UNIVERSITY OF CALIFORNIA, SAN DIEGO

An Efficient Multi-channel Wireless Switching System

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

in

Electrical Engineering (Electronic Circuits and Systems)

by

Jaewook Shim

Committee in charge:

Professor Kenneth Y. Yun, Chair
Professor Rene L. Cruz
Professor William Hodgkiss
Professor Ryan Kastner
Professor George Varghese

2010

The dissertation of Jaewook Shim is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

____

____

____

____

____

Chair

University of California, San Diego

2010

DEDICATION

To my parents ...

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Professor Kenneth Y. Yun for his advising and persistent encouragement during the course of my doctoral study. Completion of my dissertation would not have been possible without his guidance and valuable suggestions. From the time that I started working with him, he has continually encouraged, and guided me to reach the goal. He has also taught me how to enjoy the lingering aroma of espresso, which was one of the biggest joys during my time at UC San Diego.

I would like to thank Professor Rene L. Cruz for co-advising my research. With his insight and strong theoretical background in the field of networks and wireless communications, he guided me to have a broader perspective of solving a problem. He also convinced me that doctoral study is not only focused on solving a problem, but a lot of time is spent working to correctly identify the problem to be solved.

I would like to thank Professor Laurence B. Milstein for his helpful discussions at the beginning of my study, and Professor Miroslav Krstic for his valuable suggestions and comments in formulating the model of the scheduling system. I would also like to thank my committee members, Professor William Hodgkiss, Professor Ryan Kastner and Professor George Varghese for their for their valuable comments and the effort they made in reviewing this work.

I would like to thank former colleagues: Manish Amde for his collaboration in direct-sequence spread-spectrum (DSSS) research. He was very supportive in prototyping and testing the system presented in Chapter 2; Joel Marciano, who visited from the Philippines, for helping us implement the RF system. His expertise in RF systems was crucial in building the testbed. I really enjoyed the collaboration.

I would especially like to thank Sungho Huh for helping me to understand the cellular communications and for many valuable discussions. He inspired me many times when I was wandering around in the dark.

I would like to thank Keunhoi Kim, my senior colleague in Korea, for his guidance, support, and encouragement, as well as motivating me all the time.

My thanks also go to current and past people in San Diego for their friendship, encouragement, and discussions over the past years, in particular Sunhyoung Han, Jiho Park, Cheol-u Jang, Bongyong Song, Youngwan Kim, Dongwhan Jeon, Daeseob Lim, Somsak Kittipiyakul, Chia-Wei Chang, Sam and Shadi Dayeh. Special thanks go to the Jubaragi family and Seunghyun Lee.

Most importantly, I would like to thank my parents and my sister for their unbounded love, support and sacrifice. I could not have accomplished this without them.

Chapter 2, in part, is a reprint of the following two papers: J. Shim, M. Amde, K. Yun, and R. Cruz, *Synchronization at low SINR in Asynchronous Direct-Sequence Spread-Spectrum Communications*, the Second International Conference on Systems and Networks Communications, Cap Esterel, France, Aug. 2007 [1], and M. Amde, J. Shim, J. Marciano, K. Yun, and R. Cruz, *A Low SINR Synchronization System for Direct-Sequence Spread-Spectrum Communications: Radio Prototype, Verification Testbed, and Experimental Results*, the 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, Innsbruck, Austria, Mar. 2008 [2]. The work was carried out jointly with Manish Amde.

Chapter 3, in full, is a reprint of the following paper: J. Shim, K. Yun, and R. Cruz, *An Efficient Wireless Switching Architecture*, The Ninth Annual Wireless Telecommunication Symposium, Tampa, USA, Apr. 2010 [3]. The dissertation author is the primary author of this paper.

Chapter 4, in part, is a reprint of the following paper: J. Shim, K. Yun, and R. Cruz, *An Efficient Wireless Switching Architecture*, The Ninth Annual Wireless Telecommunication Symposium, Tampa, USA, Apr. 2010 [3]. The dissertation author is the primary author of this paper.

VITA

1992-1996	B.S., Control and Instrumentation Engineering, Korea University, Korea.
1996-1998	M.S., Electronics Engineering, Yonsei University, Korea.
2004-2010	Ph.D., Electrical Engineering, University of California, San Diego.

PUBLICATIONS

J. Shim and K. Yun, *A Multi-channel Wireless Switching Architecture*, in preparation.

J. Shim, M. Amde, J. Marciano, K. Yun, and R. Cruz, *A Low SINR Synchronization System for Direct-Sequence Spread-Spectrum Communications*, to be submitted to IEICE Transactions on Communications.

J. Shim, K. Yun, and R. Cruz, *An Efficient Wireless Switching Architecture*, Best Paper Award in WTS'10: The 9th Annual Wireless Telecommunications Symposium, Tampa, USA, Apr. 2010.

M. Amde, J. Shim, J. Marciano, K. Yun, and R. Cruz, *A Low SINR Synchronization System for Direct-Sequence Spread-Spectrum Communications: Radio Prototype, Verification Testbed, and Experimental Results*, in Tridentcom'08: The 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, Innsbruck, Austria, Mar. 2008.

J. Shim, M. Amde, K. Yun, and R. Cruz, *Synchronization at low SINR in Asynchronous Direct-Sequence Spread-Spectrum Communications*, Best Paper Award in ICSNC'07: The Second International Conference on Systems and Networks Communications, Cap Esterel, France, Aug. 2007.

D. Lim, J. Shim, T. S. Rosing, T. Javidi, *Scheduling Data Delivery in Heterogeneous Wireless Sensor Networks*, in ISM'06: The IEEE International Symposium on Multimedia, San Diego, USA, Dec. 2006.

J. Shim, G. Jeong, M. Lee and S. Ahn, *FPGA Implementation of a Scalable Shared Buffer ATM Switch*, in ICATM'98: The First International Conference on ATM, Colmar, France, Jun. 1998.

G. Jeong, J. Shim, M. Lee and S. Ahn, *A Scalable Shared Buffer ATM Switch Embedded SPRAMS*, in ISCAS'98: The IEEE International Conference on Circuits and Systems, Monterey, USA, Jun. 1998.

ABSTRACT OF THE DISSERTATION

An Efficient Multi-channel Wireless Switching System

by

Jaewook Shim

Doctor of Philosophy in Electrical Engineering (Electronic Circuits and Systems)

University of California, San Diego, 2010

Professor Kenneth Y. Yun, Chair

To cope with the insatiable demand for a higher data rate in today's single channel wireless communications, extending the spectral bandwidth to transition to a multi-channel communication is a natural course of action. This dissertation, we present a wireless switching architecture that allows a self-interference-free asynchronous packet communication in multi-channel wireless switching networks. We propose a system architecture to resolve the self-interference problem, which arises due to the proximity among RF devices in the switch and the large difference in strengths between receiving and transmitting signals.

We then present a straightforward solution of separating the frequency spectra used for receiving and transmitting signals and propose a MAC/PHY cross-layer protocol for efficiently managing the channel bandwidth for asynchronous packet-based communication. We show that, when a K-port wireless switch is used with each port providing 20 MHz of bidirectional bandwidth, the total communication bandwidth can be increased to $1.4K \times 20$ MHz, which is about $2K$ times as high as a wireless access point with 20 MHz per channel. Of course, the actual data rate depends on the modulation schemes used.

We also present a low SINR synchronization system as a physical layer solution of improving the immunity to the interference receiving from adjacent channels in a multi-channel communication environment.

Finally, we introduce a scheduling scheme with a dynamic load balancing to ensure global fairness for all users. The performance of our algorithm is compared to that of the Least-Loaded-First (LLF) user assignment policy using simulations.

Chapter 1

Introduction

Advances in multimedia and mobile device technologies over the past decade precipitated an explosive growth in bandwidth demands in wireless communications. Many research efforts in various directions have been actively explored to meet this demand: improving coding schemes to reach Shannon's channel capacity [4, 5]; consolidating the immunity to noise and interference through multiple-input-multiple-output (MIMO) diversity to enhance capacity [6, 7]; and improving efficiency of multiuser access schemes [8].

A wireless technology that has experienced considerable organic growth is the wireless local area networking (WLAN), in particular, IEEE 802.11 WiFi. The growth and widespread adoption of WiFi is largely due to the ease of deployment and the convenience it provides to the users. However, due to its inherent bandwidth limitation, in part caused by its inefficient contention-based multiple user access scheme, the WiFi communication suffers from a lower data rate than had been envisioned.

As defined in the IEEE 802.11 standards [9], there are 14 non-overlapping WiFi channels—three in 2.5 GHz ISM band, four in 5.2 GHz UNII band, four in 5.3 GHz UNII band, and four in 5.8 GHz UNII band. However, the communication over a single channel as defined by these standards is becoming less viable as the offered capacity is unable to meet today's insatiable demand for a higher data rate in wireless LANs. Thus, extending the spectral bandwidth of the communication is a natural course of action to increase the aggregate throughput for each user.

We believe that the utilization of multiple non-overlapping channels by a single basic service set (BSS)[1] for enhancing capacity while safely avoiding contention between neighboring BSS's is an efficient approach but has not been advocated up to now.

Although the total amount of frequency spectra available for the WiFi is about 400 MHz in four bands (2400 $\sim$ 2472 MHz band in North America, 2400 $\sim$ 2483.5 MHz band in EU, 5.15 $\sim$ 5.25 GHz band, 5.25 $\sim$ 5.35 GHz band, and 5.725 $\sim$ 5.825 GHz band), only a 20 MHz channel (40 MHz if the dual band or the channel bonding[2] methodology is employed) is usable at any given time, largely due to the ubiquitous deployment of access-point style architecture that provides access to a single channel to multiple users but only one user at a time through a contention process.

We believe that making multiple channels available to multiple users at the same time without contention is the best way to increase the total bandwidth as well as the per-user bandwidth. This dissertation presents a multi-channel wireless switching network, at the heart of which is a wireless switch and on the periphery of which are wireless users. The wireless switch serves as a platform for multiple concurrent communications amongst users by providing several non-overlapping wireless channels to be used as a conduit.

We first identify a serious issue that needs to be resolved for the wireless switch to function correctly: namely, the *self-interference* problem, which arises due to the proximity of RF devices in the switch and the large difference in strengths between receiving and transmitting signals. We then present a straightforward solution of separating the frequency spectra used for receiving and transmitting signals and propose a MAC/PHY cross-layer protocol for efficiently managing the channel bandwidth for the asynchronous packet-based communication. We claim that, with a K-port wireless switch with each port providing 20 MHz of bidirectional bandwidth, the total communication bandwidth can be increased to

[1] The basic service set is a "building block" of an IEEE 802.11 LAN that consists of a coordinator (typically an access point) and a set of stations associated with it.

[2] A recent standardization activity in IEEE 802.11n employed the *Channel Bonding* technology, which combines two adjacent 20 MHz channels to double the overall capacity of the network.

$1.4K \times 20$ MHz, which is about $2K$ times as high as a access point with 20 MHz per channel. Of course, the actual data rate depends on the modulation schemes used. We also present a low SINR synchronization system as a physical layer solution of improving the immunity to the interference received from adjacent channels in a multi-channel communication environment. Finally, when users are split into K groups and each group is served by a distinct port, providing a globally fair service to all users in the network is a non-trivial problem. Without a global intervention by a central scheduler, different levels of fairness can be provided to users if the aggregate load of each port is different, because loads are not evenly divisible into K ports. Therefore, an efficient way of distributing the load is necessary to provide a global fairness. In this dissertation, we introduce a dynamic load balancing scheme coupled with a round-robin based local scheduling to achieve a global fairness among all users across the ports.

Before describing the detailed architecture of the proposed wireless switch, the following topics are discussed as background and motivations: (1) the *self-interference* problem inherent in the multi-channel wireless switching network; (2)an interference resistent low SINR communication scheme: in particular, a synchronization method; (3) a set of algorithms utilized in current stat-of-the-art scheduling disciplines to set the stage for introducing the proposed scheduling algorithm.

1.1 Issues in Wireless Switching Systems

The wireless switch, as defined in this dissertation, has multiple input/output ports, each of which is associated with a wireless channel. The direction of communication (either uplink or downlink) at a port depends completely on each individual user and is not uniform across ports.

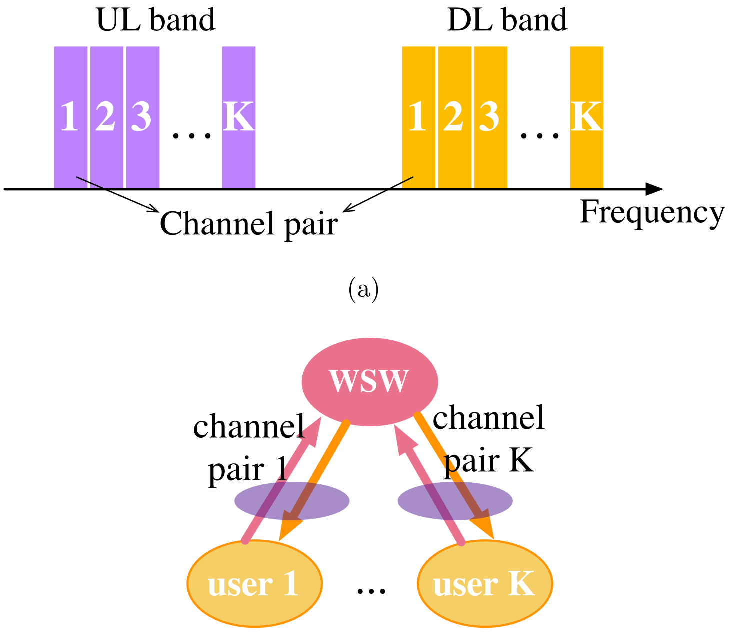
1.1.1 Self-Interference

Due to geographical proximity among ports, the signal being transmitted from a port can be detected by a neighboring port (actively in the process of receiving another signal), and its strength seen by the neighbor can be as high as that at the transmitting antenna, whereas the strength of an actual signal intended to be received typically ranges from $-50\,\mathrm{dBm}$ to $-90\,\mathrm{dBm}$, which means that the received signal strength can be $46 \sim 86\,\mathrm{dB}$ lower than the interference.[3] Since the adjacent channel rejection of channel selection filters is typically $20 \sim 30\,\mathrm{dBr}$, it is entirely possible that the interference from a neighboring transmitting port can completely wipe out the signal intended to be received.

To prevent this *self-interference*, the transmission and reception of signals at the wireless switch should be separated in time or in frequency. One possible way of achieving separation in time is the time division duplex (TDD) scheme, where the transmission and reception of signals are done in fixed but different time slots: i.e., when one port transmits so do all other ports and vice versa. This, however, requires a system-wide synchronization over the whole network as well as the identical uplink/downlink (UL/DL) traffic load ratio amongst ports. The separation in frequency can be accomplished in two ways. The first way is to ensure that a channel allocated to a port has significant separations in frequency from all others, which is clearly a huge waste of frequency spectra. The second way is the frequency division duplex (FDD), which is commonly used in cellular voice communication.

Figure 1.1 depicts a channel allocation in the FDD communication. There are two frequency bands, the uplink (UL) band and the downlink (DL) band, the spacing between which is sufficiently large that two bands can be safely separated with a duplex filter. A UL channel and a DL channel are combined to form a channel pair, and each user is assigned to a channel pair.

[3] According to the IEEE 802.11a standard [9], the maximum strength of the transmitted signal in the 5.2 GHz band is $16\,\mathrm{dBm}$ when the antenna gain is $6\,\mathrm{dBi}$. If two adjacent ports are separated by 5.7 cm (the wavelength of a 5.2 GHz microwave), the path loss is $\approx 20\,\mathrm{dBm}$ [10]. Hence the strength of the transmitting signal detected by the adjacent port is $\approx -4\,\mathrm{dBm}$.



(a)

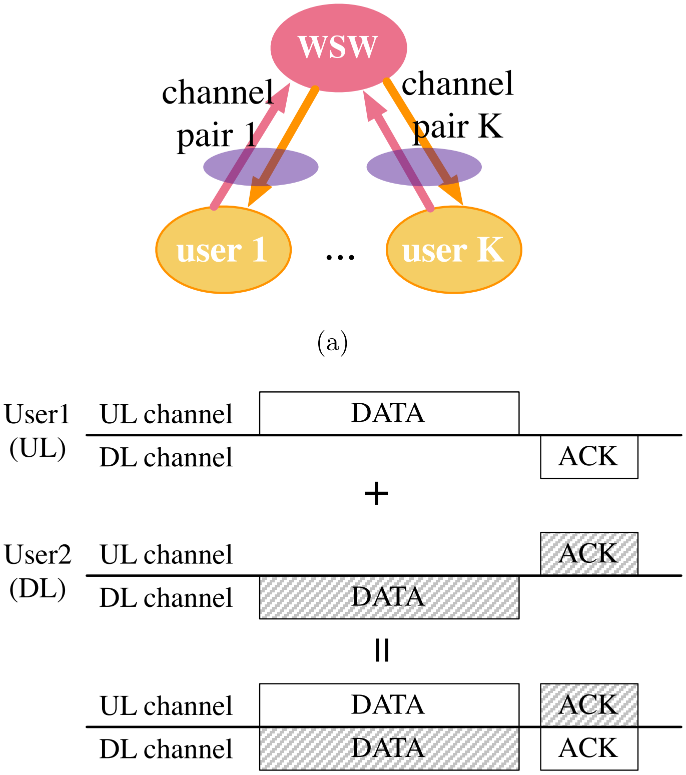
(b)

Figure 1.1: Uplink and downlink frequency bands.

1.1.2 Full-Duplex vs. Half-Duplex

The scheme shown in Figure 1.1 allows *full-duplex* bidirectional communications amongst K users. However, the asymmetric nature of data traffic leads to a waste of channel resources. To avoid this underutilization of channels, the *half-duplex* FDD (H-FDD) can be used [11–13], where UL and DL channels are independently allocated to distinct users so that a higher level of utilization can be achieved. Conventional H-FDD schemes are implemented in the TDM based framework, which requires time-slotted or fixed-duration uplink and downlink packets.

In packet-based asynchronous communications (e.g., WiFi), the ability to immediately acknowledge receiving a packet has a direct implication in the performance of the system. That is, a delay in acknowledgment leads to a delay in the subsequent transmission and increases the idle period of channel, thus reducing the throughput and efficiency. Therefore, to be able to efficiently share a channel pair with another user (henceforth referred to as the *peer* user), it is required that nodes be time-aligned and packet durations be fixed as shown in Figure 1.2. Or, conflicts can also be avoided by deferring a new transmission until the previous transmission is acknowledged as shown in Figure 1.3(a), which clearly is a waste of channel bandwidth.

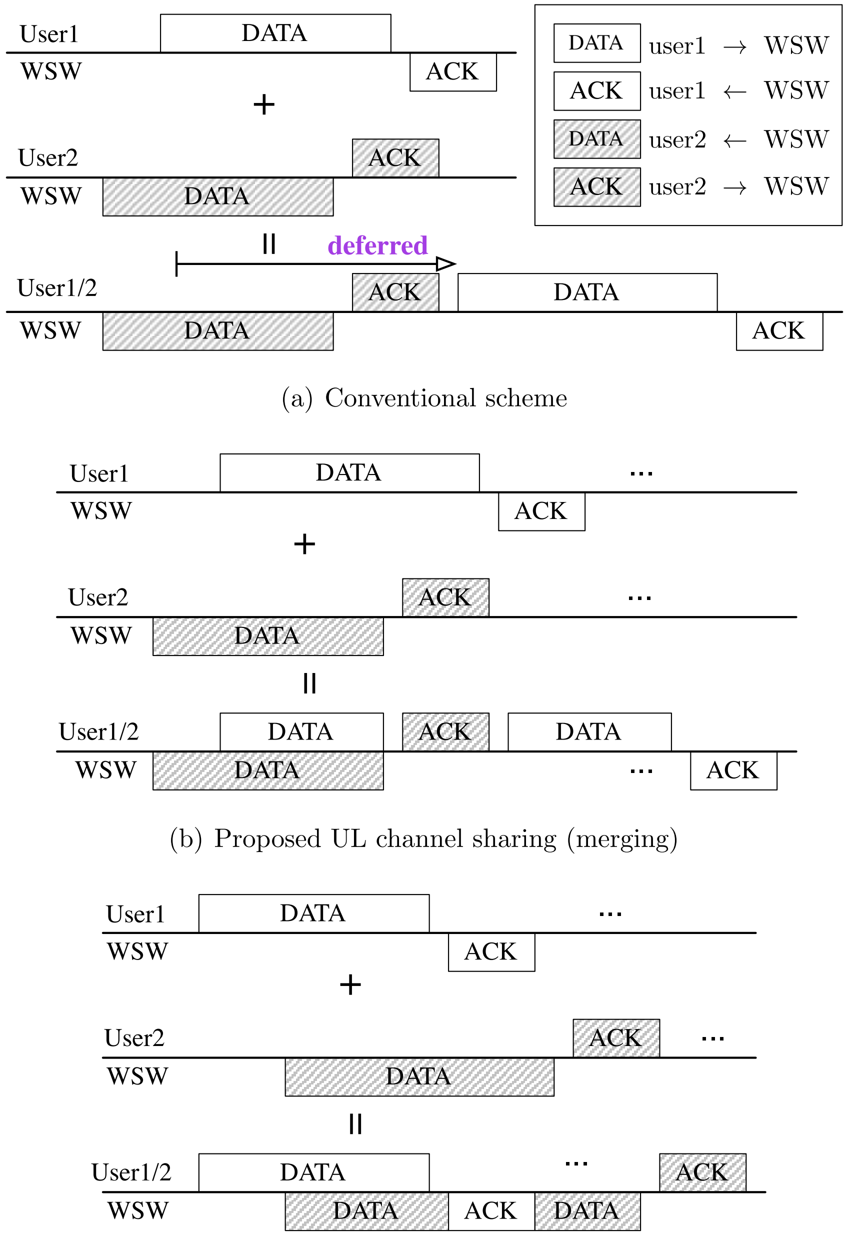


(a)

(b)

Figure 1.2: Channel pair sharing when users and wireless switch (WSW) are time-aligned.

On the other hand, the protocol proposed in this dissertation promotes



(a) Conventional scheme

(b) Proposed UL channel sharing (merging)

(c) Proposed DL channel sharing (splitting)

Figure 1.3: Channel pair sharing when UL and DL traffic are asymmetric and are not aligned.

an efficient channel sharing even when the traffic loads on the channel pair are asymmetric or not aligned in time. Figure 1.3(b) shows a UL channel shared by a UL data packet (from `User1`) and a DL acknowledgement (from `User2`), and Figure 1.3(c) depicts a DL channel shared by a DL data packet (to `User2`) and a UL acknowledgement (to `User1`). A detailed description will be provided in Chapter 3.

1.2 Low SINR Communications: Synchronization

In wireless communications, signal attenuation over a long communication distance leads to low signal strength at the receiver. Moreover, interference from other transmissions (either in the same channel or in the adjacent channel) reduces the signal to interference and noise ratio (SINR) at the receiver. Therefore, the ability to operate at low SINR is crucial for reliability of communications. The low SINR operation can be achieved by using direct-sequence spread spectrum (DSSS) modulation [14,15], where higher SINR gain can be obtained by increasing *spreading ratio* at the cost of data rate.

As high bit error rate (BER) in data can be rectified by coding and error correction techniques, synchronization is the most indispensable in successful reception of packets. Therefore, low SINR operations are essentially gated by synchronization. Synchronization is aligning received packets with those that originally transmitted in terms of time and frequency. In a DSSS system, the synchronization process consists of the following operations: *Code acquisition*, *Code tracking*, *Carrier recovery*, *Multipath search and combining*, and *Frame synchronization*. Among these, Code acquisition, Code tracking, Multipath search and combining, and Frame synchronization constitute time synchronization, which essentially locates the time boundary of packets for the correct reception of them. Carrier recovery, however, deals with the frequency mis-match between transmitter and receiver.

1.2.1 Code Acquisition

Code acquisition aligns the spreading code in the incoming signal with the locally generated replica at the receiver for accurate de-spreading. It finds the symbol boundary of a received packet by examining repeating symbols in preamble, each of which is spread by the spreading code.

1.2.2 Code Tracking

This operation performs fine synchronization and maintains spreading code generated at the receiver in sync with the incoming signal. By doing code tracking, the possible timing drift due to the sampling clock mis-match between the transmitter and the receiver can be corrected.

