Description	Default value
total_service_rate	Total service rate of the system (Poisson)	10.0
r_text	Weight of image session A	1
r_image	Weight of text session	2
r_total	Summation the weight of all sessions	r_text*4+r_image*3
a_text	Arrival rate of text session	0.15
a_image	Arrival rate of image session	0.075
s_text	Pure service rate of image session A	total_service_rate*(r_text/r_total)
s_image	Pure service rate of text session	total_service_rate*(r_image/r_total)
sac_ratio	The default sacrificing ratio of leading session	0.5
k_2(5.28)	The parameter for the implementation of adaptability	1

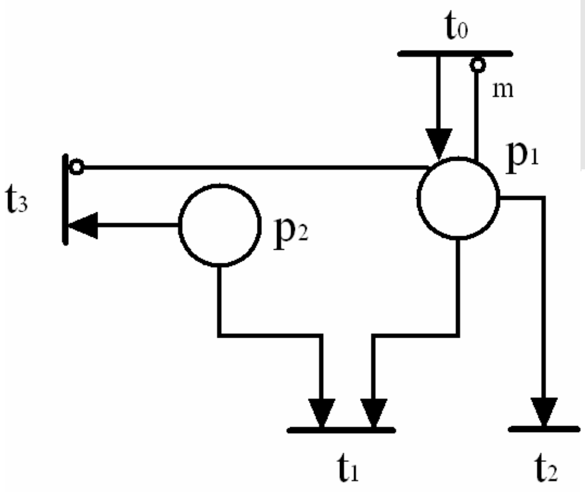

Figure 5.13: SPN model of each leading session in scenario B.

Table 5.7: Definitions of places and transitions of Figure 5.13.

	Descriptions
p_1	Number of packets queued in buffer
p_2	Leading count
t_0	Arrival of a packet
t_1	Sacrificing a transmission of packet for the lagging sessions
t_2	Transmitting a packet
t_3	Sacrificing a transmission of packet for the lagging sessions while there is no packet queued in buffer

Table 5.8: Firing rate and enable functions of transitions for text sessions.

Transition	Firing rate	Enabling function
t_0	t_0:= a_text	None
t_1	s_text-t_2	p_2>0
t_2 (PWSFQ)	If p_2 = 0, then t_2:=s_text, else t_2 := s_text*(1-(sac_ratio+(AvgWeightofLeadingSession()-r_text)* k_2))	None
t_2 (CIF-Q)	If p_2 = 0, then t_2:=s_text, else t_2 := s_text*(1-(sac_ratio))	None
t_3	t_1	p_1=0

Table 5.9: Firing rate and enable functions of transitions for image sessions.

Transition	Firing rate	Enabling function
t_0	t_0:= a_image	None
t_1	s_image-t_2	p_2>0
t_2 (PWSFQ)	If p_2 = 0, then t_2:=s_image, else t_2 := s_image*(1-(sac_ratio+(AvgWeightofLeadingSession()-r_image)* k_2))	None
t_2 (CIF-Q)	If p_2 = 0, then t_2:=s_image, else t_2 := s_image*(1-(sac_ratio))	None
t_3	t_1	p_1=0

We run the SPN of each leading session independently to solve the corresponding Markov chain with the minimum precision of 10^{-6}, and analyze the transient results of the simulation. Each leading session is assumed to have 20 leading packets. The performance of queue length and packet loss (for *m=12*) for each leading session under different packet arrival rates is shown from Figure 5.14 to Figure 5.17. The total amount of sacrificed services per time unit of all leading sessions with various values of k_2 is shown in Figure 5.18. The simulation results shown from Figure 5.14 to Figure 5.17 are similar to the results in the scenario A, but more obviously the difference between high weighted sessions and low weighted sessions is. Figure 5.18 shows that the compensation services per time unit in PWSFQ increase when the value of k_2 becomes bigger.

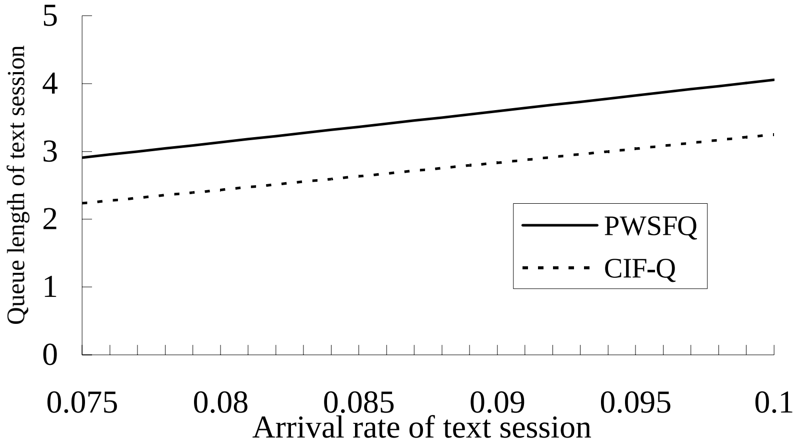

Figure 5.14: Queue length of the Text sessions.

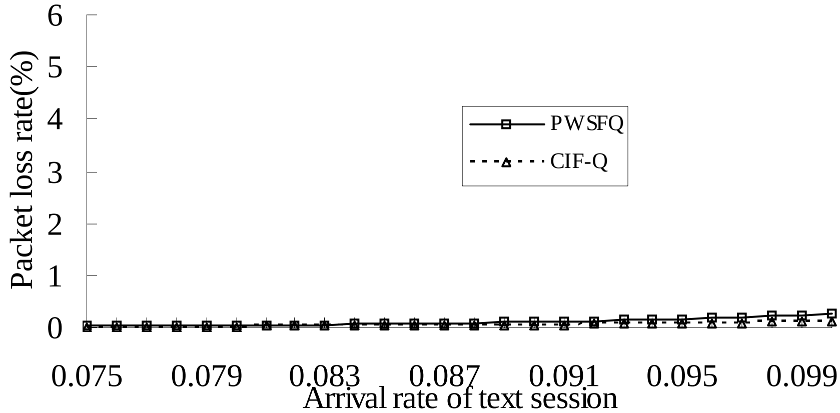

Figure 5.15: Packet loss of the Text sessions.

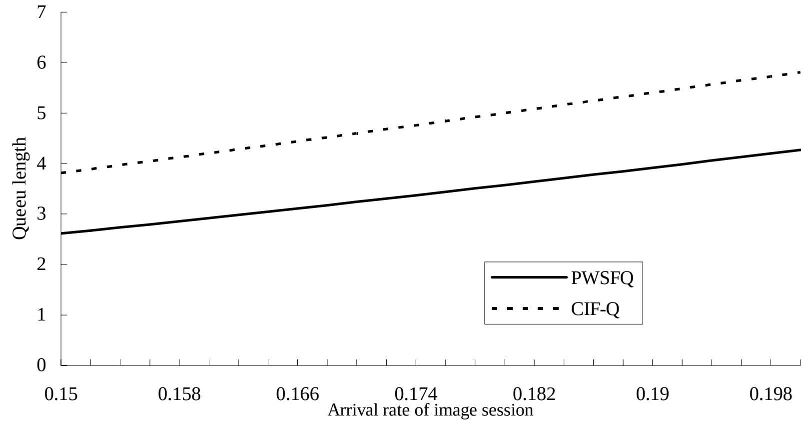

Figure 5.16: Queue length of the Image sessions.

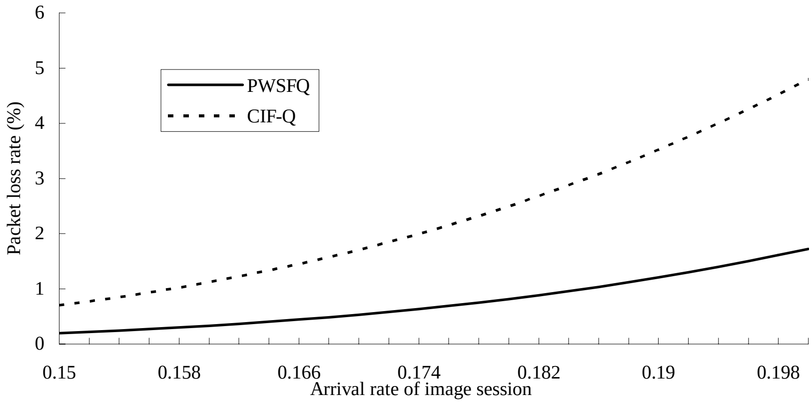

Figure 5.17: Packet loss of the Image sessions.