1.2.3 Carrier Recovery

Carrier recovery compensates for the difference of carrier frequency and phase between the transmitter and the receiver. The phase error, which causes a fixed phase rotation of the received symbols, can also be corrected using equalization. The carrier frequency error, which is normally due to inaccurate crystal oscillators and/or Doppler shift, causes phase rotation of received signal over time. Carrier recovery is essential for coherent demodulation.

1.2.4 Multipath Search and Combining

In multipath fading channel environment, collecting multiple duplicates of transmitted signal arrived through multiple paths at the receiver would enhance receiver performance. In DSSS communications, the multipath signals with different delays can be individually aligned through multiple code acquisition units to be combined using RAKE receiver [15,16].

1.2.5 Frame Synchronization

The goal of time synchronization is correctly locating when the payload data of a received packet starts. As code acquisition finds the repeating symbol boundary, the time synchronization is completed by frame synchronization. Frame synchronization finds the start of the payload of the received packet by locating *a priori* known pattern, named start frame delimiter (SFD), which is inserted between the preamble and payload data.

1.3 Scheduling and Load Balancing

Scheduling is a decision-making process to allocate resources to tasks and optimizing a given objective. It can be applied to many applications where multiple tasks share limited resources, ranging from sharing manufacturing facilities among multiple products to allocating computing jobs to share one or more servers (CPUs). The main objective of scheduling is efficiency of the system which can be achieved by maximizing the usage of resources (e.g., manufacturing facilities or servers), as well as fairness by allocating fair opportunity/resources to each task. This dissertation considers a scheduling system for a multi-port wireless switch, where the set of users (tasks) are sharing multiple ports (servers), which is coordinated by a scheduler. More precisely, a scheduler repeatedly chooses a user and assigns each user to a wireless port over time so that (1) ports are utilized as efficiently as possible and (2) users are serviced in a fair manner.

1.3.1 Fairness Index

Since scarce resources are typically shared by users, scheduling may not fully satisfy users' demands. As an objective of scheduling, fairness represents the relative satisfaction of users. That is, it represents that the resources are shared by the set of users evenly or in proportion to users' demands when weighted fairness is considered. There has been some research activity to define *fairness index*, the yardstick to characterize the fairness: Jain's fairness index (JFI) [17] measures a system-wide fairness. However, it does not measure individual user's fairness. The fairness index proposed by Shreedhar et al. [18] deals with individual user's fairness, hence it is adopted and used in this dissertation to evaluate the performance of the proposed algorithm (see Chapter 4):

$$\mathtt{FI}_i = \frac{served_i}{\sum_{j \in U} served_j} \Bigg/ \frac{w_i}{\sum_{j \in U} w_j} \tag{1.1}$$

where U denotes the set of user indices, $\mathtt{FI}_i$ the fairness index of user $i \in U$, $served_i$ the amount of resources allocated to user i, and w_i the weight of user i.

1.3.2 Generalized Processor Sharing (GPS) and Fair Queueing

Generalized Processor Sharing (GPS) is a service discipline devised to provide fair service in the context of queueing system [19]. It assumes an idealized fluid system where packets (tasks) from all links (users) are infinitely divisible so that those can be served simultaneously by the server. Though it is referred to as a reference model for an ideal scheduling discipline, GPS is not practically feasible due to its fluid assumption. Demers et al. [20] proposed packet-based approximation of GPS, named weighted fair queueing (WFQ). The authors calculate the departure time of a packet at the idealized system (in their work, bit-by-bit round-robin). Packets are then sorted in the increasing order of departure time to be serviced. Parekh et al. [21, 22] applied and analyzed this discipline in the context of integrated services networks and named it packet-based version of GPS (PGPS). According to the authors, packets serviced in WFQ fall behind GPS by at most one packet size. However, it is possible that packets of some user(s) are serviced much earlier in a WFQ than in a GPS, which leads to bursty transfer of packets (see examples in [23, 24]). This so-called *worst-case fairness* problem is addressed by worst-case fair weighted fair queueing (WF2Q) [23], which considers packet start time of GPS system as well when it chooses a packet to be serviced.

These WFQ based disciplines have an inherent complexity problem due to the virtual time (departure/start time of packet in the ideal system) calculation and sorting mechanism. For the purpose of simplifying the computing complexity, self-clocked fair queueing (SCFQ) [25] introduces an approximation algorithm for virtual time computations but it causes another packet bursting problem (refer to the example in [24]) due to inaccuracy of operation. As an attempt to reduce the complexity of WF2Q system, WF2Q+ [26] introduces a simpler but more accurate virtual time calculation scheme but it still needs a sorting scheme as WF2Q does. In spite of those efforts, since these WFQ-based and SCFQ schemes use a sorting mechanism, high computing complexity is inevitable so that it requires high cost for implementation and poses an restriction to be applied to high data rate systems.

1.3.3 Credit-based Round-Robin Scheduling

Round-Robin scheduling is a simple and easy-to-implement scheduling discipline, where each user is selected to be served one after another in a fixed and repeated order. It gives basically the same opportunity to every user hence it guarantees fair share of bandwidth when the size of packets is fixed. But when there is a *greedy* user or packet sizes are highly varying, it is prone to benefit users of larger packets. This problem can be solved by using a credit-based round-robin scheduling, named deficit round-robin (DRR) scheduling [18, 27] or its variants [28–33]. In DRR scheduling, arrived packets of each user are stored in its own queue. Each user is considered a round-robin manner. At each turn, a user, say user i, is assigned with a quantum, $\mathtt{Q}_i$, which is accumulated in its *deficit counter*, $\mathtt{DC}_i$. The size of each user's quantum is computed *a priori* based on the amount of the service to be rendered to the corresponding user (i.e., weight). The $\mathtt{DC}_i$ is a sort of credit that can be used for user i to get served. That is, if $\mathtt{DC}_i$ is greater than or equal to the size of the packet at the head of queue, then the packet is served and $\mathtt{DC}_i$ is reduced by the packet size. If $\mathtt{DC}_i$ is less than the packet size, user i is assigned with a quantum, $\mathtt{Q}_i$ but its packet is not served. This process is repeated until $\mathtt{DC}_i$ becomes greater than the packet size. Note that users may get different levels of service depending on its quantum size, which allows DRR to provide weighted fair scheduling.

One notable variant of DRR scheduling is Surplus Round-Robin (SRR) scheduling [28], where user i's packet is served when $\mathtt{DC}_i$ is positive. Once a packet is served, $\mathtt{DC}_i$ is reduced by the serviced packet size. If $\mathtt{DC}_i$ goes below zero, user

i's packet is not served until $\mathtt{DC}_i$ becomes positive. This scheme is useful for the scheduler for wireless networks, where the scheduler does not have the knowledge of user backlogs (so it cannot compare $\mathtt{DC}_i$ with user i's packet size to be served).

1.3.4 Multi-server System and Load Balancing

Fairness in a single server system can be achieved with the aforementioned scheduling algorithms. On the other hand, in a multi-server system, where the set of users is served by multiple servers, in order to be able to achieve fairness, it is desirable that each task (packet) is evenly divided and served by multiple servers simultaneously so that each server is identically loaded at all times. Unfortunately, this is not possible unless fluid assumption is made. Adopting the virtual time mechanism used in the WFQ, it can be considered that users' packets are sorted by the increasing order of *virtual* departure time in the ideal (fluid) system and served by the servers one by one.

However, the wireless switching system proposed in this dissertation, which can be viewed as a multi-server system serving wireless packets for a set of users, has an additional constraint: the service provided to a user is based upon a port-to-user wireless connection and an instantaneous change of the connection (or even a rapid one) is inherently impossible. Therefore, for the purpose of scheduling, it can be assumed that the set of users is divided into multiple sub-groups and each group is served by a server. This can guarantee fairness within each sub-group (local fairness) as the single server system does. However, if there is an imbalance of load amongst servers, fairness amongst the total set of users (global fairness) cannot be achieved. Accordingly, a load balancing scheme is necessary for global fairness.

There is a rich set of literature on multi-server scheduling for wired systems [34–38], scheduling for single server wireless systems [39, 40], scheduling for multi-server wireless systems with the complete knowledge of backlogs [37], and scheduling for multiple Access Points (APs) in WiFi networks [41]. However, to the best of our knowledge, none addresses the fairness and load balancing problem for a packet based multi-channel wireless switch system proposed in this dissertation.

1.4 Contributions of the Dissertation

This dissertation presents a multi-channel wireless switching system. It describes essential parts of wireless switching systems, including high performance synchronization, a cross-layer protocol for channel sharing (CS-CLP), media access control (MAC) protocol, and a scheduling algorithm to ensure global fairness amongst wireless users served by the system. The main contributions are described as follows:

- *Wireless Switching System*: This dissertation introduces an architecture for wireless switching system, which provides higher per-user bandwidth than is currently possible. The higher bandwidth is achieved by utilizing multiple channels. As in the network switch for wired networks, multiple users are served at the same time in this system.

- *High Performance Synchronization System Design and Implementation*: This dissertation proposes a new *design* of low SINR synchronization system for direct-sequence spread-spectrum (DSSS) communications. The system is designed and implemented on a radio prototyping platform. Then, the implemented system is verified and the lab experiments are performed. The experimental results show the reliable operation of the system at low SINR.

- *Cross-Layer Protocol*: This dissertation presents a collision-free channel sharing protocol named cross-layer protocol for channel sharing (CS-CLP). The CS-CLP manages *sharing* of the channel pair in frequency division duplex (FDD) based packet communication so that the utilization of channels is maximized. With this new protocol, it is possible for two users of opposite links (uplink and downlink) communicate with the wireless switch without conflict.

- *Fair Scheduling Algorithm*: This dissertation presents a new fair scheduling algorithm for the multi-channel wireless switching system, providing globally fair service to a set of users without wasting channel resources. The SRR based scheduling is used for local scheduling of each port and serves locally associated users in a fair manner. A dynamic load balancing scheme with compensation is used for global fairness. The load imbalance is dynamically adjusted by the dynamic load balancing, and the service discrepancy caused by the past load imbalance is adjusted by the compensation scheme.

- *Media Access Control (MAC) Layer Protocol*: This dissertation presents essential functionalities of the MAC layer protocol for the proposed wireless switching system, which are required for user admission, communication setup and scheduling contract.

1.5 Overview of the Dissertation

The rest of the dissertation is organized as follows:

Chapter 2 presents a low SINR synchronization system. A Direct-Sequence Spread-Spectrum (DSSS) modulation based synchronization system, consisting of code acquisition, frame synchronization and code tracking, is designed, and its performance is verified using simulation. Next, a prototype system is implemented on a prototyping platform and experiments are performed to measure the synchronization performance of the system.

Chapter 3 describes the physical-layer-independent cross-layer protocol for channel sharing (CS-CLP), which is a key part of the multi-channel wireless switching system to maximize the efficiency of channel pairs. The channel efficiencies for three transmission modes are analytically presented along with simulation results for random combination of the three modes. For the purpose of comparison with IEEE 802.11a, the orthogonal frequency division multiplexing (OFDM) modulation scheme is assumed for the analysis and simulation.

Chapter 4 presents a scheduling algorithm optimized for the K-port wireless switching systems. The aim of the scheduling algorithm is not only to provide local fairness among users served by each port but also to provide global fairness across ports. To this end, it performs two steps of scheduling. First, each port provides a Surplus Round-Robin (SRR) based local scheduling. The quantum size of each user, which corresponds to the fair share of the load imposed on the system by the user, is globally calculated as if the all users were served by a single port with K times the capacity. Then, by tracking the amount of service rendered to each user and dynamically balancing loads among ports, global fairness is achieved. Finally, the performance of the algorithm is compared to that of the Least-Loaded-First (LLF) user assignment policy using simulations.

Chapter 5 presents a summary of this dissertation and a direction for future research.

Chapter 2

Physical Layer: Direct-Sequence Spread-Spectrum based Synchronization System

In wireless switching systems, RF nodes communicating on adjacent channel are typically located in a geographically close space: on the wireless switch side, ports are packed in one small box; one the user side, some user nodes may be randomly located nearby. This proximity can cause serious interference problems. The *self-interference* among switch ports (or users close to each other) can be avoided by using the FDD scheme discussed in Chapter 1. However, the interference from received signals on adjacent channels still exists. As this interference cannot be completely filtered out, the ability to operate at a low SINR environment can improve the probability of successful reception of packets. The key component of successful low SINR operations is *synchronization*, which is detailed in this chapter.

This chapter represents a Direct-Sequence Spread-Spectrum (DSSS) modulation based synchronization system. The synchronization system is designed and simulations are performed. The system is implemented on a radio prototype and the performance are measured on a hardware testbed capable of emulating wireless channels.

This chapter organized as follows. Section 2.1 describes the synchronization system. Section 2.2 describes synchronization algorithms and architectures. Its system level performance are presented in Section 2.3. Section 2.4.1 and Section 2.4.2 describe the radio prototype and the hardware verification testbed respectively. Section 2.4.3 presents the experimental setup and results. Finally, the conclusions for this chapter are presented in Section 2.5.

2.1 System Description

2.1.1 System Overview

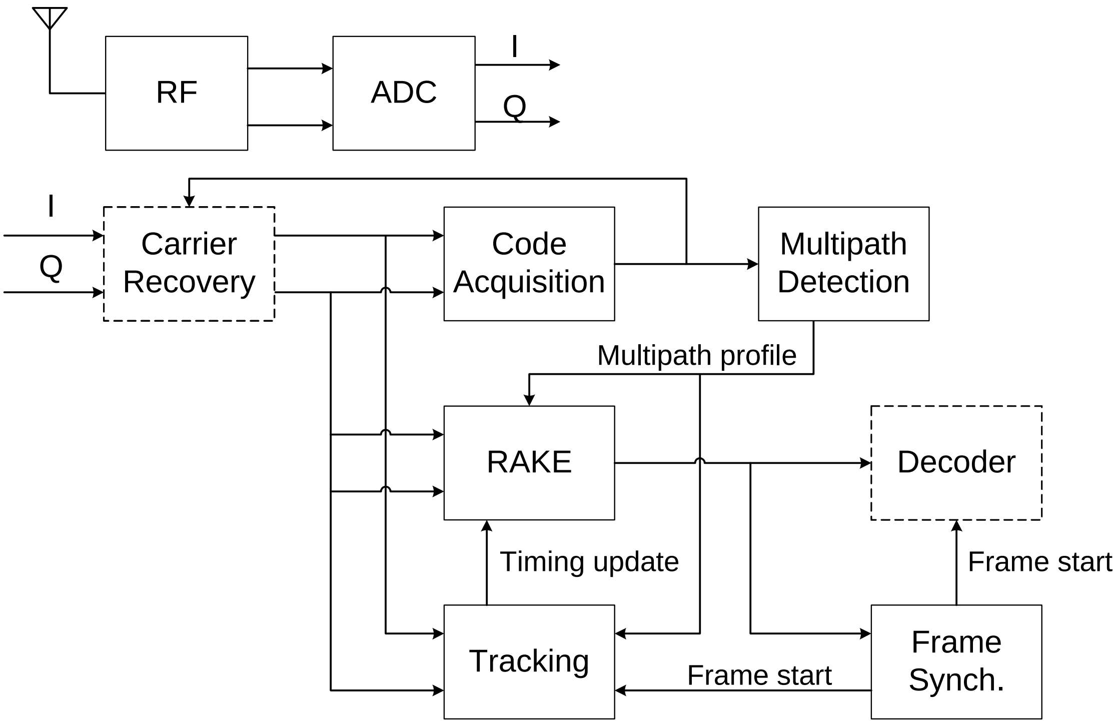


Figure 2.1: Synchronization System (SYNC).

The Synchronization System (SYNC) is depicted in Figure 2.1. The received RF signal is down-converted to baseband and sampled by an analog-to-digital converter (ADC). Multiple points of the baseband signal is sampled during a *chip* period to be used for more fine time synchronization.

In a DSSS system [16, 42], the transmitted signal is a narrowband signal modulated by a wideband spreading waveform. That is, each data bit is multiplied by a spreading sequence which consists of multiple number of *chips* for spreading the signal over the entire allocated bandwidth. The number of chips used for transmitting a bit is known as the spreading factor. This chapter describes the design of baseband synchronization system (SYNC). It consists of various synchronization blocks, which are typically implemented in digital hardware: The *code acquisition* block aligns the incoming signal with a locally generated spreading sequence in the receiver for an accurate de-spreading operation. The *multipath search* block resolves all incoming multipaths[1] using a parallel search algorithm. The multipath information is passed to the RAKE [16, 43] combining block for improving the SINR for decoding. The *frame synchronization* block finds the start of payload data in the incoming packet even during the presence of individual symbol errors. The *code tracking* block maintains the symbol-level synchronization at the receiver by compensating for the timing drift during the course of the reception of the entire packet.

The *carrier recovery* block is crucial for coherent detection. However, a non-coherent detection strategy is used for the simplicity of the system. The effect of a non-coherent detection strategy [44] on the performance of the SYNC is negligible when the phase offset is random and the frequency offset is small. Although it is not implemented, the carrier recovery techniques described in [16, 45] could be easily adapted to the proposed synchronization scheme, if a coherent operation were desired.

2.1.2 Spatial Range and Receiver Sensitivity

Receiver sensitivity, that is, the lowest strength of the signal that a receiver can detect accurately, is a practical figure of merit for comparison of receivers. We illustrate the strategy used for extending the range of networks using the IEEE 802.11b [9] wireless LAN standard as a reference. Commercially available IEEE 802.11b receivers promise a receiver sensitivity around $-90\,\mathrm{dBm}$. Suppose that an

[1] The term *multipaths* refers to signals arriving at the receiver via multiple paths.

ad-hoc network requires nodes with a receiver sensitivity of $-105\,\mathrm{dBm}$ to extend the range of the network by $15\,\mathrm{dB}$, or in terms of distance an increase of free-space coverage by 5.6 times [16, 46]. The spreading gain required to compensate for the low signal strength can be calculated using the equation below.

$$\begin{aligned}\text{Spreading Gain (dB)} &= \text{Minimum SINR (dB)}\\ &+ \text{Thermal Noise (dBm)}\\ &- \text{Receiver Sensitivity (dBm)}\end{aligned}$$

The total thermal noise power calculated for a 22 MHz RF bandwidth (i.e., a chipping rate of 11 M chips / sec), similar to the IEEE 802.11b specification, is approximately $-90.58\,\mathrm{dBm}$ [46][2]. The minimum SINR for a reliable operation is assumed to be $3\,\mathrm{dB}$ for *post de-spreading*, which will be justified in detail in Section 2.3. Hence, the spreading gain required for correct operation is $17.42\,\mathrm{dB}$ which corresponds to a spreading factor of 55.26. Since the lengths of spreading codes with good autocorrelation and crosscorrelation properties are $2^n - 1$ [47], a spreading code of length 63 in is chosen for the design. It is important to note that the receiver sensitivity can be modified using different spreading gains for different applications.

2.1.3 Packet Format

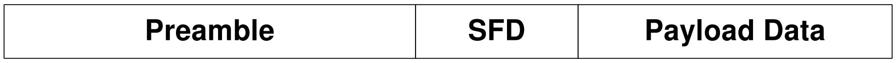


Figure 2.2: Frame format.

Figure 2.2 shows the frame format for the proposed system. A frame comprises a preamble, a start frame delimiter (SFD), and payload data. The preamble consists of M symbols (bits) of *logic one*, each of which is spread by the same pseudorandom (PN) sequence which has an impulse-like autocorrelation property [47]. The code acquisition is performed by aligning a locally generated

[2] The noise figure for the RF front-end is assumed to be $10\,\mathrm{dB}$.

PN sequence with the PN sequence in the incoming signal during the reception of a preamble. After the code acquisition is completed, a parallel search for multipaths is performed subsequently during the preamble reception.

Frame level synchronization is attained by the SFD detection after a multipath search. One PN sequence is used as a spreading sequence for the preamble because it consists of just one repeating symbol. However, the SFD and payload data are a combination of two symbols based on binary orthogonal signaling. In order to perform a non-coherent detection, two orthogonal *Kasami* sequences are chosen for spreading — one for each symbol. Kasami sequences have slightly sub-optimal autocorrelation properties as compared to the PN sequence, but they have very good crosscorrelation properties needed for a non-coherent detection of the payload [48]. Finally, the code tracking maintains the synchronization for the de-spreading operation over the reception of the entire payload.

2.2 Synchronization Algorithms and Architectures

2.2.1 Code Acquisition

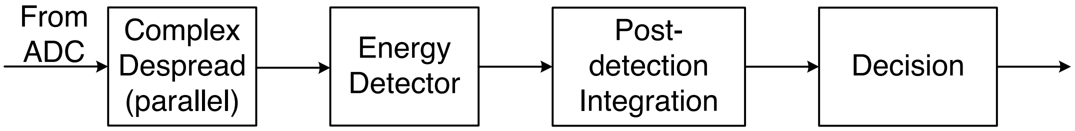


Figure 2.3: Code acquisition.

The basic code acquisition operation is shown in Figure 2.3. The code acquisition block consists of a complex despreader, an energy detector, a post-detection integration (PDI) block, and a threshold-based decision block. Both In-phase (I) and Quadrature-phase (Q) components of incoming digitized baseband signal are despread via the complex despreader. The code acquisition is performed in a non-coherent manner using an energy detection architecture [44]. For a reliable system-level synchronization, the probability of correctly acquiring a signal (probability of detection, P_D) should be high and the probability of acquisition in the absence of a signal (probability of false alarm, P_F) should be low [49]. This can be achieved by accumulating energy of multiple bits using a technique called the *post-detection integration* (PDI) [44, 50]. Note that the more bits are accumulated the more prominent contrast between P_F and P_D can be obtained. The last step of the acquisition is a threshold-based decision. The decision block check if the PDI output exceeds the threshold calculated from the variance of the noise [44] to determine the correct code phase.

2.2.2 Multipath Search

Previous studies on multipath search can be classified into two categories: *serial search* based [51–54] and *parallel search* based [55, 56]. Because of timing constraints such as mean acquisition time, most serial search based methods employed a conventional (non-PDI) acquisition scheme [51–53] or a PDI-based scheme with a small L [54]. Moreover, many parallel search based methods, in order to save the system resources [55, 56], do not employ the PDI. Instead, many studies used a *dual-dwell* scheme with a 'verification step' to improve the false alarm probability [51–53, 55]. The proposed architecture is based on a *single-dwell* PDI-based parallel acquisition [50].

After the code acquisition, all the incoming paths are searched using the same PDI-based architecture used for code acquisition. This scheme examines all code phases, hence all multipaths in parallel. After one path is discovered by code acquisition, the remaining paths are searched within half a bit period of the first path discovered. Note that the chipping rate of the simulated system is 11 MHz, the spreading factor is 63 and hence the symbol rate is 174.6 Kbps. This ensures that paths separated by up to approximately 860 m can be detected.

In the simulations, detection of strong signals occurs within the first few bits of the incoming signal whereas the lower strength signals are detected after more than 20 bits of the preamble have slid into the PDI-based energy detector. Ideally, one should start the RAKE combining only after all the resolvable paths have been detected, which requires waiting for some number of bits (L') after discovering the first path for accurately estimating the multipath. According to the simulations, $L' = 32$ works the best for searching over the entire multipath profile. Thus, the preamble length M is chosen to be $L + L'$ bits (64 bits).

2.2.3 Frame Synchronization

Frame synchronization is achieved by detecting the start frame delimiter (SFD) field in an incoming packet after the code acquisition and the multipath detection. As described in Section 2.1.3, the SFD consists of two symbols, each with a different spreading code. This format is based on the M-ary orthogonal signaling scheme.