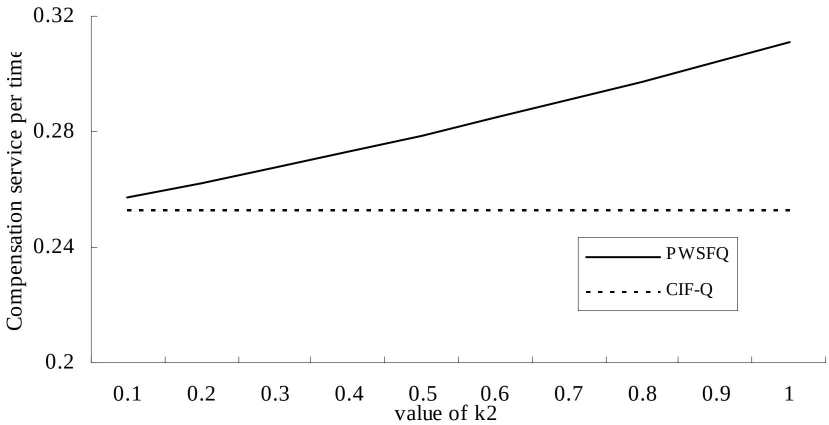

Figure 5.18: Total amounts of compensation services per time unit versus k_2.

5.5 Summary

In this chapter, we developed the WSFQ algorithm to improve the performance of queue length and queueing delay of the high weighted real-time sessions exposed to high traffic density. We also implemented the packetized fair scheduling algorithm that approaches the WSFQ. The mathematical analysis showed the fairness property and within this property, the high weighted sessions had better performance in WSFQ than in other PFQ algorithms. The simulation results showed that the PWSFQ performed well on the high weighted sessions at low cost of queueing delay of other low weighted leading sessions. Moreover, WSFQ can easily adapt itself to various kinds of traffic load. Thus applying the WSFQ into the highly variant WLANs is more suitable than applying the others.

Chapter 6

Conclusions

Since IEEE 802.11 WLAN is being accepted widely and rapidly for many different environments, it attracts many interests in some advanced issues. QoS in WLANs is one of these issues due to limited network bandwidth and unpredictable radio interference. The major goal of this dissertation is to provide QoS supports in multihop WLANs. To this end, we proposed a scheduling-based MAC, a multihop mechanism, and a delay-sensitive fair queueing algorithm in this dissertation.

The proposed MAC protocol uses a scheduling-based scheme instead of the polling-based scheme on managing transmissions to avoid the overheads of polling packets. In addition, to exploit the channel utilization, the proposed protocol allows peer-to-peer transmissions under the coordination of the coordinator node. Our protocol is composed of the weighted scheduling algorithm and QoS enhanced admission control algorithm. The weighted scheduling algorithm is designed to efficiently utilize the network bandwidth and fairly schedule the transmission for various services while the QoS-enhanced admission control algorithm is proposed to manage resources and guarantee the QoS requirements of services.

To eliminate the restriction on single hop transmission of the proposed scheduling-based MAC, we proposed the ACS multihop mechanism in this dissertation. The ACS mechanism that allowed the centralized MAC protocol to transmit data flows across subnets and alleviated the hidden terminal problem can be adapted to various network topologies. The ACS mechanism also eliminated the need for both complex initialization procedures and synchronization between subnets.

We observed that the QoS requirements of services guaranteed in the proposed MAC may be lost during multihop transmissions, since relay stations are not under the control of our scheduling algorithm. Therefore, we proposed the WSFQ for integrated services to meet their QoS requirements in multihop WLANs. WSFQ slows down the growth of queue length for real-time traffic, still maintains the property of fairness, and guarantees the throughputs of the station. Moreover, WSFQ can easily adapt itself to various traffic loads. Since the WSFQ is an abstract model, we proposed a packet-based scheduling algorithm, the Packetized Weighted Sacrificing Fair Queueing (PWSFQ), to approach the WSFQ in practice.

According to the comprehensive consideration in this dissertation, we provided a novel solution rather than distributed manner for providing QoS supports in multihop WLANs. With our solution, the high performance of real-time services in WLANs can be easily achieved.

References

[AC04] J.N. Al-Karaki and J.M. Chang, "EPCF: a Lightweight Multi-Priority PCF for QoS Support in IEEE 802.11 Wireless LANs", 2004 IEEE International Conference on Performance, Computing, and Communications, pp. 495-502, Phoenix, Arizona, April 2004.

[AKC00] C.W. Ahn, C.G. Kang and Y.Z. Cho, "Soft Reservation Multiple Access with Priority Assignment (SRMA/PA): A Novel MAC Protocol for QoS-Guaranteed Integrated Services in Mobile Adhoc Networks", Proceedings of IEEE VTC-Fall, Boston, MA, pp.942-947, New York, September 2000.