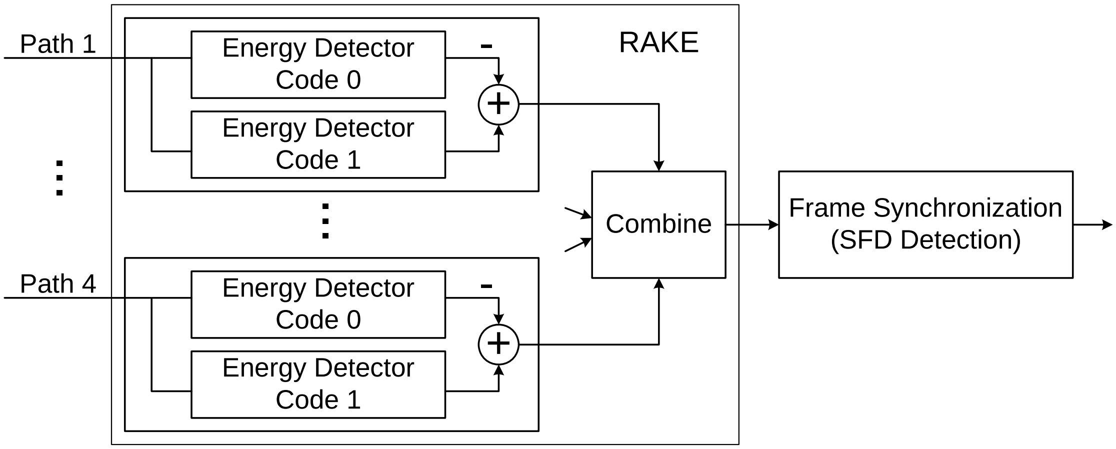


Figure 2.4: Non-coherent RAKE.

Most DSSS systems have the ability to resolve multipaths in frequency-selective fading environments and hence employ a RAKE [16, 42] architecture for combining the energy of multiple paths to improve the effective SINR for decoding operation. This dissertation uses a non-coherent RAKE as shown in Figure 2.4 for combining the energy of up to 4 multipath signals. Each finger consists of two distinct energy detectors for the two Kasami codes used to represent symbols 0 and 1. At a high SINR, this RAKE combining method results in a performance very close to the *maximal ratio combining* [16]. Even at a low SINR, it is observed from simulations that the proposed scheme works much better than the *equal gain combining* [16] where all the RAKE branches are scaled down to have the same power.

Current wireless LAN standards usually employ a short SFD field: for example, the IEEE 802.11b standard [9] uses a 16-bit long SFD field. Moreover, the *exact* sequence is searched using a 'shift and compare' technique after decoding each bit. An error in one of the SFD bits typically leads to the loss of an entire packet. Since a single bit error is highly probable at low SINR, this dissertation adopts a *correlation*-based scheme for frame synchronization in which a packet is assumed to be frame-synchronized after the correlator output crosses a certain threshold.

A 31-bit PN sequence is chosen for the SFD in the packet format. The SFD is detected using a sliding correlator window which is of the same length as the SFD field. The longer length SFD field improves the performance at low SINR by leveraging the impulse-like autocorrelation property of the PN sequence.

Evaluating the performance of this SFD detection scheme theoretically is intractable; instead the performance of the algorithm is analyzed using simulations as shown in Figures 2.7 and 2.8. The P_{SFD} curve denotes the probability of SFD detection after the code acquisition and multipath detection are performed successfully. Figure 2.7 shows the performance of SFD detection with varying SINR in an AWGN environment. Figure 2.8 depicts the SFD detection performance in multipath Rayleigh fading conditions described in Section 2.3.1.

The SFD detection algorithm has a probability of detection close to 1 for SINR greater than or equal to 3 dB under both AWGN and Rayleigh fading conditions. For SINR under 3 dB, the performance of the SFD detection block deteriorates in the AWGN environment but somewhat less under the Rayleigh fading condition.

2.2.4 Code Tracking

Clock drift due to inaccurate crystals used at the transmitter and the receiver leads to a misalignment of the spreading code in the incoming signal and the locally generated replica at the receiver. This results in a deterioration in de-spreading operation and thus a degradation in the receiver performance over the course of the payload data reception. The code tracking, which is done after the frame synchronization, maintains the accuracy of the de-spreading operation by compensating for the clock drift during the reception of the payload data.

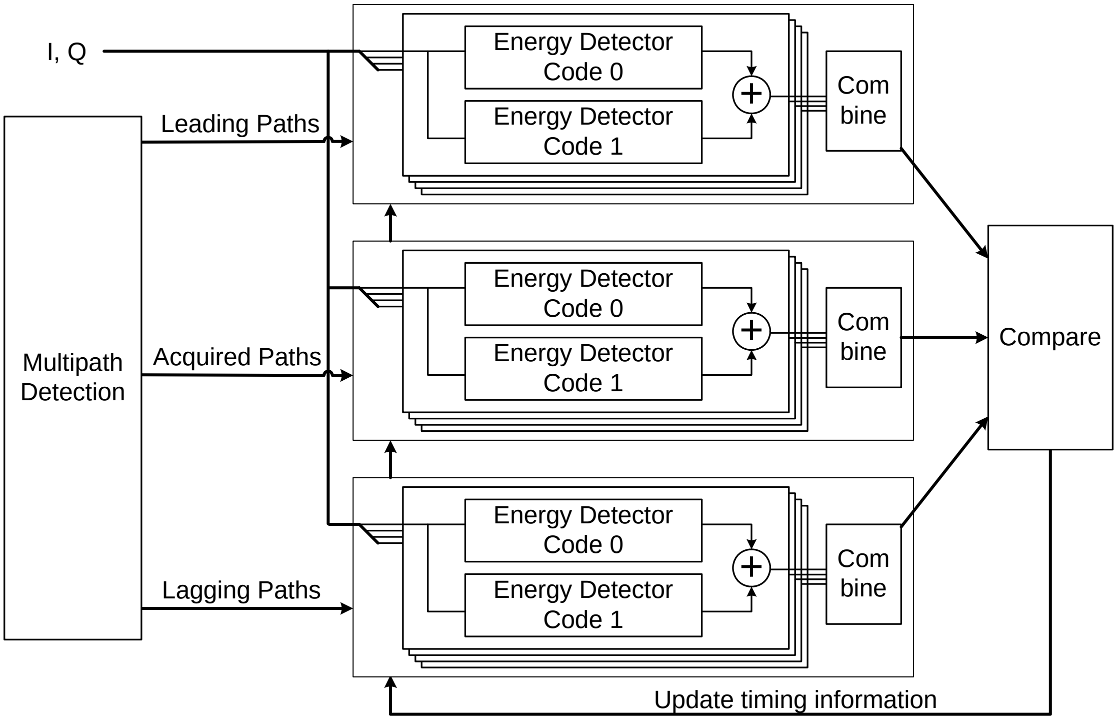


Figure 2.5: Code tracking architecture.

Previous studies on code tracking are based on the delay-locked loop (DLL) and its variations [57–60]. Both the advanced and the delayed phases of the locally generated spreading sequence are correlated with the incoming signal to keep track of the timing drift. However, these schemes are designed to track a single code over the entire packet. We have not come across tracking schemes based upon non-coherent detection of multiple spreading codes used in the orthogonal signaling.

The code tracking architecture shown in Figure 2.5 comprises 3 branches. The center branch calculates the combined energy detector outputs using a current estimate of time. The upper and lower branches perform the same operation using the leading and lagging estimates of time. The multipath detection block provides an initial information about the location (in time) of acquired paths to the three branches. The signal strengths of the peak are compared over a period of time to check whether the maximum value shifts from the center branch to the upper or lower branch.

The following parameters are used to simulate this algorithm. A 25 parts per million (ppm) clock drift and an oversampling coefficient of 8 samples per chip are assumed. For a 25 ppm drift, the clocks in the transmitter and receiver drift by 1 sample period every 79 symbol periods (for a spreading gain of 63). The upper and lower (leading and lagging) branches are spaced one sample (one-eighth of a chip) apart from the center.

Table 2.1 shows probabilities of losing a lock for various SINR under AWGN. The tracking loop is said to lose a lock when its estimated time for de-spreading drifts by more than one chip period compared to the ideal time for de-spreading over the reception of the entire payload data. It is observed that the code tracking algorithm performs accurately for SINR greater than or equal to 2 dB. The locking capability deteriorates at a lower SINR; the tracking loop does not lock at all at 0 dB.

2.3 System Level Performance

2.3.1 Channel Model

This dissertation uses a frequency-selective (multipath) Rayleigh fading model for the simulations. Suppose there are L_p path components. Let τ_l denote the relative delay of the l^{th} path from the first (reference) path, i.e., $\tau_1 = 0$. The delay profile is given by $\{\tau_l\}_{l=1}^{L_P}$. The small-scale fading is modeled by the Rayleigh distribution and the path loss by an exponentially decaying multipath intensity

Table 2.1: Code tracking performance.

SINR (dB)	probability of losing lock
5.00	0
4.00	0
3.00	0
2.00	0
1.00	0.074713
0.50	0.549020
0.00	1.000000
-1.00	1.000000

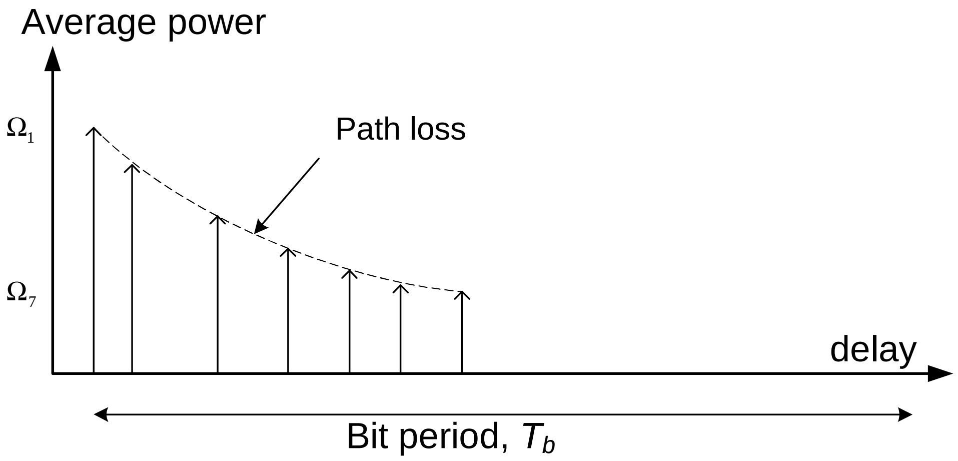


Figure 2.6: Multipath channel model.

profile (MIP) [61]:

$$Rl\Omega_l = \Omega_1 \; e^{-\tau_l/\tau_{max}}, \quad l = 1, 2, 3, ..., L_p$$

where Ω_l is the average power of the l^{th} path and τ_{max} represents the maximum delay spread.

The SINR under the multipath Rayleigh fading conditions is calculated from the signal power Ω_1, the strongest average multipath signal. Note that it does not denote the SINR based on the combined strength of all the paths in the multipath profile.

The specific delay profile used for simulation, as shown in Figure 2.6, is:

$$Rl\{\tau_l\}_{l=1}^{L_P} \;=\; \{0,\; 3.88,\; 10.63,\; 15.75,\; 19.75,\; 22.13,\; 26.5\}$$

where the unit is a chip period T_c. Therefore, for a spreading factor of 63, the maximum delay spread is 0.42 times a bit period. In other words, it is assumed that the maximum delay spread is less than a half bit period.

2.3.2 Simulation Results

In Section 2.2, the design and performance of individual synchronization blocks is described. Each of those blocks contributes to the ability of a receiver to synchronize and decode packets at a low SINR. This section presents the performance of the entire SYNC.

The code acquisition is a key component of the synchronization algorithm as it enables accurate de-spreading, which is crucial to synchronizing packets at a low SINR. This operation is followed by the multipath detection which resolves the incoming multipath signals for RAKE combining. The RAKE combiner output is used for frame synchronization which informs the decoder to start decoding the payload. The code tracking is performed during the payload data reception to maintain synchronization over the entire packet.

Figure 2.7 shows the system-level performance of the entire SYNC (tracking excluded) under AWGN. The $P_{CODEACQ}$ curve describes the probability of detection for the code acquisition algorithm for various post de-spreading SINR. The detection threshold is designed for a false alarm probability (P_F) of 10^{-6}. For a fixed P_F, it is shown that $P_{CODEACQ}$ increases with increasing SINR and is very close to 1 at 3 dB. The P_{SFD} curve, as described in Section 2.2.3, shows the probability of frame synchronization given successful code acquisition. The P_{SYNC} curve, which is obtained by multiplying the $P_{CODEACQ}$ and P_{SFD} curves, is the probability of frame synchronization for the entire SYNC. It is observed that P_{SYNC} is nearly 1 at 3 dB SINR.

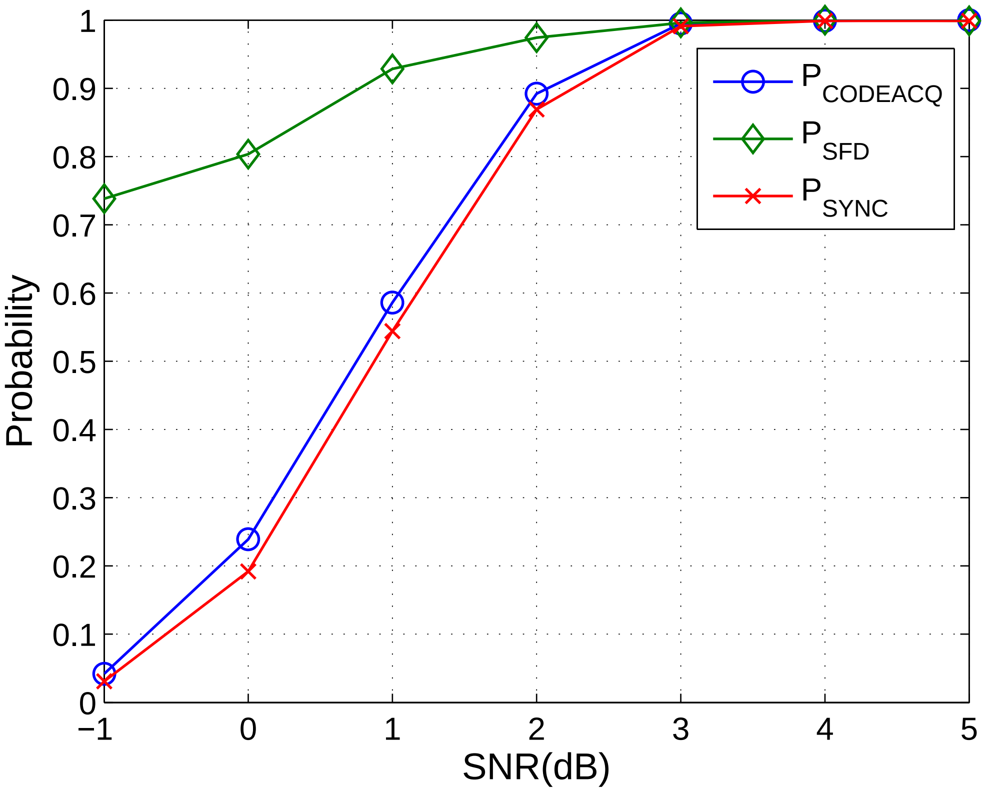


Figure 2.7: System-level synchronization under AWGN.

Figure 2.8 shows the system level performance under the multipath Rayleigh fading environment. The performance of the SYNC is better at low (≤ 3 dB) SINR, compared to Figure 2.7, due to a multipath diversity gain. It is again observed that the probability of synchronization is higher than 0.95 for SINR of 3 dB which

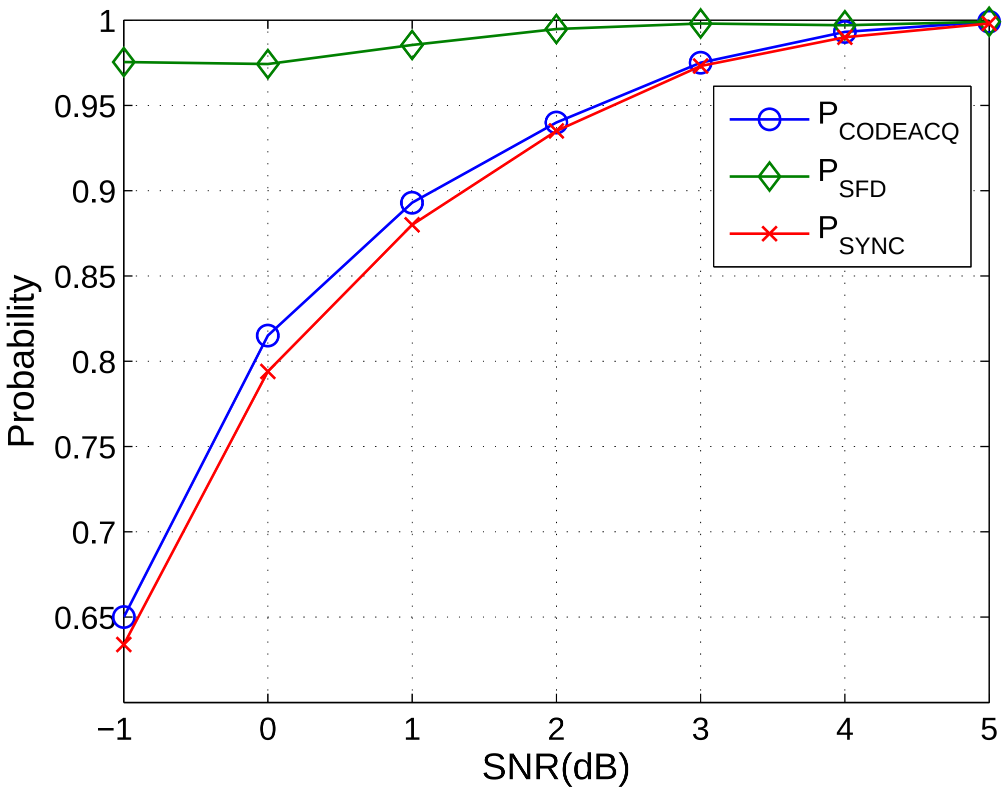


Figure 2.8: System-level synchronization under multipath Rayleigh fading.

is an acceptable level for most networks. It is observed that at 4 dB SINR the synchronization probability is close to 1 which implies that all the packets arriving at 4 dB post de-spreading SINR (−14 dB input SINR) will be received correctly.

The code tracking algorithm is responsible for maintaining the synchronization during the payload reception. Section 2.2.4 presents that the code tracking block works well under AWGN at SINR $\geq$ 2 dB. Hence, it can be claimed that the proposed design can *synchronize* packets when the input signal strength is very low (SINR of 3 dB). More specifically, the sensitivity of the receiver design is −105 dBm, as described in Section 2.1.2.

2.4 Prototyping and Experiments

2.4.1 Radio Prototype

The SYNC is implemented on a radio prototyping platform to demonstrate the feasibility of its implementation.

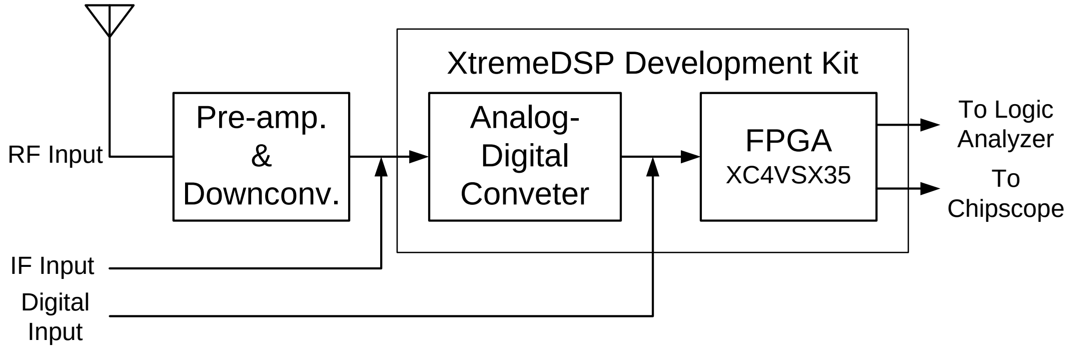


Figure 2.9: Block diagram of the radio prototyping platform.

Figure 2.9 shows a block diagram of the radio prototyping platform. The RF receiver front-end is implemented using a programmable RF equipment from National Instrument (NI). The digital baseband component of the synchronization system is implemented on a Xilinx XtremeDSP Devolpment Kit, which is equipped with a Xilnix Virtex-4 (XC4VSX35) FPGA and 14-bit DAC with a maximum sampling rate of 105 MSPS.

A RF signal is down-converted to a intermediate frequency (IF) of 15 MHz and IF signal is first digitized at 80 MSPS by the ADC and then digitally down-converted to baseband before being processed by the baseband section of the synchronization system.

The following parameters were used for designing the VLSI architecture of the digital baseband section:

- Chipping rate = 10 Mchips/sec
- Sampling rate = 80 Msamples/sec
- Oversampling ratio (N_s) = 8
- Spreading ratio (N_c) = 63
- Symbol rate = 158.73 Kbits/sec
- Length of PDI operation = 32 symbol periods
- Number of RAKE fingers (N_{FINGER}) = 4

The area utilized on the FPGA was proportional to the number of symbols used in the post-detection integration operation for code acquisition and the spreading factor. A brief summary of the FPGA implementation is shown in Table 2.4.1. For better understanding of these parameters, one can refer to the Virtex-4 documentation [62].

Table 2.2: FPGA implementation details.

Logic Distribution	Used	Avail.	Utilized
Number of 4-Input LUTs	25298	30720	82%
Number of BRAMs	157	192	81%
Number of DSP48s	18	192	9%

2.4.2 Verification Testbed

The implemented synchronization system is verified using the hardware test setups available on the testbed. The verification consists of two steps:

- *Digital Baseband Test*: The baseband test vectors used for simulations are fed to the SYNC implemented on the FPGA as digital input for verification of the digital hardware. Note that the signals used for the simulations and digital hardware verification are *exactly identical* including the noise characteristic.

- *RF Test*: The baseband test vectors used for simulations (without the noise) are up-converted to RF and fed as input to the prototyping platform. This test verifies the operation of the design under more realistic environment.

Digital Baseband Test Setup

Figure 2.10 shows a block diagram of the digital baseband test setup. The baseband test vectors are replayed as a digital stimulus to test the digital subsystem of the SYNC. This is achieved by using a combination of the Agilent Baseband Studio N5110B [63] and N5102A [64] modules. An logic analyzer is used for monitoring and calibrating the performance of the digital logic inside the FPGA.

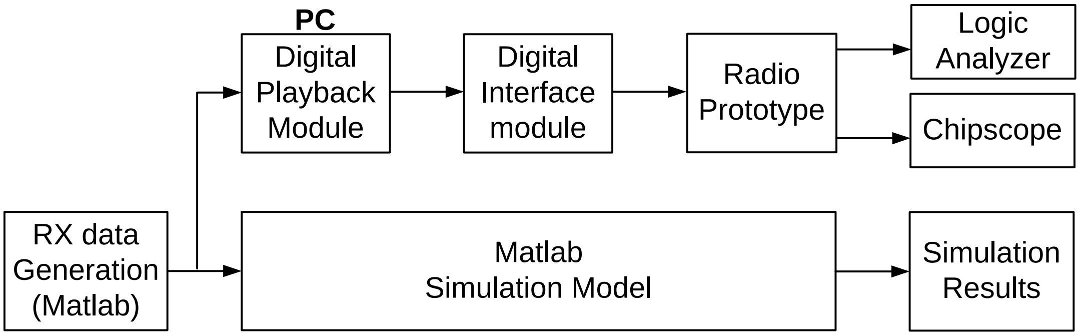


Figure 2.10: Digital Baseband Test Setup.

RF Test Setup

Figure 2.11 shows a block diagram of the RF test setup. In this setup, the baseband test vectors are up-converted to 2.4 GHz by an Agilent ESG 4438C [65] signal generator which serves as a RF transmitter. The transmission bandwidth is set to 20 MHz, for a chipping rate of 10 Mchips/sec, similar to that used in WLANs [9]. The transmitter output is connected to the radio prototype input via a tunable attenuator. This setup emulates a single line-of-sight link between the transmitter and receiver. The received signal level can be adjusted using the tunable attenuator. The logic analyzer in this setup helps in real-time monitoring of the synchronization performance. It is also used for displaying test results reported by Chipscope module embedded in the digital hardware design.

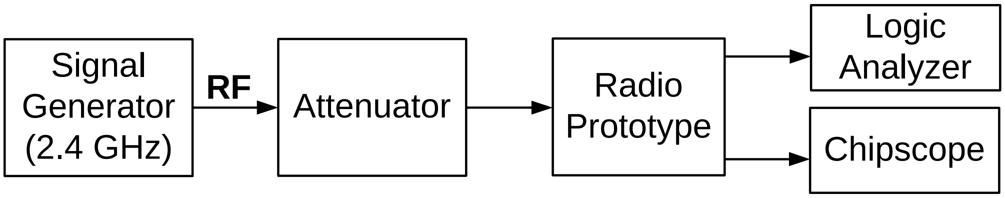


Figure 2.11: RF Test Setup.