[Bia00] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", IEEE Journal of Selected Areas in Communications, vol. 18, no. 3, pp. 535-547, Mar 2000.

[BTC01] B. Bensaou, D.H.K. Tsang and K.T. Chan, "Credit-Based Fair Queueing (CBFQ): a Simple Service-Scheduling Algorithm for Packet-Switched Network", IEEE/ACM Transactions on Networking, vol.9, pp. 591-604, October 2001.

[BZ96] J.C.R. Bennett and H. Zhang, "WF2Q: Worst-Case Fair Weighted Fair Queueing", Fifteenth Annual Joint Conference of the IEEE Computer Societies (INFOCOM '96), pp. 120-128, San Francisco, CA, March 1996.

[CG00] R. Carces and J.J. Garcia-Luna-Aceves, "Collision Avoidance and Resolution Multiple Access for Multichannel Wireless Networks", Proceedings of IEEE Infocom 2000, pp. 595-602, March 2000.

[CK74] V. Cerf and R. Kahn, "A Protocol for Packet Network Intercommunications", IEEE Trans. on Communication, vol. 22, pp. 637-648, May 1974.

[CW04] Z.T. Chou and S.L. Wu, "A New QoS Point Coordination Function for Multimedia Wireless LANs", 24th International Conference on Distributed Computing Systems, Tokyo, pp. 40-47, Japan, March 2004.

[CWK+97] B.P. Crow, I. Widjaja, J.G. Kim and P.T. Sakai, "Investigation of the IEEE 802.11 Medium Access Control (MAC) Sublayer Functions", Proceedings of INFOCOM'97, vol. 1, pp. 126-133, 1997.

[DC99] D.J. Deng and R.S. Chang, "A Priority Scheme for IEEE 802.11 DCF access method", IEICE Transactions on Communications, pp. 96-102, 1999.

[DH98] J. Deng and Z. Haas, "Dual Busy Tone Multiple Access (DBTMA): A New Medium Access Control for Packet Radio Networks", Proceedings of IEEE ICUPC, Florence, Italy, 1998.

[DKS89] A. Demers, S. Keshav and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm", SIGCOMM Symposium on Communications Architectures and Protocols, pp. 1-12, Texas, USA, September 1989.

[DY05] D.J. Deng and H.C. Yen, "Quality-of-Service Provisioning System for Multimedia Transmission in IEEE 802.11 Wireless LANs", IEEE Journal on Selected Areas in Communications, vol. 23, pp. 1240-1252, 2005.

[FHF02] B. Fong, G.Y. Hong and A.C.M. Fong, "A Modulation Scheme for Broadband Wireless Access in High Capacity Networks", IEEE Transactions on Consumer Electronics, vol. 48, pp. 457-462, 2002.

[GVC96] P. Goyal, H.M Vin and H. Chen, "Start-Time Fair Queueing: a Scheduling Algorithm for Integrated Services", Proceedings of the ACM-SIGCOMM'96, pp. 157-168, Palo Alto, CA, August 1996.

[GW91] D.J. Goodman and S.X. Wei, "Efficiency of Packet Reservation Multiple Access", IEEE Trans. on Vehicular. Technology, vol. 40, pp. 170-176, Feb, 1991.

[HCG01] C. Heegrad, J. Coffey, S. Gummadi, P. A. Murphy, R. Provencio, E. J. Rossin, S. Schrum and M. B. Shoemake, "High Performance Wireless Ethernet", IEEE Communication Magazine, vol. 39, pp. 64-73, November, 2001.

[HLG02] W.C. Hung, K.L.E. Law and A.L. Garcia, "A Dynamic Multi-Channel MAC for Ad-Hoc LAN", Proceedings of the 21 st. Biennial Symposium on Communications pp. 31-35, April 2002.

[IEE02] IEEE draft std 802.15.3, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks", Inst. Elec. Electron. Eng, New York, USA, 2002.

[IEE03] IEEE 802.11 WG, "Draft Supplement to STANDARD FOR Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancement for Quality of Service (QoS)", IEEE 802.11e/D4.4, June 2003.

[IEE97] IEEE Std 802.11-1997, "Wireless LAN Medium Access Control (MAC) and Physical (PHY) Layer Specifications", IEEE 802.11, New York, USA, 1997.

[JD01] N. Jain and S. Das, "A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks", Proceedings of the 9th Int. Conference on Computer Communications and Networks, October 2001.