Hardware Test Results

Figure 2.12 shows the probability of correct synchronization at various SINR under additive white Gaussian noise (AWGN) environment. As mentioned earlier in Section 2.2.1, the SINR is calculated after the de-spreading operation which includes the spreading gain of approximately 18 dB (for a spreading factor of 63). Figure 2.12 shows that the digital verification results closely match the simulation results and thus verify the accuracy of the digital implementation. It is also observed that the controlled RF verification tests, using the setup described in Section 2.4.2, accurately match the simulation and digital verification results.

Figure 2.13 shows that digital verification results for the multipath Rayleigh fading channel described in sub-section 2.3.1 accurately match the simulation results. RF verification tests, similar to the experiments conducted for the AWGN channel, could not be conducted due to the difficulty in emulating a specific multipath environment in the laboratory. From Figures 2.12 and 2.13, we concludes that the implementation of the SYNC on the prototyping testbed faithfully translates the SYNC design into real hardware.

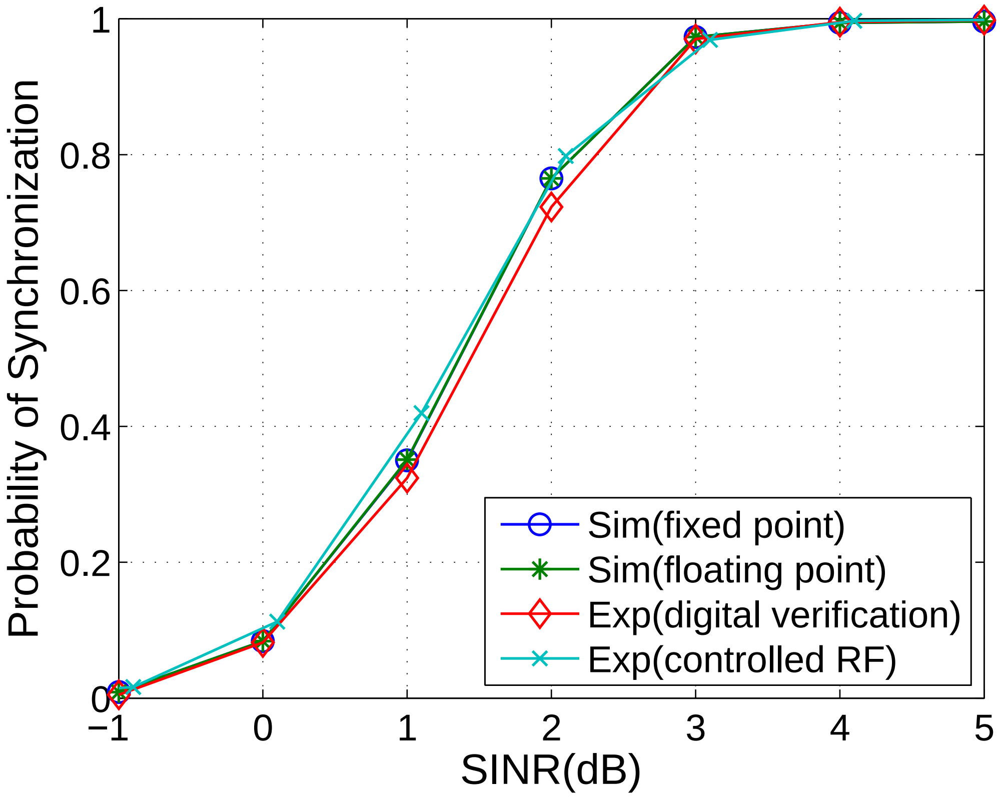


Figure 2.12: Simulation and verification results under AWGN.

2.4.3 Experimental Results

The ability of the SYNC to receive low SINR signals has already been established via simulations and hardware tests in Sections 2.1 and 2.4.2, respectively. This sections describes experiment conducted to calculate the sensitivity of the SYNC prototype. Sensitivity is the minimum signal level required for reli-

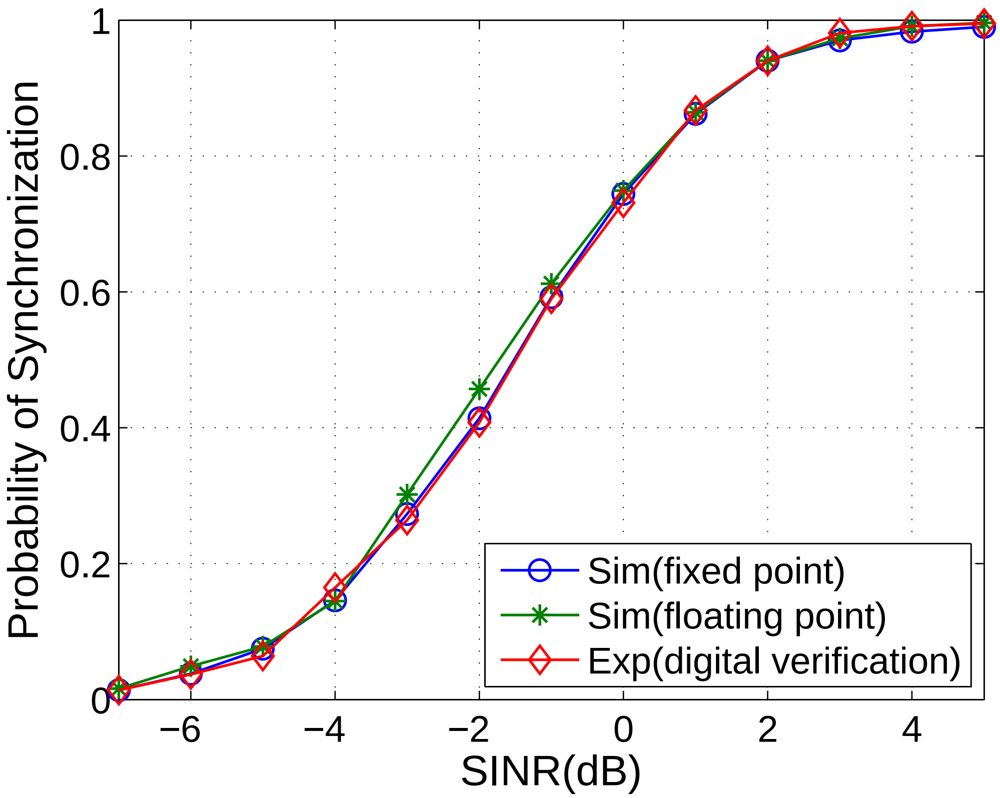


Figure 2.13: Simulation and verification results for multipath environment.

able operation of the SYNC. A 95% synchronization rate is used as the yardstick for reliable operation of the SYNC. The RF test setup described in Section 2.4.2 is used for conducting the sensitivity experiment. It is equivalent to an unfaded AWGN channel without multipath. The received signal level is controlled using an attenuator between the Tx RF output and the Rx RF input.

Receiver sensitivity is used as a figure of merit for calibrating the performance of wireless receivers. For example, according to the IEEE 802.11 standard [9], the receiver sensitivity is the minimum input signal level for which the frame error rate shall be less than 8% at a payload length of 1024 bytes encoded using 2 Mbps Differential Quadrature Phase Shift Keying (DQPSK) modulation. The sensitivity for commercial IEEE 802.11a/b/g wireless receivers varies from vendor to vendor and also depends on the data rates and modulation used. −94 dBm is the lowest sensitivity level we have encountered in literature [66].

From Figure 2.14, it is observed that the prototype is able to synchronize reliably at low signal levels at the RF input of the receiver. For example, it is able to synchronize 99.37% of the received packets when the signal level is −101.7 dBm. The sensitivity of the SYNC is observed to be at least −102 dBm since it correctly synchronizes more than 95% of the received packets at that signal level.

2.5 Conclusion

This chapter describes an implementation of a synchronization system for low SINR operation in asynchronous packet-based DSSS communications on a reconfigurable radio prototyping platform. The prototype was verified on a hardware testbed, and lab experiments were carried out to calibrate its performance. Experimental results show that the prototype has a lower receiver sensitivity compared to commercial IEEE 802.11 receivers, which is useful for receiver operations in interference-rich multi-channel wireless systems described in the next two chapters.

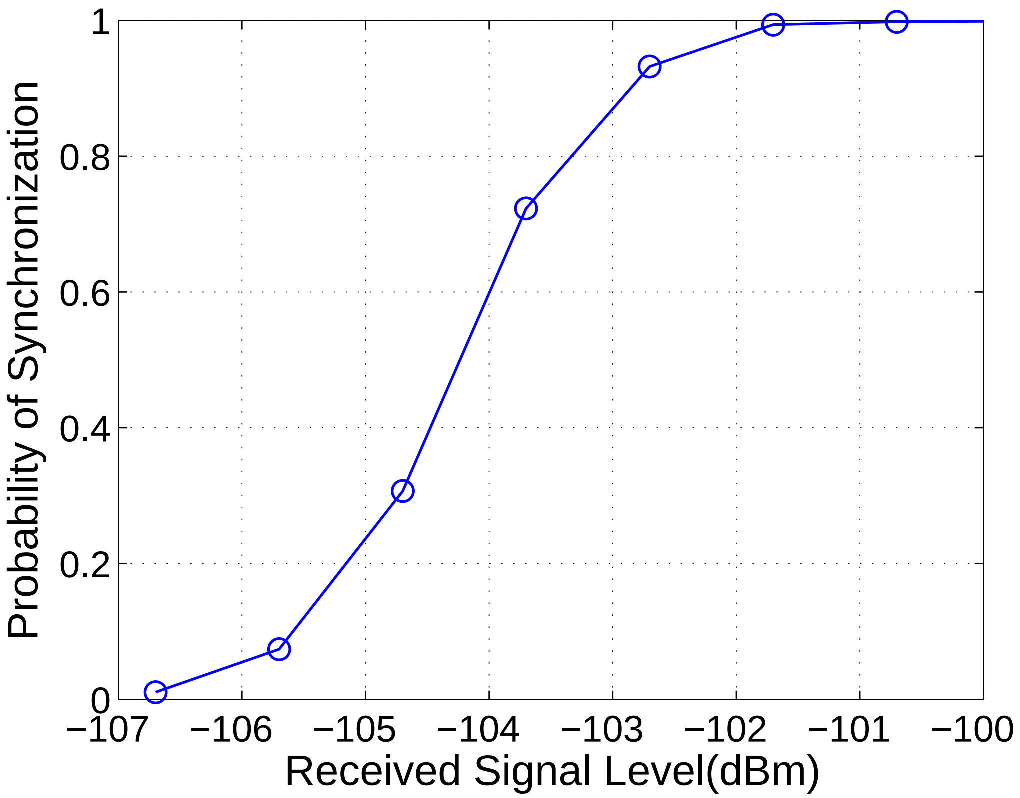


Figure 2.14: Sensitivity experiment results.

Acknowledgment

The work in this chapter, in part, is a reprint of the following two papers: J. Shim, M. Amde, K. Yun, and R. Cruz, *Synchronization at low SINR in Asynchronous Direct-Sequence Spread-Spectrum Communications*, the Second International Conference on Systems and Networks Communications, Cap Esterel, France, Aug. 2007 [1], and M. Amde, J. Shim, J. Marciano, K. Yun, and R. Cruz, *A Low SINR Synchronization System for Direct-Sequence Spread-Spectrum Communications: Radio Prototype, Verification Testbed, and Experimental Results*, the 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, Innsbruck, Austria, Mar. 2008 [2]. The work was carried out jointly with Manish Amde.

Chapter 3

Cross-Layer Protocol

This chapter presents the cross-layer protocol for channel sharing (CS-CLP). The main objective of CS-CLP is to maximize the utilization of channel pairs in packet-based asynchronous communications using the frequency division duplex (FDD) scheme. The CS-CLP resides between the Physical layer and the MAC layer and enables the channel pair sharing by managing the delivery of acknowledgement frames at the physical layer. Though it is independent of the modulation scheme used, for the purpose of comparison with IEEE 802.11a, the orthogonal frequency division multiplexing (OFDM) [67] modulation scheme is assumed.

This chapter is organized as follows. Section 3.1 describes the system model. The operation of CS-CLP is described in Section 3.2 and its channel efficiency calculated in Section 3.2.2. Section 3.3 presents simulation results of channel efficiency for random transmission of packets. Finally, we draw conclusions in Section 3.4.

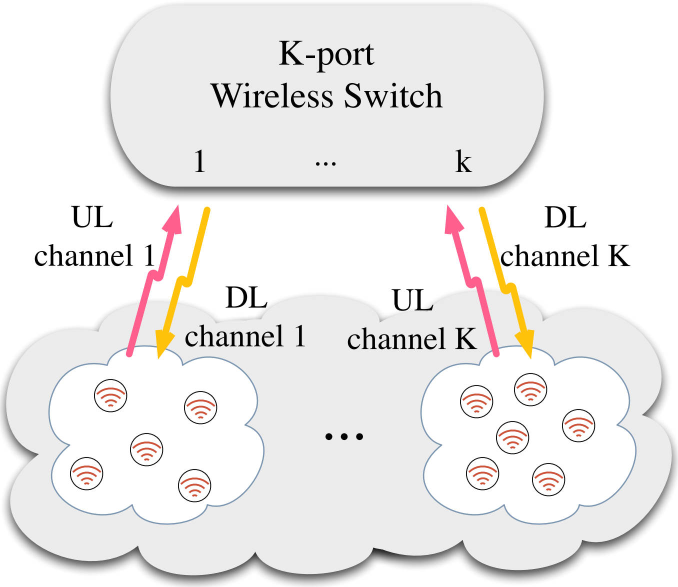


Figure 3.1: Wireless switching network.

3.1 System Model

3.1.1 Wireless Switching Network

Figure 3.1 represents a wireless switching network, which consists of a K-port wireless switch and N users. $U = \{1, 2, \ldots, N\}$ is the set of user indices and $P = \{1, 2, \ldots, K\}$ the set of port indices. Port k is associated with channel pair k. N_k users are dynamically assigned to port k, according to a scheduling criterion, where $\sum_k N_k = N$. The set of indices of users associated with port k is U_k and $|U_k| = N_k$. At each port, one user is scheduled for UL transmission and another for DL. If all users are backlogged, K users transmit to the switch and K users receive from it at any instant. Note that the set of UL users and that of DL users are not necessarily disjointed.

3.1.2 Protocol Stack

The proposed protocol deals with the delivery of acknowledgment frames at the *physical* layer, although it is typically handled in the MAC layer, to avoid

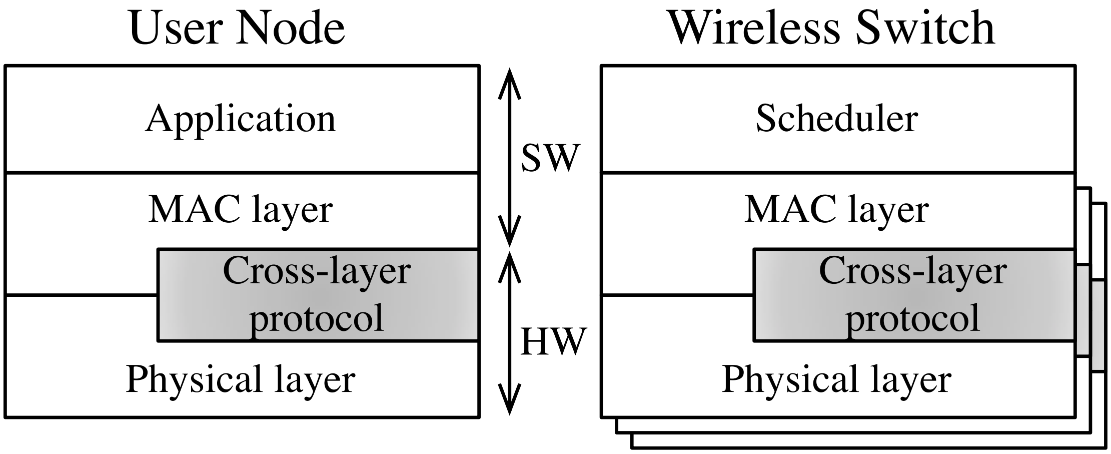
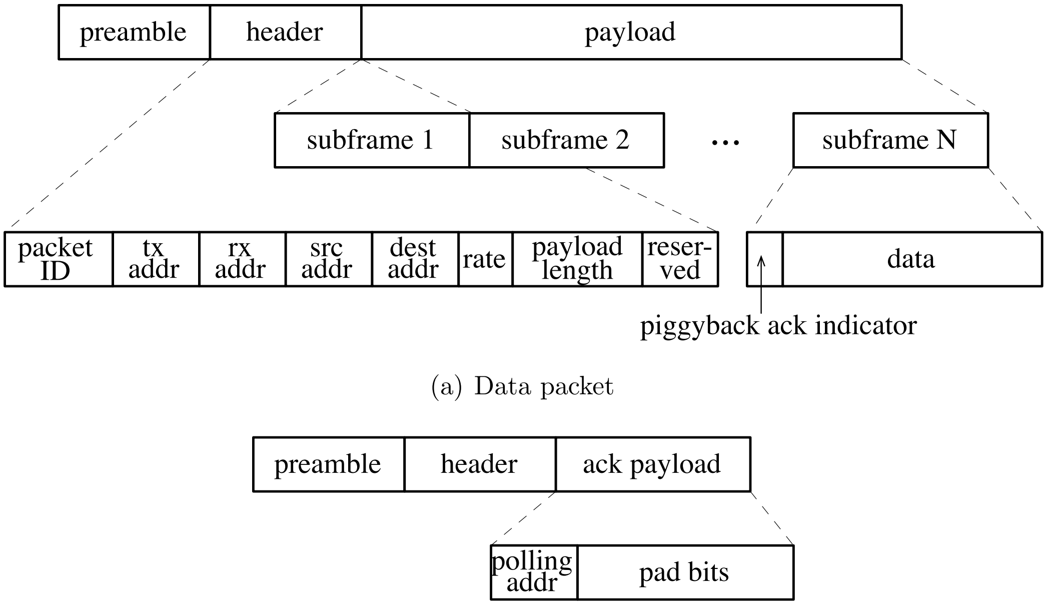


Figure 3.2: Protocol stack.

conflict between the users sharing a channel pair and to maximize the utilization of a channel. Functionally, it is part of the MAC layer but is handled by the physical layer for speed. Since it does not require a modification of the modulation scheme, the proposed protocol resides between the physical and MAC layers as shown in Figure 3.2; hence it is called cross-layer protocol for channel sharing (CS-CLP). This protocol is implemented in hardware as most physical layer protocols are.

3.1.3 Packet Format

Figure 3.3 depicts the physical layer packet format for the proposed system. A packet consists of a preamble, a header, and payload data. The header contains a packet ID, a transmitter address (`tx addr`), a receiver address (`rx addr`), a source address (`src addr`), a destination address (`dest addr`), a rate, a payload length, and reserved bits. If, for example, a user on port 1 transmits a packet to a user on port 2, then, on the UL side, `src addr` = `tx addr` = 1, `rx addr` = 0, and `dest addr` = 2; on the DL side, `src addr` = 1, `tx addr` = 0, and `rx addr` = `dest addr` = 2. Note that the address of the switch is always 0. The payload data is split into subframes and transmitted one after another. The payload of an acknowledgment packet consists of a polling address and pad bits (although the acknowledgment itself does not require any information). The polling address field



(a) Data packet

(b) Acknowledgement packet

Figure 3.3: Packet format.

(`polling addr`) holds the address of a user scheduled for the next transmission. The pad bits are added if a fixed payload size is required (e.g., to fit into an OFDM symbol).

3.2 Cross-Layer Protocol for Channel Sharing: CS-CLP

The complete implementation of the aforementioned wireless switching network requires two functionalities: CS-CLP and a scheduling algorithm. The CS-CLP defines additional constraints for the local operation between a wireless switch port and users associated with it. As shown in Figure 3.2, a scheduler resides in the upper layer of MAC and delivers the scheduling information to wireless ports. Various scheduling policies can be implemented based on QoS requirements and target applications [18, 27–32, 34–37, 39–41]. An example implementation will be discussed in Chapter 4.

3.2.1 Protocol Description

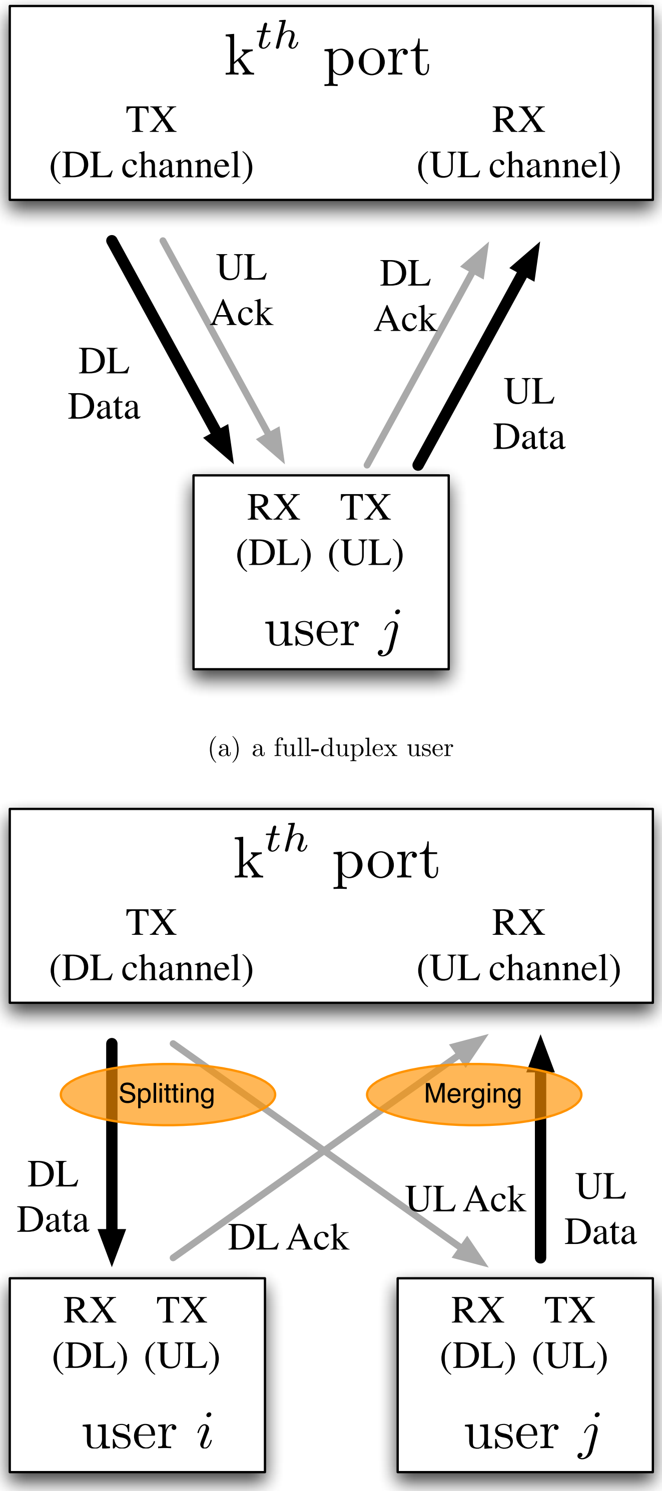
The CS-CLP manages the channel pair sharing to maximize the utilization of a channel pair in packet-based asynchronous communications using the FDD scheme. We assume that a group of users is allocated to a switch port (based on a scheduling policy). If the traffic to a user is bidirectional, the user operates both the uplink (UL) and the downlink (DL), as shown in Figure 3.4(a) and runs in the full-duplex mode. On the other hand, if two users are scheduled on opposite links of a channel pair, as shown in Figure 3.4(b), both users run in the half-duplex mode. It should be noted that the ports on the wireless switch always operate in the full-duplex mode.

In Figure 3.4(a), both the switch port and the user operate in the full-duplex mode. The acknowledgment to a DL data packet can either be piggybacked on a UL data packet or sent as an independent packet if there is no UL data. In Figure 3.4(b), both links are shared by two users. On the DL channel, one user receives data packets and the other user acknowledgment packets whereas, on the UL channel, one user transmits data packets and the other user acknowledgment packets.

The CS-CLP supports three transmission mode to share a channel pair effectively between two links without conflict: *merging* transmission, *splitting* transmission, and *deferred* transmission.

Merging transmission

In the scenario shown in Figure 3.5(a), user i, upon receiving a packet on the DL channel, must send an acknowledgment on the UL channel immediately. User j must therefore relinquish control of the channel temporarily to user i in order to avoid conflict. In order to accomplish this, user j must continuously listen to the DL channel and track the completion times of DL packets. Before user j can send out a subframe, it must estimate the transmission time of the acknowledgment packet from user i and ensure that there is no overlap of the subframe transmission and the acknowledgment period (see Figure 3.6). If there is a possibility of conflict,



(a) a full-duplex user

(b) two half-duplex users sharing a channel pair

Figure 3.4: Full-duplex and half-duplex users.

user j must delay transmission of the subframe until the channel is clear. As the packets from two different transmitters are "merged" before being delivered to the wireless switch, we refer to this as a *merging* transmission. Figure 3.5(a) depicts an example of the merging transmission.

The *merging interval*, $T_{merging}$, is defined as the period during which the user transmitting data, user j in our previous example, must relinquish control of the channel to the user transmitting an acknowledgment packet on the same channel, user i in our previous example. $T_{merging}$ is a multiple of subframe duration, T_{sfrm}, and must be greater than T_{ack_period}. Thus

$$T_{merging} = \left(\left\lceil \frac{T_{ack\_period}}{T_{sfrm}} \right\rceil + 1 \right) \cdot T_{sfrm} \tag{3.1}$$

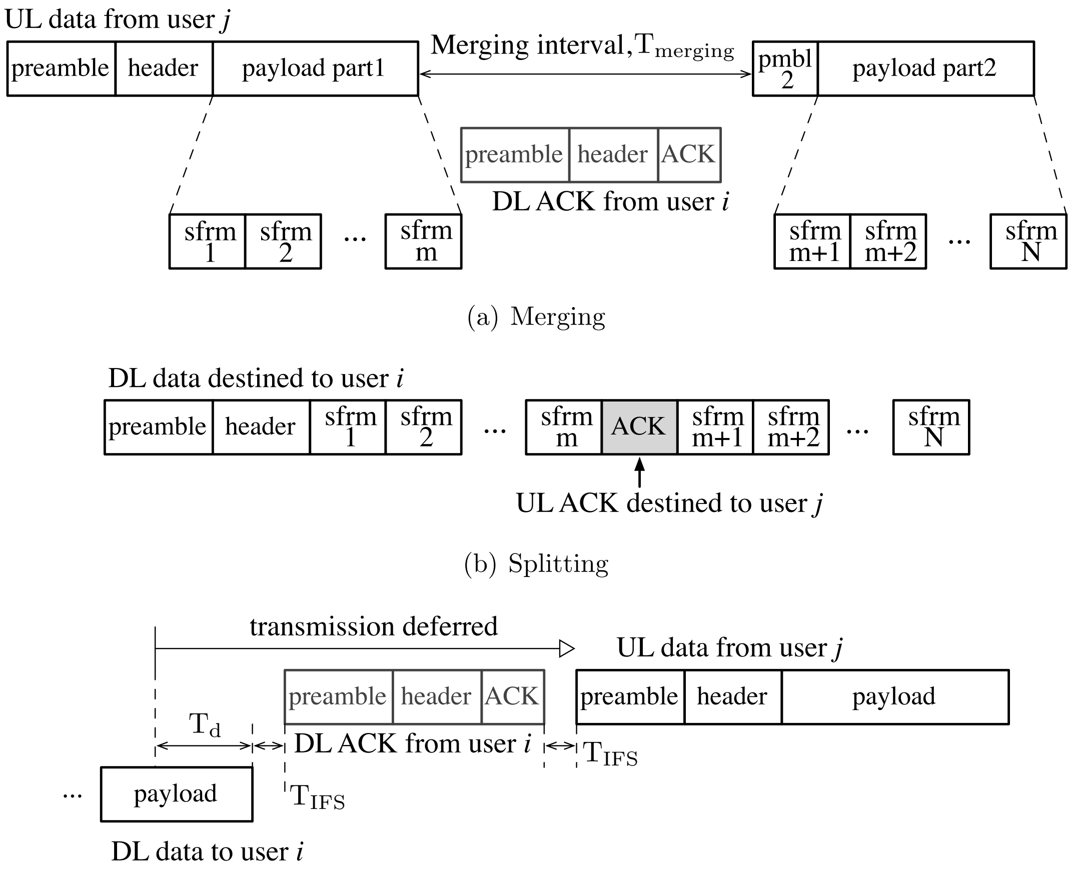
where $\lceil \cdot \rceil$ is a ceiling operator. When the packet transmission resumes, a short preamble (`pmbl2` in Figure 3.5(a)) can be *optionally* added to compensate for any frequency/phase offset that may arise during the merging interval.

Splitting transmission

The wireless switch, when acting as a transmitter, sends data packets to the user on the DL channel. If a data packet is received on the UL channel during a data packet transmission on the DL channel, the switch piggybacks an acknowledgment (to the UL data packet) on the DL data packet. Since the DL packet and the piggybacked UL acknowledgment are directed to two different users, we refer to this as a *splitting* transmission, an example of which is shown in Figure 3.5(b).

Deferred transmission

The last but not least important scenario that needs to be considered is the *deferred* transmission as shown in Figure 3.5(c). In the merging transmission, we postpone transmission of a partial payload if an overlap is predicted between the subframe transmission and the acknowledgment. However, if the arrival of the acknowledgment is predicted before the preamble and header part of the data



(a) Merging

(b) Splitting

(c) Deferred

Figure 3.5: Merging, splitting, and deferred transmission.

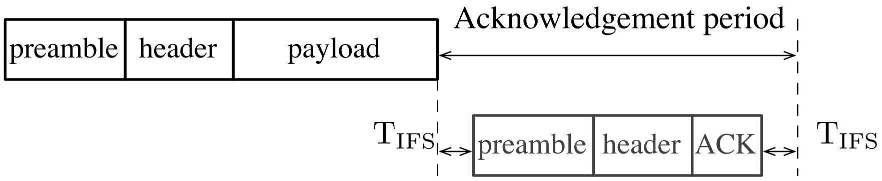


Figure 3.6: Acknowledgment period.

packet can be sent out, then the transmission of the entire data packet must be deferred.

3.2.2 Channel Efficiency

In this section, we analyze the channel efficiency, achieved by the CS-CLP. We define channel efficiency η_{ch}, as

$$\eta_{ch} = \frac{T_{pyld}}{T_{pyld} + T_{overhd}} \tag{3.2}$$

where T_{pyld} is the payload duration and T_{overhd} is for all the rest, including the duration for header and preamble, inter-frame spacing, idle time, and peer's acknowledgment period.

The maximum efficiency is achieved when both UL and DL channels are fully occupied by maximum-size data packets, acknowledgments, and inter-frame spacing without any channel idle time. This ideal case occurs when the UL traffic and the DL traffic are identical and time-aligned as shown in Figure 1.2, where acknowledgments do not interfere with data packets. However, traffic patterns are neither identical nor time-aligned, so some efficiency loss is inevitable. Unless merging and splitting transmissions are allowed, in the worst case the efficiency can be as low as 50% as shown in Figure 1.3(a). In the CS-CLP, data transmissions can be interrupted by acknowledgment either via merging or splitting transmissions, resulting in a drastic improvement in channel efficiency.

An example of UL and DL communications in the proposed system is depicted in Figure 3.7, which shows how and when *merging*, *splitting*, and *deferred* transmissions occur. We observe that channels are better utilized by employing merging and splitting transmissions.

We assume that the OFDM is used as a physical layer modulation. Basic parameters, such as inter-frame spacing, preamble duration, OFDM frame size, are from the IEEE 802.11 MAC and 11a PHY standards [9]. Following are additional assumptions made:

- The data size is 2304 bytes, which is the maximum size defined in IEEE 802.11 standards.

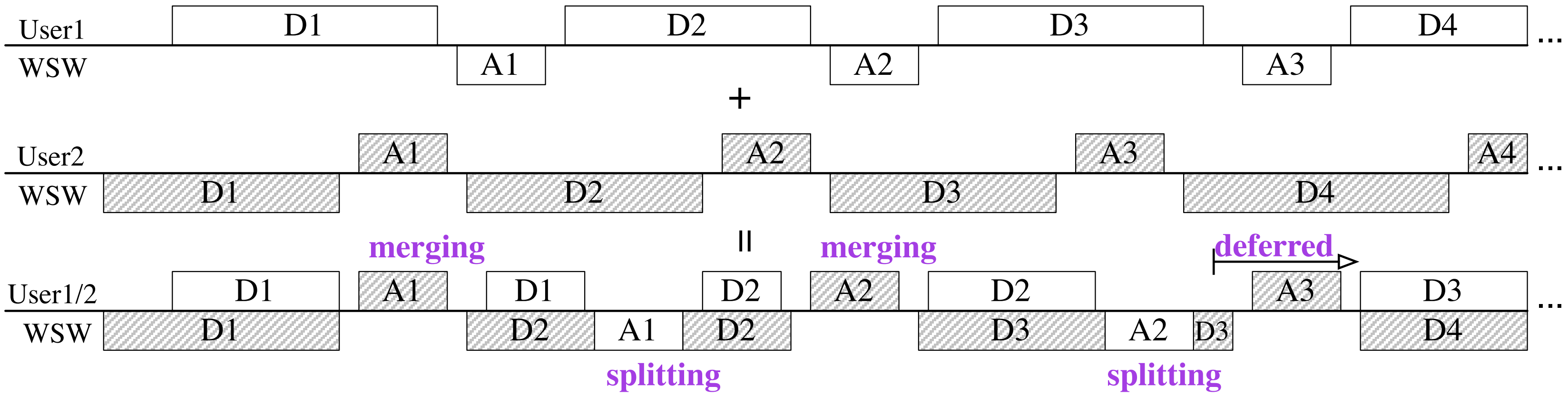


Figure 3.7: Channel sharing.

- The data rate is fixed at maximum, 54 Mbps, so that one OFDM symbol (of $4\,\mu s$ duration) carries 216 bits. Thus the payload duration is $T_{pyld} = 344\,\mu s$ (86 OFDM symbols).
- The long preamble defined in the IEEE 802.11 standard is used for the second preamble in the merging transmission (`pmbl2` in Figure 3.5(a)).
- The acknowledgment packet is BPSK-modulated (with the lowest data rate) so that its duration is fixed, and the acknowledgment payload is one OFDM symbol (24 bits). When it is piggybacked, it occupy one whole subframe.
- The header size is 72 bits (4, 32, 4, 12, and 20 bits respectively for the packet ID field, four address fields, the rate field, the payload length field, and the reserved field) and modulated with three OFDM symbols.
- One subframe consists of two OFDM symbols.
- A packet encounters at most one of these events: merging, splitting or deferring.

Table 3.1: Parameters.

Parameter	Value
Inter-frame spacing, T_{ifs}	$10\,\mu s$
Preamble duration, T_{pmbl}	$16\,\mu s$
2nd preamble duration, T_{pmbl2}	$8\,\mu s$
Header duration, T_{hdr}	$12\,\mu s$
Acknowledgment duration, T_{ack}	$32\,\mu s$
Subframe duration, T_{sfrm}	$8\,\mu s$
Ack. piggyback duration, T_{ackpg}	$8\,\mu s$
Payload duration, T_{pyld}	$344\,\mu s$

Parameters used for this analysis are summarized in Table 3.1. Based on the above assumptions and given parameters, we compute overheads and channel efficiencies for the ideal case (when packets are not interrupted), for the merging transmission, for the splitting transmission, and for the deferred transmission. Finally, we perform simulations to obtain the overall channel efficiency.

1. *Ideal case:*

$$T_{ack\_period} = T_{ifs} + T_{ack} + T_{ifs} = 52\,\mu s \tag{3.3}$$

$$\begin{aligned} T_{ideal\_overhd} &= T_{pmbl} + T_{hdr} + T_{ack\_period} \\ &= 80\,\mu s \end{aligned} \tag{3.4}$$

   From (3.2) and (3.4), the channel efficiency for the ideal case is $\eta_{ch}^{i} \approx 81.1\ \%$

2. *Merging transmission:* As previously mentioned, we assume that at most one interruption occurs during a packet transmission.

$$T_{merging} = \left(\left\lceil \frac{T_{ack\_period}}{T_{sfrm}} \right\rceil + 1 \right) \cdot T_{sfrm} = 64\,\mu s \tag{3.5}$$

$$\begin{aligned} T_{merging\_overhd} &= T_{ideal\_overhd} + T_{merging} + T_{pmbl2} \\ &= 152\,\mu s \end{aligned} \tag{3.6}$$

   From (3.2) and (3.6), the channel efficiency for merging transmission is $\eta_{ch}^{m} \approx 69.4\ \%$

3. *Splitting transmission:*

$$T_{splitting\_overhd} = T_{ideal\_overhd} + T_{ackpg} = 88\,\mu s \tag{3.7}$$

   From (3.2) and (3.7), the channel efficiency for splitting transmission is $\eta_{ch}^{s} \approx 79.6\ \%$

4. *Deferred transmission:* The worst-case overhead for the deferred transmission is $T_d = T_{pmbl} + T_{hdr}$ (see Figure 3.5(c)). Thus,

$$\begin{aligned} T_{ideal\_overhd} &\leq T_{deferred\_overhd} \\ &\leq T_{ideal\_overhd} + T_{ack\_period} + T_{pmbl} + T_{hdr} \end{aligned} \tag{3.8}$$

   From (3.2), (3.3), (3.4), and (3.8), the channel efficiency for the deferred transmission is: $68.3\ \% \leq \eta_{ch}^{d} \leq 81.1\ \%$.

The overhead for UL, $T_{overhd,UL}$, is a linear combination of T_{ideal_overhd}, $T_{merging\_overhd}$, and $T_{deferred\_overhd}$, and the overhead for DL, $T_{overhd,DL}$, is a linear combination of T_{ideal_overhd} and $T_{splitting\_overhd}$. However, due to the randomness of packet size and arrival times, deriving the coefficients of these combinations is analytically intractable. Moreover, long packets are more likely to encounter interruptions (merging, splitting or deferred) multiple times, which confounds the problem. Therefore, we resort to simulations to obtain channel efficiencies for different scenarios:

1. Both UL and DL backlogged and payload size fixed to maximum (2304 bytes, 43 subframes);

2. Both UL and DL backlogged and payload size uniformly distributed between 270 bytes (5 subframes) and 2304 bytes (43 subframes);

3. Both UL and DL backlogged and payload size fixed to 1296 bytes (24 subframes), which is the average value of scenario 2;

4. UL backlogged and DL idle.

3.3 Simulation

The simulation model was constructed using the Verilog hardware description language (Verilog-HDL) to verify the concurrent operation of the system. Simulations were conducted for 1000 packets transmitted in both directions. Results are given in Table 3.2. We observe that the DL is more efficient than the UL, mostly because the merging transmission, dominant in UL, has a larger overhead than the splitting transmission, dominant in DL. Scenarios 2 and 3 are less efficient than Scenario 1 because of the smaller average payload size. However, as the results of Scenarios 2 and 3 indicate, variations in packet length do not appear to have much effect on efficiency.

Table 3.2: HDL simulation results.

Condition	$\eta_{ch,UL}$	$\eta_{ch,DL}$	η_{ch}
scenario 1	72.01	79.84	75.93
scenario 2	58.78	69.09	63.94
scenario 3	57.63	68.95	63.29
scenario 4	81.20	—	—

3.4 Conclusion

The cross-layer protocol for channel sharing (CS-CLP) is presented in this chapter. The CS-CLP is a protocol devised to maximize channel utilization in a multi-channel wireless switching system. From the analysis and simulation results, a channel efficiency of $\approx 75\,\%$ is observed when users are backlogged. Based on this, it can be claimed that the proposed scheme using a K-port wireless switch with each port providing 20 MHz bidirectional bandwidth yields about 2K times as much capacity as an access point with 20MHz per channel.

Acknowledgements

The work in this chapter, in full, is a reprint of the following paper: J. Shim, K. Yun, and R. Cruz, *An Efficient Wireless Switching Architecture*, The Ninth Annual Wireless Telecommunication Symposium, Tampa, USA, Apr. 2010 [3]. The dissertation author is the primary author of this paper.

Chapter 4

Scheduling and Load Balancing

A key objective of the proposed wireless switch is to increase the usable bandwidth. To this end, the CS-CLP plays an important role in maximizing the efficiency of a channel pair allocated to a port. However, there are other concerns such as local fairness among users associated with a port, global fairness across ports, and load balancing across ports. All of these can be managed by a scheduler: a central coordinator that resides in the wireless switch (see Figure 3.2). The scheduler performs two layers of scheduling:

- *Local scheduling* for a port is responsible for providing fair services to the users associated with the port, which can be achieved by an algorithm such as the deficit round-robin (DRR) [18, 27] or its variants [28, 29].
- *Global scheduling* is needed to provide global fairness across ports as well as to maximize utilization of channels[1]. For this, more sophisticated scheduling algorithm is required.

This chapter introduces a dynamic scheduling scheme to achieve fairness and load balancing and compare it to the Least-Loaded-First (LLF) [68, 69] based one-time assignment of users, in which the scheduler selects a user and assigns it to the least loaded port one by one. The LLF assignment is assumed to be predetermined and to remain unchanged during the scheduling period.

[1] In non-backlogged case, underloaded ports are not fully utilized whereas users associated overloaded ports are starving.

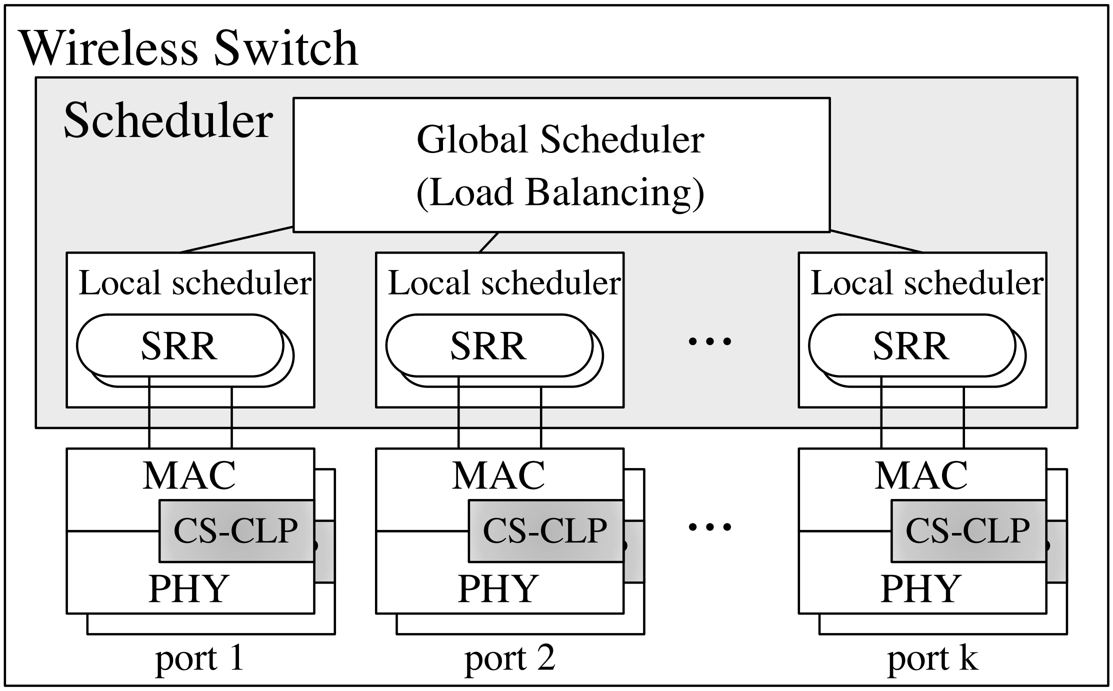


Figure 4.1: Wireless switch architecture.

4.1 Scheduler Architecture

As shown in Figure 4.1, the scheduler resides above the MAC layers of switch ports and delivers the scheduling information to users through each port. The scheduler consists of K local schedulers and a global scheduler: a local scheduler comprises two surplus round-robin (SRR) [28] based schedulers, each of which manages UL and DL separately to guarantee a local fairness at each port. The global scheduler balances loads among K ports so that users associated with different ports receive the same level of fairness. The goal of global scheduling is to balance the *accumulated* service received by every user, regardless of the port to which it is wirelessly connected at any instant. To achieve this, the proposed scheduler *compensates* for the lagging service experienced by users while being associated with a heavily loaded port, as explained in Section 4.4 in detail. The compensation is independently applied to UL and DL; however, a single load balancing is applied to all users, because the set of UL users and the set of DL users are not necessarily disjointed.

4.2 MAC Layer Support for Scheduling

The Media Access Control (MAC) layer provides a channel access control mechanism so that multiple nodes within a network can communicate with one another and share the medium properly. This section describes an essential part of the MAC protocol to support the scheduling operation in wireless switching systems.

The switch periodically broadcasts *beacon* packets on all DL channels (so that every user can "hear" them no matter which channel it is on). There are three operation periods: *communication setup period*, *scheduling period* and *user admission period*. As shown in Figure 4.2, each period follows a *beacon* packet which indicates the kind of service that the corresponding switch port provides during that period.

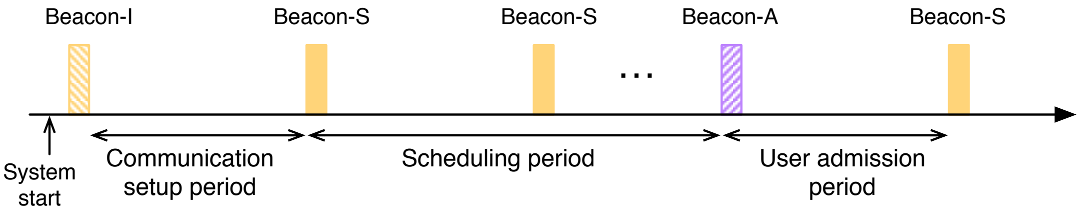


Figure 4.2: Operation periods.

4.2.1 Communication Setup Period

A *communication setup period* is the first time period immediately after the system starts. It is initiated by transmitting *Beacon-I* packets on all DL channels, which delivers a mapping information of channel pairs and switch ports as well as the duration of the period. Each user then hears the *Beacon-I* packet and transmits an admission request packet to the switch via the assigned channel. During this period, a contention based scheme [9] is used for selecting users to communicate with the switch. Users who have not been selected must wait until the next *user admission period* (described in Section 4.2.3).

When a user is admitted to the switching network, it requests its desired data rate to the switch—the desired data rate of a DL user is determined by the corresponding UL user—so that a scheduling contract between the switch and the user is made. The switch then computes the scheduling parameters to provide fair services to all users. The scheduling contract is updated at the end of *user admission period* if the set of users is changed (because some users go off-line during any of three periods and/or some users join during *user admission period*).

4.2.2 Scheduling Period

A *scheduling period* is a normal communication period, which is initiated by the switch when a *Beacon-S* packet is transmitted on a DL channel and is repeated periodically. During this period, services are provided to users who have been selected during the *communication setup period* or during *user admission periods* (described in the next subsection) based on the contract. Users may go off-line anytime during this period but, in order to receive service again, must be re-admitted.

4.2.3 User Admission Period

During each *user admission period*, users who were not admitted during the *communication setup period* or during previous *user admission periods*, and users who wish to start communication (either for the first time or after going off-line) can contend to be selected for communication with the switch.

A *user admission period* is initiated by the switch when a *Beacon-A* packet is transmitted on a DL channel. It is inserted periodically between *scheduling periods*. As it is a non-data-communicating period, frequent insertion decreases throughput of the network, whereas intermittent insertion causes delay of users' admission. In order to remedy this problem, *user admission periods* are inserted in a staggered manner as shown in Figure 4.3. Using this scheme, users have options to switch to a channel whose *user admission period* starts sooner. This is possible because *Beacon-S* delivers the information including the channel number for the next *user admission period* and when the following *user admission periods* will be inserted. For example, consider a user who wants to join the network. *Current time* is indicated by the dotted line in Figure 4.3, and the user is currently on channel pair 3 (CP3). Then the first beacon that the user hears on CP3 is the second *Beacon-S*, which notifies the user that the next *user admission period* will be on CP2 and the following on CP3. The user can either switch to CP2 to contend for the admission during the next *user admission period* or stay on the current channel for the following *user admission period*. If it fails to win the contention, it can switch to the next CP and try again until it is admitted.