[JMA01] M. Jeong, H. Morikawa and T. Aoyama, "A Fair Scheduling Algorithm for Wireless Packet Networks", IEICE Trans. of Fundamental, vol. 84, no. 7, pp.1624-1635, July 2001.

[JMB01] D.B. Johnson, D.A. Maltz and J. Broch, "DSR: the Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", Ad Hoc Networking, Addison-Wesley, ch. 5, pp. 139-172, 2001.

[Joh99] M. Johnson, "HiperLAN/2 - the Broadband Radio Transmission Technology Operating in the 5GHz Frequency Band", available via the H/2 Global Forum version 1.0, http://www.hiperlan2.com, 1999.

[KL03] A. Kanjanavapastit and B. Landfeldt, "An Analysis of a Modified Point Coordination Function in IEEE 802.11", 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, pp. 1732-1736, Beijing, China, September 2003.

[KTB+04] Z.N. Kong, D.H.K. Tsang, B. Bensaou and D. Gao, "Performance Analysis of IEEE 802.11e Contention-Based Channel Access", IEEE Journal on Selected Areas in Communications, vol. 22, pp. 2095-2106, 2004.

[LAS01] A. Lindgren, A. Almquist and O. Schelen, "Evaluation of Quality of Service Schemes for IEEE 802.11 Wireless LANs", Proceedings of LCN 2001, 26th Annual IEEE Conference, pp. 348-351, Florida, USA, November, 2001.

[LBD+00] P. Lin, B. Bensaou, Q.L. Ding and K.C. Chua, "A Wireless Fair Scheduling Algorithm for Error-Prone Wireless Channels", Proceedings of the 3rd ACM International Workshop on Wireless Mobile Multimedia, August 2000.

[LBS99] S. Lu, V. Bharghavan, and R. Srikant, "Fair Scheduling in Wireless Packet Networks." IEEE/ACM Transactions on networking, vol. 7, no. 4, pp. 473-488, August 1999.

[LH00] Y.D. Lin and Y.C. Hsu, "Multihop Cellular: A New Architecture for Wireless Communications", Proceedings of IEEE INFOCOM, pp. 1273-1282, 2000.

[LYG04] H. Li, D. Yu and Y. Gao, "Spatial Synchronous TDMA in Multihop Radio Network", IEEE 59th Vehicular Technology Conference, pp. 1334-1338, Spring 2004.

[MCM+02] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz and L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service", Proceedings of the European Wireless, vol. 1, pp. 32-39, Florence, Italy, Feb. 2002.

[MG98] A. Muir and J.J. Garcia-Luna-Aceves, "A Channel Access Protocol for Multihop Wireless Networks with Multiple Channels", Proceedings of IEEE ICC '98, pp.1617-1621, June 1998.

[MKS02] K. Mizuno, M. Katayama and H. Suda, "A New QoS-Guaranteed Multichannel MAC Protocol for Multihop Wireless Networks", IEEE 55th Vehicular Technology Conference, pp. 967-971, May 2002.

[MS98] M.J. Markowski and A.S. Sethi, "Fully Distributed Wireless MAC Transmission of Real-Time Data", 4th IEEE Real-Time Technology and Application Symposium, pp.49-57, Denver, CO, June 1998.

[MWN+03] X. Ma, Y. Wu, Z. Niu and T. Saito, "Performance Analysis of the Packetized Voice Transmission with PCF in an IEEE 802.11 Infrastructure Wireless LAN", The 9th Asia-Pacific Conference on Communications, pp. 571-575, Penang, Malaysia, September 2003.

[ND00] A. Nasipuri and S.R. Das, "Multichannel CSMA with Signal Power-Based Channel Selection for Multihop Wireless Networks", Proceedings of IEEE Vehicular Technology Conference, September 2000.

[NLB02] T. Nandagopal, S. Lu and V. Bharghavan, "A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithms", Wireless Networks, vol.8, March 2002.

[NO00] K. Nakano and S. Olariu, "Randomized Initialization Protocols for Ad Hoc Networks", IEEE Transactions on Parallel and Distributed System, vol. 11, pp. 749-759, 2000.

[NSZ98] T.S.E. Ng, I. Stoica and H. Zhang, "Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors", In Proceedings of IEEE INFOCOM, 1998.

[NZD99] A. Nasipuri, J. Zhuang and S.R. Das, "A Multichannel CSMA MAC Protocol for Multihop Wireless Networks", Proceedings of IEEE Wireless Communications and Networking Conference, September 1999.