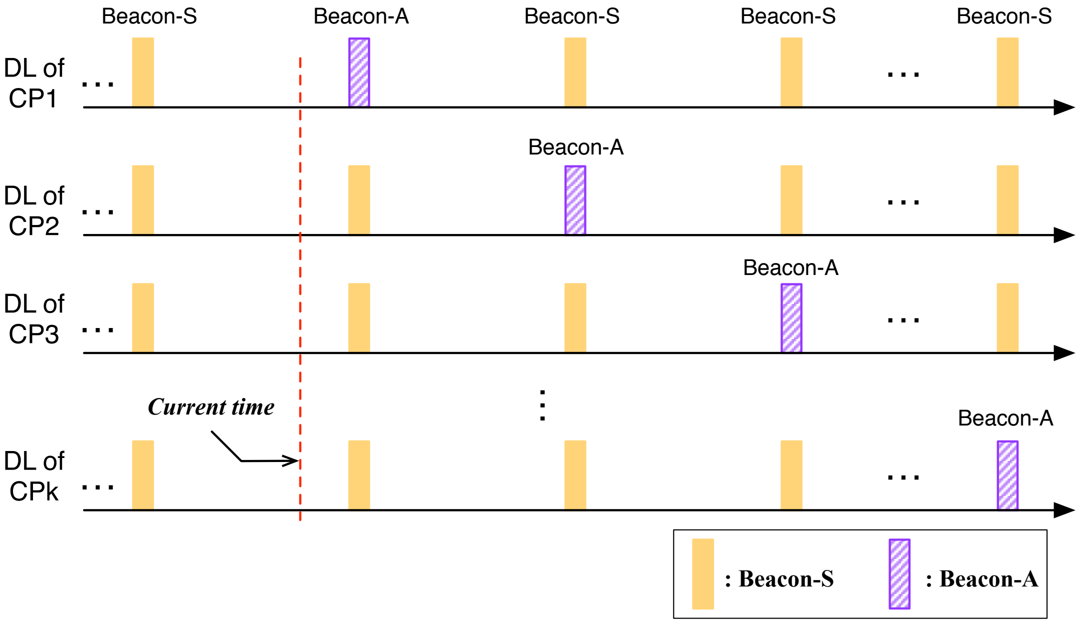


Figure 4.3: Staggered insertion of *user admission periods*.

4.3 Local Scheduling

The local scheduler at each port provides a scheduling function to a group of users associated with the port. A pair of SRR-based schedulers, which manages UL and DL traffic independently, constitutes a local scheduler. The local scheduler can schedule UL and DL traffic, as if they were independent, because any possible collision between UL (DL) ACK and DL (UL) data packets would be avoided by the CS-CLP. In the SRR scheduling—a variant of DRR—the scheduling decisions do not depend on the current queue status but on the past scheduling history, which is useful in scheduling for wireless communication where the server does not have knowledge of users' queues.[2] In the SRR scheduling, each user is served in a round robin manner. In each round, quantum Q_i is added to user i's deficit counter DC_i. If DC_i has a positive value, user i is allowed to transmit/receive data in this round, regardless of the packet size, and DC_i is decremented by the packet size. If DC_i has a non-positive value, user i is forced to skip this round and not allowed to transmit/receive until DC_i becomes positive.

Inherently, the SRR scheduler provides a weighted fair service: i.e., the service provided to a user is proportional to the weight allocated to each user. This dissertation considers a weighted *proportional* fair service: i.e., a user's achieved data rate is proportional to its weight *and* link rate.

Let w_i represent user i's requested data rate, which is used as the weight allocated to user i, c_i the link rate between user i and the switch, and d_i the offered data rate, which is the data rate offered to user i by the switch: i.e., the maximum data rate achievable by user i when it is backlogged. Thus $d_i \propto w_i c_i$. The next section describes the case for which users are backlogged. The non-backlogged case will be considered in the following section (Section 4.3.2).

4.3.1 Load and Quantum Calculation: Backlogged Case

The *load* imposed on a server (port) by a user is defined as the server's effort rendered to the user, in terms of the amount of *time* allocated to serve the user. Thus the load assigned to user i, $l_{o,i}$, is proportional to $\frac{d_i}{c_i}$ or w_i. If the total normalized load in a K-port wireless switch is K, user i's load is:

$$l_{o,i} = K \cdot \frac{w_i}{\sum_{j \in U w_j}}, \ U = \{1, \ 2, \ \cdots, \ N\}. \tag{4.1}$$

[2] Sharing the queue information among users and the server would require a significant overhead.

where

$$\sum_{i \in U} l_{o,i} = K. \tag{4.2}$$

Note that $l_{o,i} = \frac{d_i}{c_i}$. Because the offered data rate to a user is a function of the load allocated to the user and the link rate, user i's quantum, which represents the amount of data to be served on average during a certain time interval (e.g., a round), is given by

$$Q_{o,i} = l_{o,i} c_i \bar{q}. \tag{4.3}$$

where $\bar{q}$ is a scaling coefficient, which represents the average amount of time *per round* to serve all users when loads are balanced across all ports. Also, user i's quantum in time unit is given by

$$Q^t_{o,i} = \frac{Q_{o,i}}{c_i} = l_{o,i}\bar{q}. \tag{4.4}$$

4.3.2 Load Calculation: Non-backlogged Case

When the requested data rate, w_i, is less than the offered data rate, d_i, the excess service offered, $\frac{d_i - w_i}{c_i} = l_{o,i} - \frac{w_i}{c_i}$, should be assigned to other users whose requested data rate is greater than the offered data rate. Consider, for example, a case in which $N = 2, K = 1, c_1 = 10\,\text{Mbps}, c_2 = 5\,\text{Mbps}$. Users 1 and 2 request 6 Mbps and 3 Mbps respectively ($w_1 = 6\,\text{Mbps}$, $w_2 = 3\,\text{Mbps}$). Then $l_{o,1} = \frac{2}{3}$ and $l_{o,2} = \frac{1}{3}$. From $d_i = l_{o,i} c_i$,

$$\begin{aligned} d_1 &= \frac{2}{3} \times 10 = 6.67\,\text{Mbps} > w_1\,(= 6\,\text{Mbps}); \\ d_2 &= \frac{1}{3} \times 5 = 1.67\,\text{Mbps} < w_2\,(= 3\,\text{Mbps}). \end{aligned}$$

In this case, the excess bandwidth (0.67 Mbps) offered to user 1 should be reallocated to user 2. Note that cases such as this occur when $\min(c_i) < \sum w_i < \max(c_i)$.

Using the following process, we can obtain a set of loads which yields no excess service, $\mathbf{l}^*_o = \{l^*_{o,1}, l^*_{o,2}, \cdots, l^*_{o,N}\}$:

1. Let $\mathcal{U} = U$ and the maximum available load for the wireless switch be K.

2. Compute $l_{o,i}$ $\forall\, i \in \mathcal{U}$ using equation (4.1).

3. If $w_i \geq d_i (= l_{o,i} c_i)$ $\forall\, i \in \mathcal{U}$, then $l^*_{o,i} \leftarrow l_{o,i}$ $\forall\, i \in \mathcal{U}$ and terminate.

4. For each user $i \in \mathcal{U}$, if $w_i < d_i (= l_{o,i} c_i)$, then $l^*_{o,i} \leftarrow \frac{w_i}{c_i}$, $K \leftarrow K - l^*_{o,i}$, and $\mathcal{U} \leftarrow \mathcal{U} - \{i\}$.

5. Repeat (2)–(4) until $\mathcal{U} = \emptyset$.

The ascending ordered set for the resulting load set, $\mathbf{l}^*_o$, is *lexicographically maximal* among all the feasible load sets. This optimization is performed during the scheduling parameter computation described in section 4.2.

4.4 Global Scheduling (Load Balancing)

Load $l_{o,i}$ represents a share of time allocated to user i when the time to serve all users each round is normalized to K. In general, $\sum_{i \in U_j} l_{o,i}, j \in P$ is not equal to 1, although it would move toward 1 when the dynamic load balancing algorithm is applied. Consider, as an example, a system with four users ($N = 4$) and two ports ($K = 2$). Users $1 - 4$ request 10 Mbps, 10 Mbps, 10 Mbps, and 5 Mbps respectively. Then $l_{o,1}$, $l_{o,2}$, $l_{o,3}$, and $l_{o,4}$ are $\frac{4}{7}$, $\frac{4}{7}$, $\frac{4}{7}$, and $\frac{2}{7}$. Because $K = 2$, users are divided into two groups: one with an aggregate load of $\frac{8}{7}$ and the other with $\frac{6}{7}$. Since each group can only be served by the corresponding port, without dynamic load balancing, a fairness issue arises.[3]

The goal of global scheduling is achieving fairness across all users as if they were served by a single server. In order to attain this goal, the global scheduling scheme proposed in this dissertation performs two functions: *compensation* and *load balancing*.

[3] This is a typical situation in which loads cannot be evenly divided into K ports. Worse yet, when UL and DL loads are not identical, even divisions in both UL and DL are even more unlikely.

4.4.1 Lagging Service Compensation

The compensation scheme adjusts the loads of users who have been penalized while being assigned to ports with heavier loads so that the services that have been “lagging” are compensated. The degree of fairness is measured by keeping track of each user’s *service count*: the number of times a unit service is rendered to each user. Formally, service count, r_i, is defined as the number of times $Q_{o,i}$ is added to user i’s deficit counter. This quantity is supposed to be an integer value in a normal DRR scheduler. However, as will be discussed in this section, due to the compensation scheme considered, non-integer times of quanta can be assigned, hence the size of quanta is not necessarily an integer. As the objective of the fair scheduling is providing fair amount of service to users, the scheduler attempts to equalize every user’s service count, i.e., $r_i = r_j$, $\forall i \neq j \in U$.

User i’s lagging service is then the discrepancy between its service count, r_i, and the service count of the user who has benefitted the most by the load imbalance, $r_{\max}$: i.e., $r_{\max} - r_i$. The modified load and quantum after the compensation are then given by

$$\begin{aligned} l_i &= l_{o,i} + l_{o,i}\{\alpha\,(r_{\max} - r_i)\} \\ &= l_{o,i}\{1 + \alpha\,(r_{\max} - r_i)\} \\ &= l_{o,i}\beta_i. \end{aligned} \tag{4.5}$$

$$Q_i = l_i c_i \bar{q} = Q_{o,i}\beta_i \tag{4.6}$$

$$Q_i^t = l_i \bar{q} = Q_{o,i}^t \beta_i. \tag{4.7}$$

where α denotes the convergence coefficient and $\beta_i = 1 + \alpha(r_{\max} - r_i)$ user i’s *compensation factor*. When loads are balanced, $\beta_i = 1, \; \forall \; i \in U$. On the other hand, when loads are not balanced, each user receiving a lagging service is compensated with an adjusted load: e.g., if user j receives a lagging service, its *modified load* becomes $l_i = l_{o,j}\beta_j > l_{o,j}$.

Figure 4.4 shows a block diagram of the lag compensation control. The set of quanta, $\mathbf{Q} = \{Q_i | i \in U\}$, and the set of modified loads, $\mathbf{l} = \{l_i | i \in U\}$, are computed from the set of weights $\mathbf{w} = \{w_i | i \in U\}$. $\mathbf{Q}$ is used for SRR-based

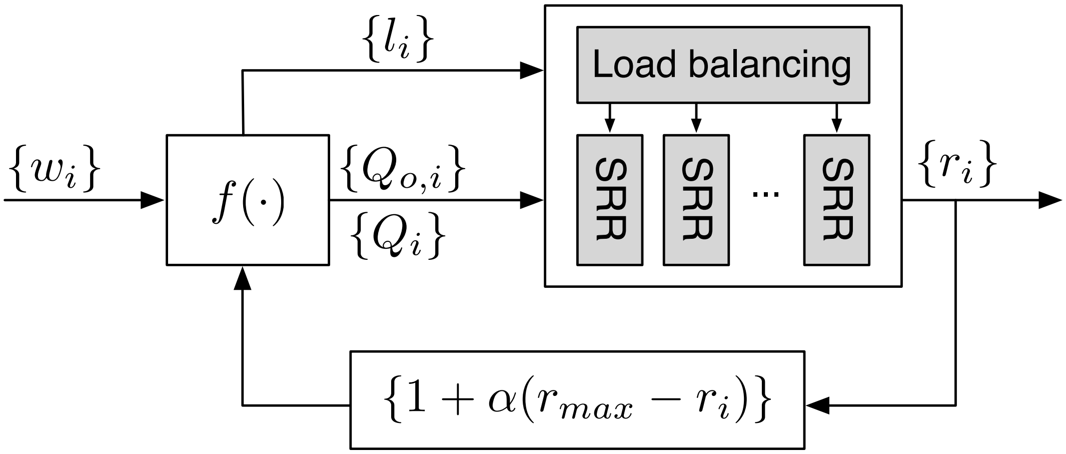


Figure 4.4: Lag compensation.

local scheduling and **l** for load balancing. At each round, the service count of each user is updated. Any discrepancies in users' service counts, which signifies load imbalance, cause **Q** and **l** to be updated.

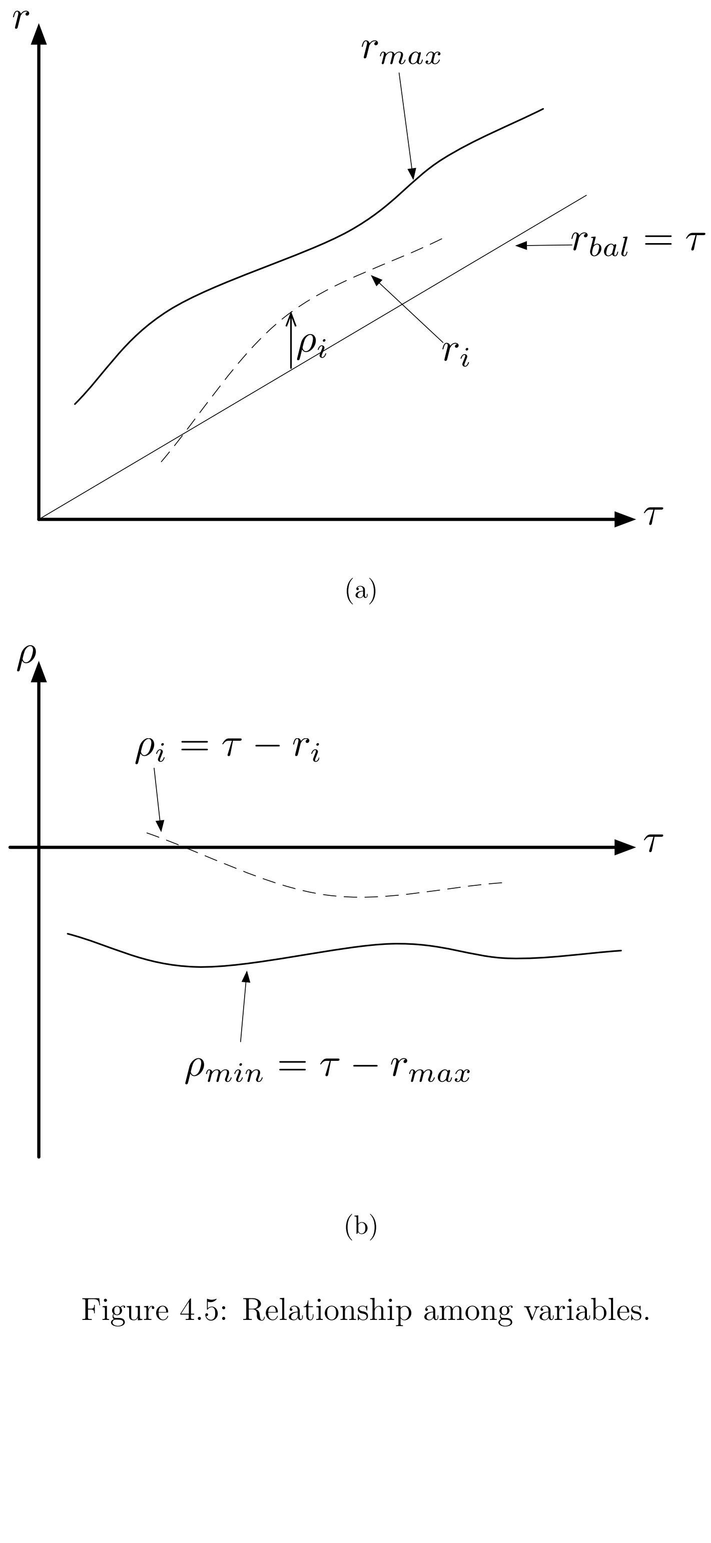
4.4.2 Analysis of the system

Let L_j denote the aggregate modified load for port $j \in P$: i.e., $L_j = \sum_{i \in U_j} l_i$. From (4.7) the mean round-robin turn in port j is $\sum_{i \in U_j} Q_i^t = \sum_{i \in U_j} l_i \bar{q} = L_j \bar{q}$ and the incremental service count of user $i \in U_j$ per round is β_i. Assume that quanta are assigned and consumed (packets are transmitted/received) continuously over time so that the service count also increases continuously. If normalized time $\tau = t/\bar{q}$, then the derivative of user i's service count, r_i, with respect to τ[4] is given by

$$\begin{aligned} \dot{r}_i = \frac{dr_i}{d\tau} &= \frac{\beta_i}{L_j} \\ &= \frac{1}{L_j}\{1 + \alpha(r_{max} - r_i)\}. \end{aligned} \tag{4.8}$$

where $r_{max} = \max_{i \in U}(r_i)$ is also a function of time.

[4] Throughout this chapter, a dot ($\dot{}$) over the variable, for example $\dot{r}_i$, implies the derivative with respect to τ, i.e., $\frac{d}{d\tau}$.



(a)

(b)

Figure 4.5: Relationship among variables.

The service count in a load balanced system is $r_{bal} = t/\bar{q} = \tau$. Let ρ_i, $i \in U$, the lagging amount of r_i compared to r_{bal}: i.e., $\rho_i = \tau - r_i$. This is referred to as the *skew* of user i's service count (user i's skew in short). Note that the skew for r_{max} is given by

$$\tau - r_{max} = \min_{i \in U} \rho_i \triangleq \rho_{min}. \tag{4.9}$$

From $\beta_i = 1 + \alpha(r_{max} - r_i)$, it follows that

$$\beta_i = 1 + \alpha(\rho_i - \rho_{min}). \tag{4.10}$$

when, substituted in (4.8), and from $\dot{\rho}_i = -\dot{r}_i + 1$, it follows that

$$\begin{aligned} \dot{\rho}_i &= -\dot{r}_i + 1 \\ &= -\frac{\alpha}{L_j}\left(\rho_i - \rho_{min} + \frac{1}{\alpha}\right) + 1. \end{aligned} \tag{4.11}$$

From (4.5), L_j is given by

$$\begin{aligned} L_j &= \sum_{i \in U_j} l_{o,i}\beta_i \\ &= \sum_{i \in U_j} l_{o,i}\left(1 + \alpha(\rho_i - \rho_{min})\right) \\ &= \alpha \sum_{i \in U_j} l_{o,i}\rho_i + \alpha L_{o,j}\left(\frac{1}{\alpha} - \rho_{min}\right). \end{aligned} \tag{4.12}$$

where $L_{o,j} = \sum_{i \in U_j} l_{o,i}$. By substituting (4.12) to (4.11),

$$\dot{\rho}_i = -\frac{\rho_i - \rho_{min} + \frac{1}{\alpha}}{\sum_{i \in U_j} l_{o,i}\rho_i + L_{o,j}(\frac{1}{\alpha} - \rho_{min})} + 1, \ \forall i \in U_j. \tag{4.13}$$

The dynamic system described by (4.13) is a non-linear system. The aim of this dissertation is to have the system remain in a stable state as well as to equalize the skews amongst users in the system. It will be shown that globally equalizing skews is equivalent to achieving fairness. Before proceeding to further discussions, definitions of fairness and equilibrium of skew are given below.

Definition 1 (Local fairness in group $j \in P$)**.** *The local fairness in group $j \in P$ is achieved at $t = t_o$ ($\tau = \tau_o$ accordingly), when $r_i = r_k$, $\forall i \neq k \in U_j$ for $t \geq t_o$ (or $\tau \geq \tau_o$).*

Definition 2 (Local equilibrium of skew in group $j \in P$)**.** *The local equilibrium of skew in group $j \in P$ is reached at $t = t_o$ (or $\tau = \tau_o$), when, for constant value $\rho_{ss}^{(j)}$, $\rho_i = \rho_{ss}^{(j)}$, $\forall i \in U_j$ for $t \geq t_o$ (or $\tau \geq \tau_o$).*

Definition 3 (Global fairness)**.** *The global fairness is achieved at $t = t_o$ ($\tau = \tau_o$ accordingly), when local fairness is achieved in each group and $r_i = r_k$, $\forall i \neq k \in U$ for $t \geq t_o$ (or $\tau \geq \tau_o$).*

Definition 4 (Global equilibrium of skew)**.** *The global equilibrium is reached at $t = t_o$ ($\tau = \tau_o$ accordingly), when local equilibrium is achieved in each group and, for constant value ρ_{ss}, $\rho_{ss} = \rho_{ss}^{(j)}$, $\forall j \in P$ for $t \geq t_o$ (or $\tau \geq \tau_o$).*

Lemma 1. *Local fairness in group $j \in P$ is achieved if local equilibrium of skew in group $j \in P$ is reached.*

Proof of Lemma 1. From, $\rho_i = \tau - r_i, \forall i \in U_j$, if $\rho_i = \rho_k, \forall i \neq k \in U_j$, $r_i = r_k$, $\forall i \neq k \in U_j$ trivially. □

It is worth noting that necessary conditions for local equilibrium are: (1) $\dot{\rho}_i = 0$, $\forall i \in U_j$, or equivalently, $\dot{r}_i = 1$, $\forall i \in U_j$ and (2) $\rho_i = \rho_k$, $\forall i \neq k \in U_j$. Since only the latter condition is guaranteed by local fairness, the converse of Lemma 1 is not true.

Lemma 2. *Global fairness is achieved if and only if global equilibrium of skew is reached.*

Proof of Lemma 2. ($\Longleftarrow$) From, $\rho_i = \tau - r_i$, $\forall i \in U$, if global equilibrium is reached so that $\rho_i = \rho_k$, $\forall i \neq k \in U$, $r_i = r_k$, $\forall i \neq k \in U$ trivially. Hence global fairness is achieved.

($\Longrightarrow$)Suppose that global fairness is achieved at $t = t_1$, then by definition of global fairness, for $t \geq t_1$, $r_i = r_k$, $\forall i \neq k \in U$, which leads to $\rho_i = \rho_k$, $\forall i \neq k \in U$ as well as $\dot{\rho}_i = \dot{\rho}_k, \forall i \neq k \in U$. The latter condition holds only when $\dot{\rho}_i = 0$, $\forall i \in U$. Therefore, global equilibrium is reached. □

Suppose that K ports are served during time interval $[t_o, t_1]$, the duration of which is denoted as $T = t_1 - t_o$, and there is no idling of ports during this interval. The total time serviced by K ports are given by

$$\begin{aligned} KT &= \sum_{i\in U} Q^t_{o,i}(r_i - r_{o,i}) \\ &= \sum_{i\in U} l_{o,i}\bar{q}(r_i - r_{o,i}) \\ &= \sum_{i\in U} l_{o,i}\bar{q}(\hat{\tau} - (\rho_i - \rho_{o,i})). \end{aligned} \tag{4.14}$$

where $r_{o,i}$ denotes an initial value of r_i at the beginning of the interval (at $t = t_o$), $\rho_{o,i}$ an initial value of ρ_i, and $\hat{\tau}$ the duration of the interval in the scaled unit, that is, $\hat{\tau} = T/\bar{q}$. Note that $\sum_{i\in U} l_{o,i}\bar{q}\hat{\tau} = KT$ hence (4.14) reduces to

$$\sum_{i\in U} l_{o,i}\rho_i = \sum_{i\in U} l_{o,i}\rho_{o,i}. \tag{4.15}$$

As $l_{o,i}\bar{q}\rho_i$ represents the lagging service of user $i \in U$ in time unit compared to the reference, the normalized lagging service of user i is denoted as $l_{o,i}\rho_i$. Therefore, (4.15) represents the total normalized lagging service of the system. If the initial values of ρ_i's at the beginning of the interval are zeros, i.e., $\rho_{o,i} = 0$, $\forall i \in U$, (4.15) becomes 0. This implies that the lagging service for some users in the system indicates the leading service for other users so that the summation of those terms is zero.

Borrowing from the concept of the *center of mass*, $\bar{\rho}$ is defined as *skew center* of the system, which is represented by

$$\bar{\rho} = \frac{\sum_{i\in U} l_{o,i}\rho_i}{\sum_{i\in U} l_{o,i}} = \frac{\sum_{i\in U} l_{o,i}\rho_i}{K}. \tag{4.16}$$

From (4.15), the skew center of the overall system $\bar{\rho}$ is constant over time. It also follows that, when zero initial values of ρ_i's are assumed (i.e., $\rho_{o,i} = 0$, $\forall i \in U$), $\bar{\rho} = 0$, indicating that skews are distributed around the origin.

Similarly, the skew center of group j, $\bar{\rho}^{(j)}$, is defined by

$$\bar{\rho}^{(j)} = \frac{\sum_{i\in U_j} l_{o,i}\rho_i}{\sum_{i\in U_j} l_{o,i}} = \frac{\sum_{i\in U_j} l_{o,i}\rho_i}{L_{o,j}}, \ \forall j \in P. \tag{4.17}$$

Lemma 3. *The skew center of group $j \in P$ is an increasing function of time t (or τ) when $L_{o,j} > 1$, an decreasing function of time when $L_{o,j} < 1$, or a static point when $L_{o,j} = 1$.*

Proof of Lemma 3. Suppose a set of users in group $j \in P$ is served during time interval, $[t_o, t_1] (= [\bar{q}\tau_o, \bar{q}\tau_1])$, the duration of which is given by $T = t_1 - t_o$, and there is no idling of the port during this interval. By the same analysis used in (4.14),

$$T = L_{o,j}\bar{q}\hat{\tau} - \sum_{i \in U_j} l_{o,i}\bar{q}(\rho_i - \rho_{o,i}). \tag{4.18}$$

where $\hat{\tau} = \tau_1 - \tau_o = T/\bar{q}$. (4.18) becomes

$$\sum_{i \in U_j} l_{o,i}\rho_i = \sum_{i \in U_j} l_{o,i}\rho_{o,i} + (L_{o,j} - 1)\hat{\tau}. \tag{4.19}$$

when, substituted in (4.17),

$$\bar{\rho}^{(j)} = \frac{\sum_{i \in U_j} l_{o,i}\rho_{o,i}}{L_{o,j}} + \left(1 - \frac{1}{L_{o,j}}\right)\hat{\tau}. \tag{4.20}$$

From (4.20), when $L_{o,j} = 1$ the second term goes to zero such that the skew center becomes a constant, whereas when $L_{o,j} > 1$ it is decrease over time (as $\hat{\tau}$ increases), and is increases when $L_{o,j} < 1$. □

Lemma 4. *Given the system described by (4.13), skews, ρ_i's ($i \in U_j$), converge to one another as $\tau \to \infty$.*

Proof of Lemma 4. Consider (4.13),

$$\dot{\rho}_i = -\frac{\rho_i - \rho_{min} + \frac{1}{\alpha}}{L_{o,j}\left(\frac{\sum_{i \in U_j} l_{o,i}\rho_i}{L_{o,j}} - \rho_{min} + \frac{1}{\alpha}\right)} + 1,\ \forall i \in U_j. \tag{4.21}$$

Without loss of generality, assume that user indices, $U_j = \{1, 2, \cdots, N_j\}$, are sorted in the increasing order of $\rho_i, i \in U_j$, then

$$\rho_1 \leq \rho_2 \leq \cdots \leq \rho_{N_j}. \tag{4.22}$$

Since $l_{o,i} > 0$, $\forall i \in U_j$ and the denominator of the first term of (4.21) is common to all users in group $j \in P$, substituting (4.21) in (4.22) gives

$$\dot{\rho}_1 \geq \dot{\rho}_2 \geq \cdots \geq \dot{\rho}_{N_j}. \tag{4.23}$$

where equality between adjacent derivatives holds only when the equality of the corresponding pair of ρ_i's in (4.22) holds.

Now consider ρ_p and ρ_q, such that $p, q \in U_j$, $p = q+1$ and $\rho_p > \rho_q$. Because $\dot{\rho}_p < \dot{\rho}_q$, the difference, $\rho_p - \rho_q$, decreases until it reaches equality: i.e., $\rho_p = \rho_q$. Then the equality between corresponding derivatives is also achieved: i.e., $\dot{\rho}_p = \dot{\rho}_q$. This applies any pair of ρ's whose indices are adjacent to each other. Once these equalities are reached, they remain equal. Therefore, as $\tau \to \infty$, all ρ_i's, $i \in U_j$, converges to a common point. □

Lemma 5. *For the system described in Lemma 4, the converging point of skews in group j is given by*

$$\rho_{ss}^{(j)} = \frac{\sum_{i \in U_j} l_{o,i}\rho_{o,i}}{L_{o,j}} + \left(1 - \frac{1}{L_{o,j}}\right)(\tau - \tau_o). \tag{4.24}$$

where $\rho_{o,i}$, $i \in U_j$ represents the initial value of ρ_i at $\tau = \tau_o$.

Proof of Lemma 5. The linear combination of $\dot{\rho}_i$'s, $i \in U_j$, given by (4.21) with $\{l_{o,i} | i \in U_j\}$ as coefficients yields

$$\sum_{i \in U_j} l_{o,i}\dot{\rho}_i = \sum_{i \in U_j} l_{o,i} - 1 = L_{o,j} - 1. \tag{4.25}$$

Define

$$\sigma_i \triangleq \left(\frac{1}{L_{o,j}} - 1\right)\tau + \rho_i,\ i \in U_j. \tag{4.26}$$

which is equivalent to $\sigma_i \triangleq \frac{1}{L_{o,j}}\tau - r_i$. It follows that the converging point of σ_i's is given by

$$\sigma_{ss}^{(j)} = \left(\frac{1}{L_{o,j}} - 1\right)\tau + \rho_{ss}^{(j)}. \tag{4.27}$$

and the initial value of σ_i, $i \in U_j$ is given by

$$\sigma_{o,i} = \left(\frac{1}{L_{o,j}} - 1\right)\tau_o + \rho_{o,i},\ \forall i \in U_j. \tag{4.28}$$

Taking derivatives of both sides of (4.26) yields $\dot{\rho}_i = \dot{\sigma}_i - \left(\frac{1}{L_{o,j}} - 1\right)$, when, substituted in (4.25),

$$\sum_{i \in U_j} l_{o,i}\dot{\sigma}_i = 0. \tag{4.29}$$

Taking integrals over the interval $[\sigma_{o,i}, \sigma_{ss}^{(j)}]$ on both sides of (4.29) yields

$$\sum_{i \in U_j} \int_{\sigma_{o,i}}^{\sigma_{ss}^{(j)}} l_{o,i} \dot{\sigma}_i d\tau = \sum_{i \in U_j} l_{o,i} \left(\sigma_{ss}^{(j)} - \sigma_{o,i} \right) = 0. \tag{4.30}$$

which leads to

$$\begin{aligned} \sum_{i \in U_j} l_{o,i} \sigma_{ss}^{(j)} &= \sum_{i \in U_j} l_{o,i} \sigma_{o,i} \\ \equiv \quad \sigma_{ss}^{(j)} &= \frac{\sum_{i \in U_j} l_{o,i} \sigma_{o,i}}{L_{o,j}}. \end{aligned} \tag{4.31}$$

By substituting (4.27) and (4.28),

$$\rho_{ss}^{(j)} = \frac{\sum_{i \in U_j} l_{o,i} \rho_{o,i}}{L_{o,j}} + \left(1 - \frac{1}{L_{o,j}} \right) (\tau - \tau_o). \tag{4.32}$$

□

Note that the converging point of skew in group j given in (4.24) is the same as the group j's skew center given in (4.20). Therefore, from Lemma 4 and Lemma 5, Theorem 1 follows.

Theorem 1. *Given the system described by (4.13), ρ_i, $\forall i \in U_j$ converges to the skew center of group j, $\forall j \in P$, as $\tau \to \infty$.*

Proof of Theorem 1. The proof of this theorem is trivial from Lemma 4 and Lemma 5. □

Now consider a special case of Theorem 1, where $L_{o,j} = 1, i \in P$. As described in Lemma 3, the skew center of a group is a static point if the corresponding group has an aggregate load of unity. Hence, the following Corollary:

Corollary 1. *If a system is load-balanced, where each port is evenly loaded: i.e., $L_{o,j} = 1, \forall j \in P$, local equilibrium of skew in each group can be reached and, for the set of initial values $\{\rho_{o,i} \,|\, i \in U\}$, the converging point, $\rho_{ss}^{(j)}$, is given by $\rho_{ss}^{(j)} = \sum_{i \in U_j} l_{o,i} \rho_{o,i}$, which is the skew center of group j.*

From Corollary 1, it follows that, in a load-balanced system, the skew center of group $j \in P$, its static converging point, is a function of the set of initial values, $\{\rho_{o,i} | i \in U_j\}$. Thus, with the identical skew centers or identical initial values converge to an identical skew center; hence the global equilibrium of skew is reached.

4.4.3 Dynamic Load Balancing

The aim of the scheduling scheme is to provide fair service to users in the same group as well as to guarantee fairness across groups (ports). As seen Theorem 1, in the (continuous) system described by (4.13), each user's skew in group $j \in P$ converges to the skew center of the group regardless of the initial skew of each user, $\{\rho_{o,i} | i \in U_j\}$, and the aggregate load, $L_{o,j} (= \sum_{i \in U_j} l_{o,i})$, $j \in P$. However, since the skew center of each group is moving toward a different direction and with a different rate depending on its aggregate load, $L_{o,j}$, $j \in P$ (see (4.20) and Lemma 3), unless loads are balanced, global equilibrium of skew cannot be reached; thus neither is global fairness.

Because the weighted service in SRR is based on each user's quantum size relative to those of others served by the same server, the compensation scheme, albeit compensating lagging users in the same group, would not guarantee fairness globally across all the groups. The load balancing, the main feature of the global scheduling scheme, remedies this problem by moving users from the ports with heavier loads to the ports with lighter loads.

If the set of loads, $\{l_{o,i} | i \in U\}$, can be evenly divided into K groups, simply balancing loads by re-assigning users from one group to another may be one solution to achieve fairness. As can be seen from Corollary 1, even in this case, global fairness is achieved only when the skew center of every group is identical; otherwise discrepancies amongst skew centers of groups upon completion of load balancing[5] will remain as service discrepancies. These discrepancies, caused by initial skews and the skews occurring while re-assigning users, may be negligible if

[5] Note that the skew center of the group, out of(or into) which a user moves will change depending on the user's skew

it is small enough when compared to the actual value of each user's total service count. However, in practical scenarios, it is not always true: for example, balancing load takes time, in the meantime non-trivial amount of skew can occur.

Furthermore, loads are not generally evenly divisible. As Lemma 3, the remaining load imbalance (after a best-effort based load balancing), causes divergence of skew centers hence divergence of each user's skew (Theorem 1). Thus it is impossible to attain global fairness (Lemma 2). In fact, instantaneously re-assigning users as in an idealized fluid system so as to maintain each group's skew center, $\bar{\rho}_j$, $j \in P$, at the skew center of the system, $\bar{\rho}$, is needed, which clearly is not feasible in a packet based wireless communication system. The most reasonable way to approximate fair service is to prevent divergence of skews by dynamically re-assigning users whenever possible so that skews are *bounded* in a certain range. This would not guarantee global fairness of Definition 3, but, by having skews be bounded, global fairness in an approximate sense can be achieved. Before proceeding with discussion, a metric is introduced to measure the fairness of users, called *fairness index* (FI) defined in [18]:

$$\mathtt{FI}_i = \frac{served_i}{\sum_{j \in U} served_j} \Bigg/ \frac{l_{o,i}}{\sum_{j \in U} l_{o,j}}. \tag{4.33}$$

where $\mathtt{FI}_i$ is the fairness index of user $i \in U$ and $served_i$ is the amount of airtime that it spends transmitting packets (in the UL case)/receiving packets (in the DL case). Note that when a user receives the exact amount of service that it is supposed to receive, its $\mathtt{FI} = 1$, if it receives more service (leading service), $\mathtt{FI} > 1$, and if it receives less service (lagging service), $\mathtt{FI} < 1$. Now *global fairness in an approximate sense* is defined to measure the fairness of users in practical systems:

Definition 5 (global fairness in an approximate sense)**.** *Global fairness in an approximate sense for $\delta \geq 0$ is achieved, at $t = t_o$ ($\tau = \tau_o$ accordingly), if $|\mathtt{FI}_i - \mathtt{FI}_k| \leq \delta, \forall i \neq k \in U$ for $t \geq t_o$ (or $\tau \geq \tau_o$).*

Consider a K-port wireless switching system, equipped with the SRR based scheduling shown in Figure 4.1, where (1) load and quantum computation and local scheduling are performed as described in Section 4.3 and (2) modified load and quantum for compensation of lagging service is described by (4.5), (4.6) and (4.7). The service count in an idealized continuous system with balanced load, r_{bal}, is given by $r_{bal} = \tau$. Note that normalized time $\tau = t/\bar{q}$. Let $\rho_i = \tau - r_i$ denote the skew of the service count of user $i \in U$ compared to the ideal case. Then, the aggregate modified load of group $j \in P$ is given by

$$L_j = L_{o,j} + \alpha L_{o,j}(\bar{\rho}^{(j)} - \rho_{min}),\ \forall j \in P \tag{4.34}$$

where $\rho_{min} = \min_{i \in U} \rho_i = \tau - r_{max}$. The objective of the dynamic load balancing is the global fairness in an approximate sense for the smallest possible value for the skew bound. To this end, this dissertation proposes a dynamic load balancing algorithm based on the following conjecture.

Conjecture 1. *Consider aggregate modified load, $L_j, j \in P$, given in (4.34), to represent the load imposed on the port j. L_j can be regarded as a combination of two terms with convergence coefficient α: the aggregate load of group $j \in P$ and its aggregate skew of service compared to the smallest skew* [6]*: that is, it represents the aggregate lagging service of the group.*

Suppose there is a dynamic load balancing policy, which attempts to balance the aggregate modified loads (L_j's, $j \in P$) amongst groups by dynamically re-assigning users from one group (port) to another. Consider group $l \in P$ whose aggregate modified load, L_l, is given by (4.34). If it has larger skew, $(\bar{\rho}^{(l)} - \rho_{min})$, L_l becomes larger for some given $L_{o,l}$ and no additional load can be assigned to group l and users are prompted to move out of the group sooner to balance the loads. This allows the group to keep $L_{o,l}$ small so that it can reduce skew center $\bar{\rho}^{(l)}$ (thus reducing L_l). Similarly, if group $m \in P$ has a smaller skew, accordingly a smaller L_m for given $L_{o,m}$, then it has more capacity to accept additional loads from other groups. Thus some time the $L_{o,m}$ and its skew center, $\bar{\rho}^{(m)}$ (hence L_m) would be made larger. After the load balancing is applied, the situations for group l and group m would be reversed. Therefore, dynamically re-assigning users from

[6] ρ_{min} is considered as a reference instead of $\bar{\rho}$, from which the deviation of skew center is measured, in order to keep the aggregate load, L_j, positive. Note that the term ρ_{min} corresponds to r_{max} in (4.5) and $\bar{\rho}$ to $r_{bal}(= \tau)$. If r_{bal} replaces r_{max}, for a large skew, l_i becomes negative, which leads to a negative quantum, Q_i, which is not feasible in SRR scheduling.

a group with a larger aggregate modified load to a group with a smaller one, would bound its skew center within a certain range and $|\mathtt{FI}_i - \mathtt{FI}_k|, i \neq k \in U$ *to a small value of* δ*. Consequently, global fairness in an approximate sense for* $\delta > 0$ *can be achieved.*

Dynamic Load Balancing Algorithm

Based on the conjecture described previously, a dynamic load balancing policy that can achieve global fairness in an approximate sense is introduced: the decision on whether or not to re-assign a user to another port is made each time a user is served (transmits/receives a packet) by a local scheduler. Once a user has been served, its load and quantum are updated based on equations (4.5) and (4.6). If the following conditions are satisfied, user i is prompted to move from its current port k to another port r where it will be added to the end of the user list and to wait for its turn to be served:

- port k is excessively loaded ($\sum_{j \in U_k} l_j > 1$);
- port r is less loaded than port k, and user i's load is less than the difference between the aggregate load of port k and that of port r: i.e., $\sum_{j \in U_k} l_j - \sum_{j \in U_r} l_j > l_i$.

Although UL and DL are managed independently by a pair of SRR schedulers at each port, load balancing cannot be applied to UL and DL independently *unless* the set of UL users and that of DL users are disjointed. In other words, re-assigning users to balance one link (say UL) may cause the other link (say DL) unbalanced. In order to remedy this problem, load balancing is applied *alternately* to UL and DL. That is, each time a user is served, a load balancing decision is made on the alternate link. The imbalance of one link induced by load balancing of the other link can be alleviated utilizing the compensation scheme.

4.4.4 Uplink versus Downlink Scheduling

On a DL link, the only transmitter of data packets is a switch port, where a local scheduler resides. Thus there is no need to deliver scheduling information to users. However, on a UL link, multiple users contend to transmit packets to a switch port. Therefore, a mechanism to deliver scheduling information to users is needed. A *polling* scheme is used to disseminate scheduling information: scheduling information is delivered in the polling address field (`polling addr`) of the payload part of acknowledgment packet (see Figure 3.3(b)). The polling address is the address of the user who is scheduled for the next transmission after the acknowledgment.

4.5 Simulation

The performance of the dynamic load balancing algorithm is compared to that of the Least-Loaded-First (LLF) user assignment by means of simulations. For the duration of observation interval, it is assumed that all N users are active and no additional user joins the network.[7] The LLF user assignment is done at the beginning of the simulation. The SRR algorithm is used for the local scheduling for the LLF user assignment.

A four-port wireless switch with 20 users in the network is used for simulations: seven uni-directional UL users, seven uni-directional DL users, and six bi-directional users. The link rate for a UL/DL pair is the same and randomly chosen from the set $\{6, 9, 12, 18, 24, 36, 48, 54\,\text{Mbps}\}$. The link rate is assumed to be fixed during the simulation. Each user's data rate is known to the switch a priori and is used as the weight for scheduling (see Equation (4.1)). The arrival rate of data is either a *Poisson* process or constant. The data arrival rate for each user is chosen randomly so that the load would be between 0.1 and 0.45.

As observed in Table 3.2, the channel efficiency drops off when shorter packets are transmitted. The level of degradation depends on the packet length *in time* since overheads are not affected by the link rate and fixed in time. Thus, in order to keep the channel efficiency consistent regardless of the link rate, the

[7] As it is assumed that the set of active users does not change, the observation interval can be regarded as one *scheduling period*(see Section 4.2). To extend the observation interval for the purpose of improving fairness (see Section 4.5.2), multiple *scheduling periods* can be considered as a single observation interval rather than having the scheduling period itself extended and users wait for until the end of the extended *scheduling period.* This can help minimize admission delay.

packet length is set to $368\mu s$, as in Section 3.2.2. The port switching overhead [70] is assumed to be $130\mu s$, which represents the amount of time required to stabilize the RF front-end when the user is assigned to another port.

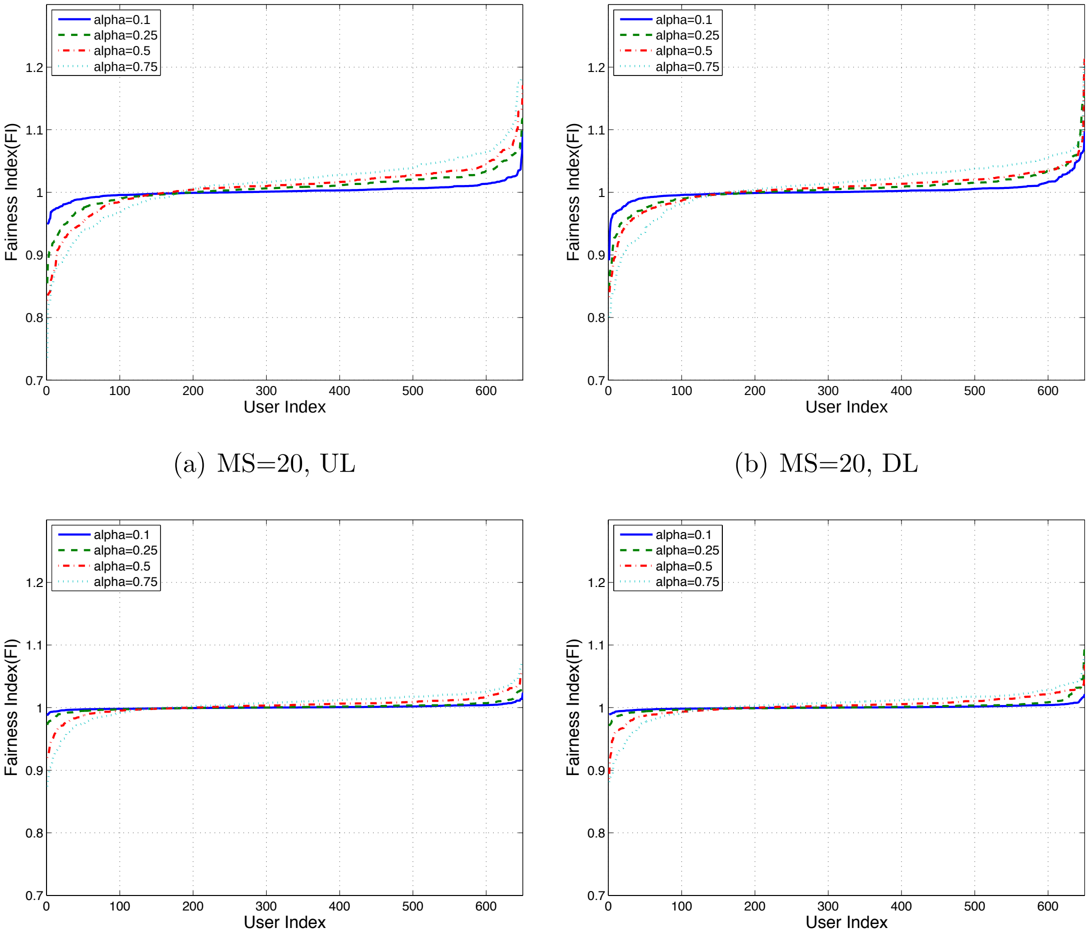
The scaling coefficient, $\bar{q}$, in (4.3), which represents the average time per round is chosen to be $\bar{q} = 368$ us, which is the same as a packet duration. In addition, two more simulation parameters are considered: (1) *Convergence coefficient* α, which adjusts the effect of skew on the corresponding group's modified aggregate load; (2) *MinimumServiceBeforeSwitching* (`MS`, in short), which is the minimum residence (in terms of service count) required in a group before it is allowed switching. These parameters are described below in detail.

4.5.1 Simulation parameter: *Convergence coefficient* (α)

Simulations are performed for $\alpha \in \{ 0.1, 0.25, 0.5, 0.75 \}$ and iterated 50 times with one second per run. The fairness indices (FIs) for 13 active users over 50 iterations of simulation (total of 650 instances) are sorted and plotted in Figure 4.6. The convergence coefficient adjusts how the skew, $\bar{\rho}^{(j)} - \rho_{min}$, affects user j's modified load. A larger value leads to a fast convergence although it may make a skew dynamics unstable due to potential over-compensation. On the other hand, a smaller value of α guarantees stability albeit with a slower convergence. As shown in the Figure 4.6, when $\alpha = 0.1$, FIs becomes more evenly distributed. It is observed that as α becomes smaller (down to ≈ 0.05), the convergence speed is drastically reduced (but with a negligible improvement to FIs).