[Pee01a] J. Peetz, "HiperLAN/2 Multihop Ad Hoc Communication by Multiple-Frequency Forwarding", Proceedings of Vehicular Technology Conference, pp. 2118-2122, May 2001.

[Pee01b] J. Peetz, "A Concept for Interconnecting HiperLAN/2 Ad Hoc Subnets Operating on Different Frequency Channels", Proceedings of the 4th European Personal Mobile Communications Conference session 21, 2001.

[PG93] A. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control - The Single Node Case", ACM/IEEE Transactions on Networking, vol. 1, pp. 344-357, June 1993.

[PWS+04] R. Pabst, B.H. Walke, D.C. Schultz, P. Herhold, H. Yanikomeroglu, S. Mukherjee, H. Viswanathan, M. Lott, W. Zirwas, M. Dohler, H. Aghvami, D.D. Falconer and G.P. Fettweis, "Relay-Based Deployment Concepts for Wireless and Mobile Broadband Radio", IEEE Communication Magazine, pp. 80-89, September 2004.

[RA98] P. Ramanathan and P. Agrawal, "Adapting Packet Fair Queueing Algorithms to Wireless Networks", ACM MOBICOM'98, pp.1-9, 1998.

[RGC04] A. Raniwala, K. Gopalan, and T.C. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks", Mobile Computing and Communications Review, vol. 8, no. 2, pp. 50-65, 2004.

[SK96] J.L. Sobrinho and A.S. Krishnakumar, "Real-Time Traffic over the IEEE 802.11 Medium Access Control Layer", Bell Labs Technical Journal, pp. 172-187, 1996.

[SK99] J.L. Sobrinho and A.S. Krishnakumar, "Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks", IEEE Journal on Selected Areas in Communications, vol. 17, no.8, pp. 1353-1368, 1999.

[SLW+04] J.P Sheu, C.H Liu, S.L. Wu and Y.C. Tseng, "A Priority MAC Protocol to Support Real-Time Traffic in Ad Hoc Networks", Wireless Networks, vol.10, pp. 61-69, 2004.

[SS01] S.T. Sheu and T.F. Sheu, "DBASE: A Distributed Bandwidth Allocation/Sharing/Extension Protocol for Multimedia over IEEE 802.11 Ad Hoc Wireless LAN", Proceedings of INFOCOM'01, vol. 3, pp. 1558-1567, 2001.

[SV03] J. So and N.H. Vaidya, "A Multi-Channel MAC Protocol for Ad Hoc Wireless Networks", Technical report, Dept. of Comp. Science, Univ. of Illinois at Urbana-Chamapign, 2003.

[SV04] J. So and N.H. Vaidya, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver", Proceedings of ACM MobiHoc pp.222-233, May 2004.

[TG01] A. Tzamaloukas and J.J. Garcia-Luna-Aceves, "A Receiver-Initiated Collision-Avoidance Protocol for Multi-Channel Networks", Proceedings of IEEE INFOCOM, 2001.

[TG99] Z. Tang and J.J. Garcia-Luna-Aceves, "Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks", Proceedings of IEEE INFOCOM, 1999.

[TKN03] Technical Report TKN-03-019, "An IEEE 802.11e EDCF and CFB Simulation Model for NS-2", Telecommunication Networks Group, Berlin, Germany, 2003.

[VBG00] N.H. Vaidya, P. Bahl and S. Gupta, "Distributed Fair Scheduling in a Wireless LAN", Sixth Annual International Conference on Mobile Computing and Networking, Boston, August 2000.

[WC01] K. Wang and Y.L. Chin, "A Fair Scheduling Algorithm with Adaptive Compensation in Wireless Networks", IEEE Global Telecommunications Conference, vol.6, pp. 25-29, November 2001.

[WLT+00] S.L. Wu, C.Y. Lin, Y.C. Tseng and J.P. Sheu, "A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks", Int's Symposium on Parallel Architectures, Algorithms and Networks, 2000.

[YS93] O. Yaron and M. Sidi, "Performance and Stability of Communication Networks via Robust Exponential Bounds", IEEE/ACM Transactions on Networks, vol. 1, pp. 372-385, 1993.

[ZF02] L. Zhao and C. Fan, "M-PCF: Modified IEEE 802.11 PCF Protocol Implementing QoS", Electronic Letter, vol. 38, pp. 1611-1613, 2002.