4.5.2 Simulation parameter: *MinimumServiceBeforeSwitching*

To determine a good `MS` value, simulations are performed for the set of $\{ 1, 10, 20 \}$. The FIs are plotted in Figure 4.7 in the same way as before. The minimum value of δ for global fairness in an approximate sense (in Definition 5) for each case is calculated and given in Table 4.2. The simulation time is one second and convergence coefficient, α, is chosen to be 0.1. As seen in the simulation



(a) MS=20, UL

(b) MS=20, DL

(c) MS=10, UL

(d) MS=10, DL

Figure 4.6: Fairness Index versus convergence coefficients (50 iterations of simulation × 13 active users, sorted).

results, a smaller value of MS yields better fairness indices (and δ), most prominent when users are backlogged. When MS is small, re-assigning users is more frequent, which makes balancing load is easier. However, when MS is large, for example, when MS=20, users need to stay in one group for at least 20 quanta, balancing loads is more difficult as shown in the Figure 4.7.

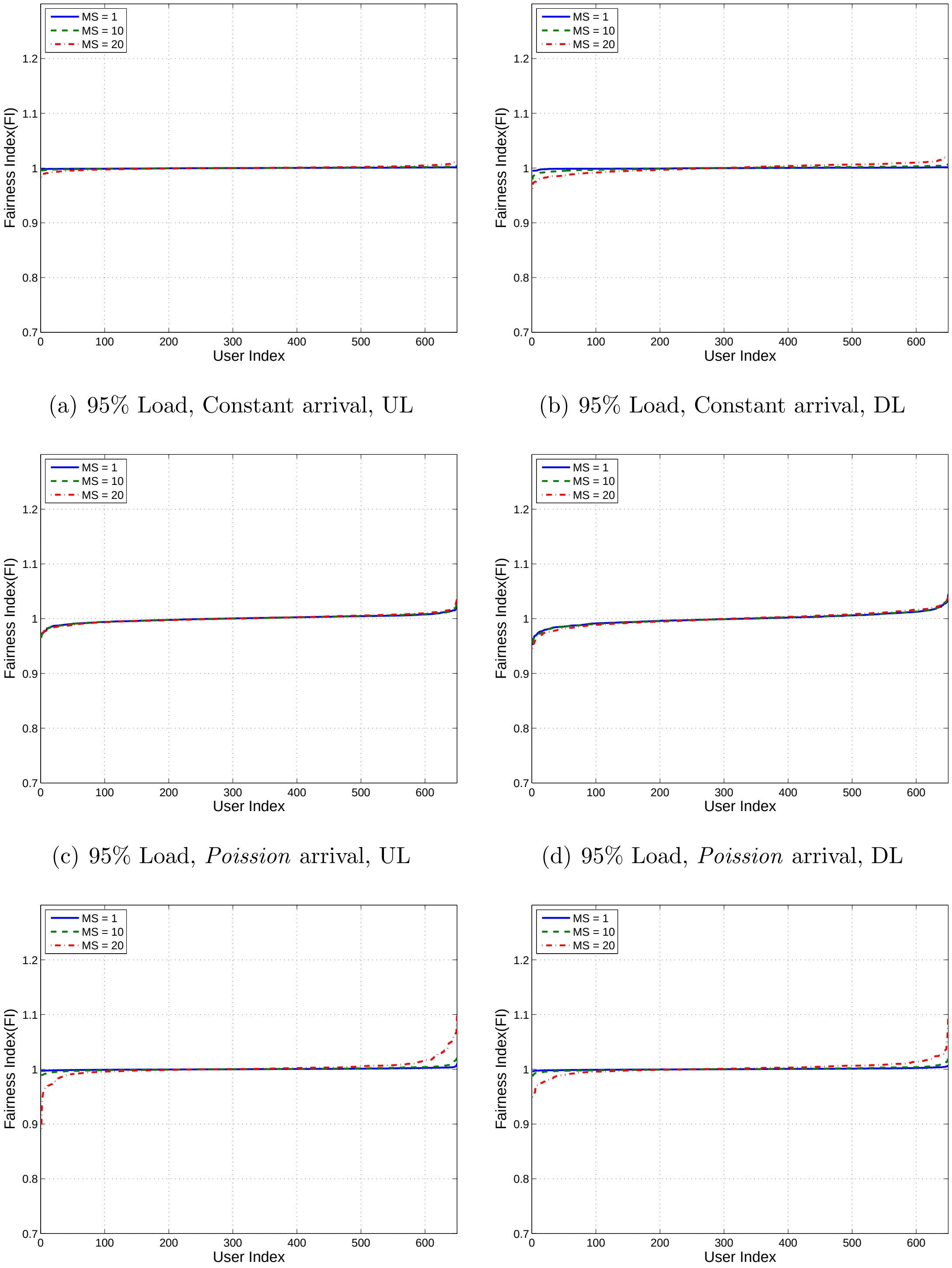
On the other hand, smaller MS causes frequent re-assigning users from one group to another. Since re-assigning users requires switching RF channels (or bands), it leads to a higher power consumption in RF front-end. Therefore, a trade-off should be made depending on fairness and power consumption requirement of the system. Table 4.1 shows the number of channel switching during one second of operation for different values of MS. When MS = 20, users experience 10 $\sim$ 14 times fewer switching on average compared to when MS = 1 (at the cost of fairness, of course) (Table 4.2, Figure 4.7). This impairment of fairness can be improved by increasing the observation interval. Figure 4.8 shows that FIs are improved as the interval increases. Corresponding δ_{min}'s are given in Table 4.3.

Table 4.1: Number of average switching (per link, one second period).

	MS = 1	MS =10	MS = 20
95% Load (Const.)	297.0	58.0	27.6
95% Load (*Poisson*)	299.7	60.5	30.1
Backlogged	274.3	46.6	20.9

4.5.3 Main results

Based on the simulation results from the previous subsections, the parameter values are chosen as follows: MS=20 and $\alpha = 0.1$. Simulations are performed to compare the fairness of the proposed scheduler and that of the LLF user as-



(a) 95% Load, Constant arrival, UL

(b) 95% Load, Constant arrival, DL

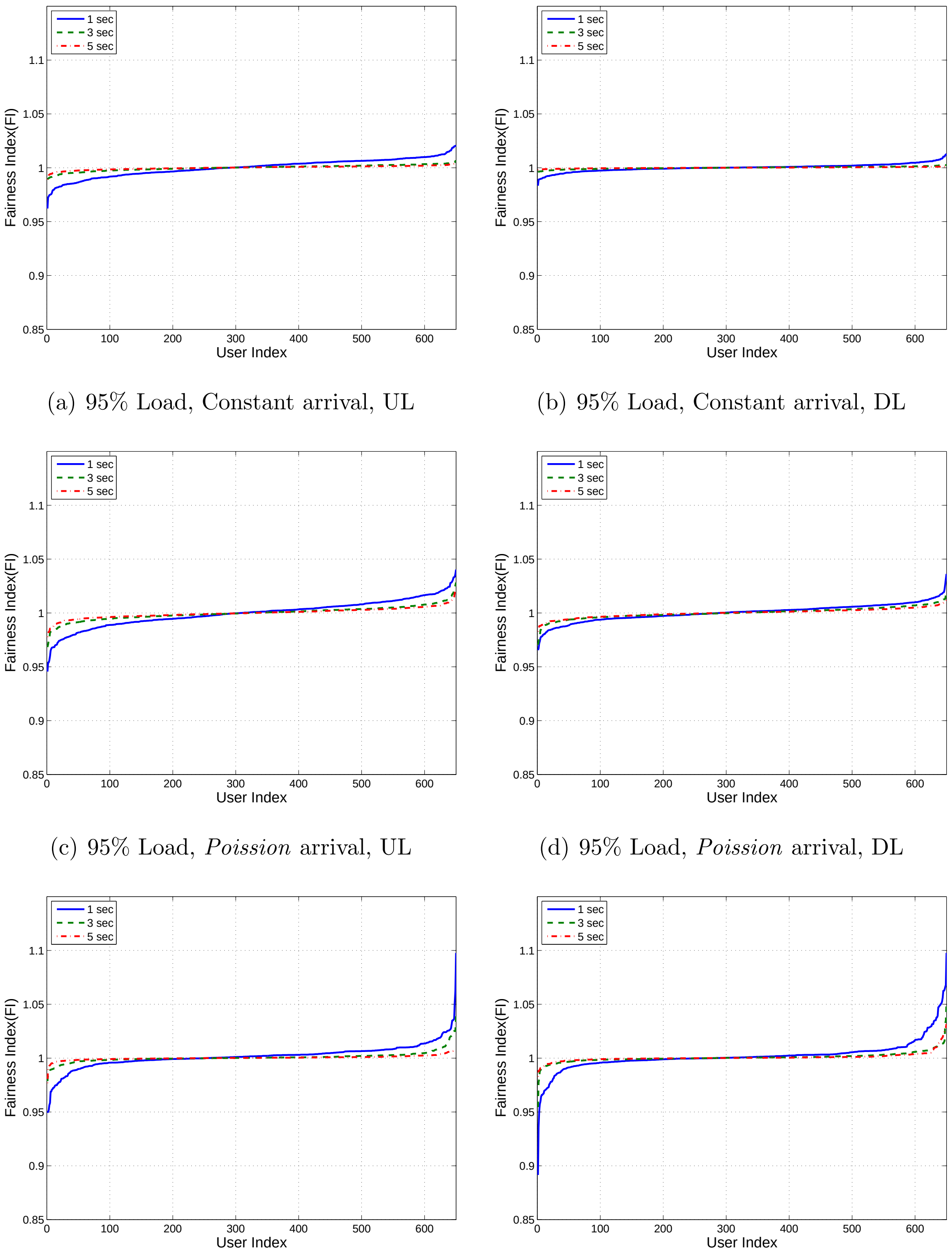
(c) 95% Load, *Poission* arrival, UL

(d) 95% Load, *Poission* arrival, DL

(e) Backlogged, UL

(f) Backlogged, DL

Figure 4.7: Fairness Index versus *MinimumServiceBeforeSwitching* (50 iterations of simulation × 13 active users, sorted).



(a) 95% Load, Constant arrival, UL

(b) 95% Load, Constant arrival, DL

(c) 95% Load, *Poission* arrival, UL

(d) 95% Load, *Poission* arrival, DL

(e) Backlogged, UL

(f) Backlogged, DL

Figure 4.8: Fairness Index versus Simulation time (50 iterations of simulation × 13 active users, sorted).

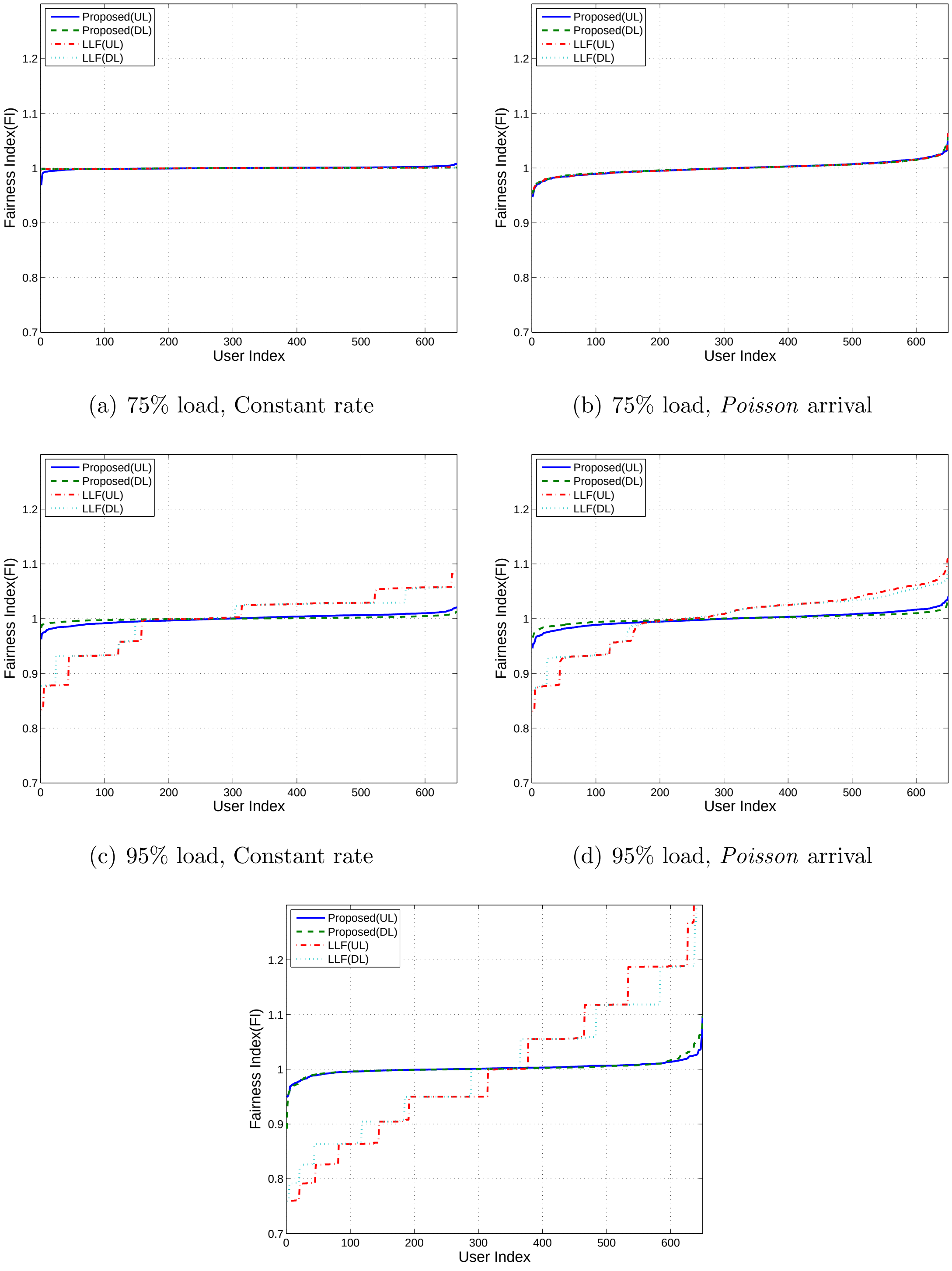
Table 4.2: $\delta_{min} = \max_{i \neq k \in U} (|\mathtt{FI}_i - \mathtt{FI}_k|)$.

(UL / DL)	MS = 1	MS =10	MS = 20
95% Load (Const.)	0.0067 / 0.0035	0.0275 / 0.0086	0.0583 / 0.0295
95% Load (*Poisson*)	0.0926 / 0.0589	0.0905 / 0.0647	0.0947 / 0.0708
Backlogged	0.0100 / 0.0139	0.0369 / 0.0325	0.1483 / 0.2062

Table 4.3: $\delta_{min} = \max_{i \neq k \in U} (|\mathtt{FI}_i - \mathtt{FI}_k|)$ ($\mathtt{MS}$ = 20).

(UL / DL)	t = 1 sec	t = 3 sec	t = 5 sec
95% Load (Const.)	0.0583 / 0.0295	0.0168 / 0.0064	0.0150 / 0.0037
95% Load (*Poisson*)	0.0947 / 0.0708	0.0600 / 0.0476	0.0436 / 0.0289
Backlogged	0.1483 / 0.2062	0.0620 / 0.0986	0.0315 / 0.0489

signment for three cases: (1) the switch is 75% loaded; (2) the switch is 95% loaded; (3) all users are backlogged. As before simulations are iterated 50 times with one second per run. The fairness indices (FIs) for 13 active users over 50 iterations of simulation are sorted and plotted in Figure 4.9; corresponding δ_{min}'s are calculated and given in Table 4.4. As shown in Figure 4.9(d) and 4.9(c), when the dynamic load balancing is applied, fair service is rendered to users when the switch is 95% loaded for both arrival processes: Poisson process and constant rate. More precisely, based on the minimum δ_{min} given in the Table 4.4, fairness in an approximate sense with $\delta_{min} = 0.095$ and $\delta_{min} = 0.058$ is achieved respectively; Furthermore, it can be improved with a lager observation window as discussed in Section 4.5.2. In contrast, the LLF scheme does not guarantee fairness. For the backlogged case shown in Figure 4.9(e), the LLF scheme yields significantly less fairness. Under lightly loaded conditions such as 75% load, when every port is underloaded (due to the fine granularity of each user's load used for this simulation $(0.1 \sim 0.45)$), the LLF scheme offers the same degree of fairness as the dynamic



(a) 75% load, Constant rate

(b) 75% load, *Poisson* arrival

(c) 95% load, Constant rate

(d) 95% load, *Poisson* arrival

(e) Backlogged

Figure 4.9: Fairness comparison between dynamic load balancing and LLF (50 iterations of simulation × 13 active users, sorted).

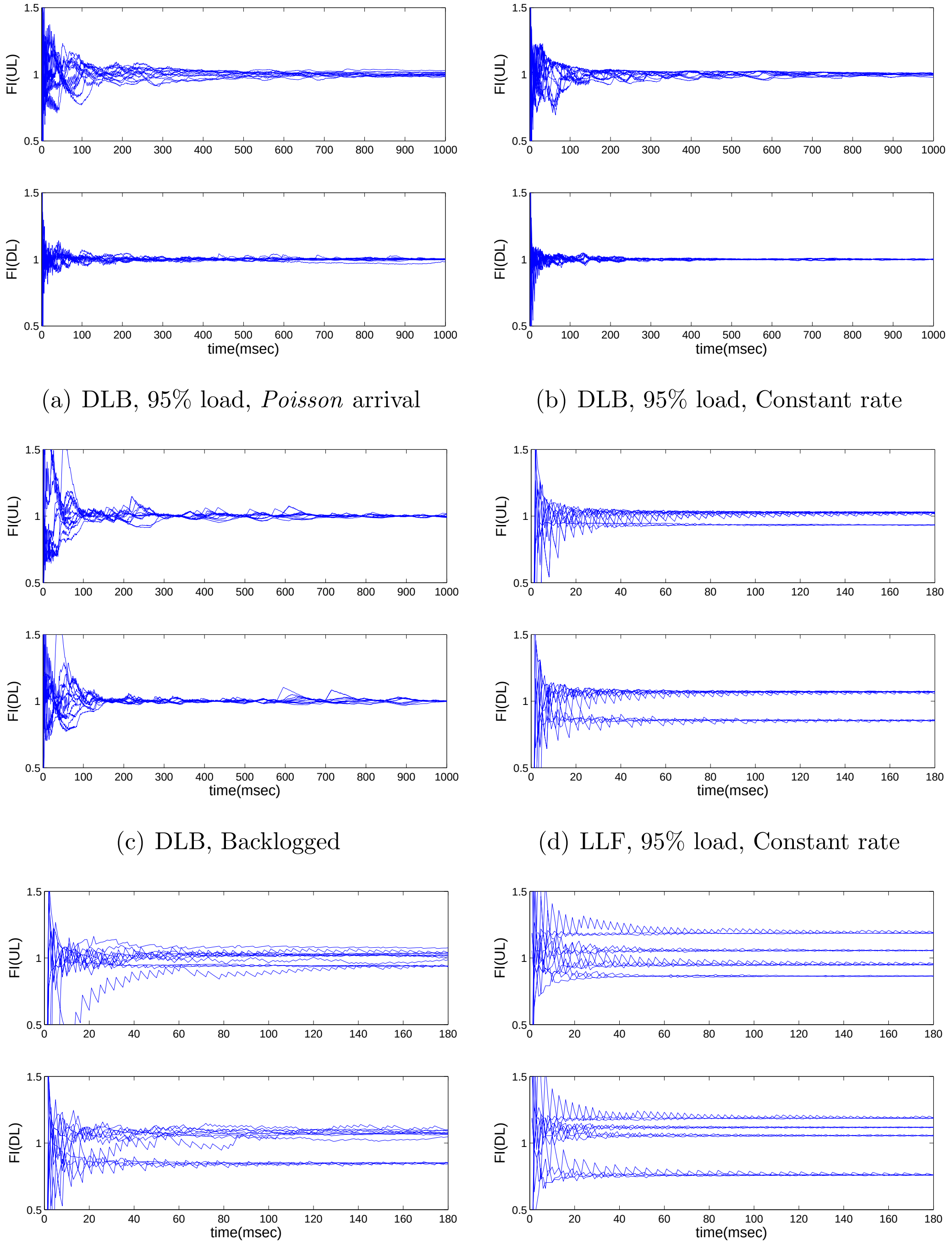
Table 4.4: $\delta_{min} = \max_{i \neq k \in U} \left(|\mathtt{FI}_i - \mathtt{FI}_k| \right)$ ($\mathtt{MS} = 20$).

(UL / DL)	DLB	LLF
75% Load (Const.)	0.03933 / 0.00295	0.00627 / 0.00270
75% Load (*Poisson*)	0.11666 / 0.10296	0.11039 / 0.10318
95% Load (Const.)	0.05826 / 0.02952	0.25504 / 0.18057
95% Load (*Poisson*)	0.09469 / 0.07079	0.27944 / 0.22401
Backlogged	0.14834 / 0.20622	0.70228 / 0.59886

load balancing algorithm.

To compare the convergence performance of the two schedulers, FIs of users are plotted as functions of time in Figure 4.10 (one run out of 50 iterations is selected for each of the six cases). FIs of 13 active users are overlaid in each plot. When the dynamic load balancing (DLB) scheduler is used, FIs of most users converge within 200 $\sim$ 350 ms, regardless of the arrival process. Note that the speed of convergence is a function of MS. When $\mathtt{MS} = 1$, convergence is achieved within 100 ms. On the other hand, with the LLF scheme, since the loads are unevenly distributed across ports, FIs converge to different values, depending on the aggregate loads of the ports with which they are associated. With the 95% load, the LLF scheme causes three ports to be underloaded and one port overloaded in both UL and DL. Therefore FIs converge to two distinct values (see Figures 4.10(e) and 4.10(d)): greater than 1 when underloaded and less than 1 when overloaded. When users are backlogged, as shown in Figure 4.10(f), FIs converge to 4 different values because an aggregate load is different on every port.

Finally, the LLF scheme utilizes the channel inefficiently due to the load imbalance. Table 4.5 shows the differences in channel utilization between the DLB and the LLF. The DLB fully utilizes the channel (very close to 95%) whereas the LLF scheme wastes around 4.5%. This discrepancy would be larger, if the load has a coarser granularity, because the resulting degree of load imbalance would be



(a) DLB, 95% load, *Poisson* arrival

(b) DLB, 95% load, Constant rate

(c) DLB, Backlogged

(d) LLF, 95% load, Constant rate

(e) LLF, 95% load, *Poisson* arrival

(f) LLF, Backlogged

Figure 4.10: Convergence of fairness: the FI trace of 13 active users are overlaid in each plot.

Table 4.5: Channel utilization (average of UL and DL), 95% load.

Scheduling	*Poisson* arrival	Constant rate
DLB	94.7%	94.3%
LLF	90.6%	90.3%

greater for the LLF.

4.6 Conclusion

This chapter described a scheduling scheme in which an SRR algorithm is used for local scheduling and a dynamic load balancing coupled with a compensation scheme for global scheduling which ensure global fairness across ports. A detailed analysis showed that the compensation scheme successfully compensates for the lagging service of users served by the same port. The proposed load balancing scheme was compared to the LLF scheme. Simulation results show that the proposed scheme, unlike the LLF, achieves fairness without a degradation in channel utilization.

Acknowledgements

The work in this chapter, in part, is a reprint of the following paper: J. Shim, K. Yun, and R. Cruz, *An Efficient Wireless Switching Architecture*, The Ninth Annual Wireless Telecommunication Symposium, Tampa, USA, Apr. 2010 [3]. The dissertation author is the primary author of this paper.

Chapter 5

Conclusion

5.1 Summary and Impact of Current Work

This dissertation presents a multi-channel wireless switching system that expands the overall capacity of the wireless network. Thorough investigations on the practical multi-channel wireless networks produced a crucial observation. That is, wireless nodes equipped with multiple radios, even when they are operating on distinct, non-overlapping channels, may suffer from *self-interference* problems if channels are adjacent. To remedy this problem, the FDD is offered as a first-cut solution; a channel sharing scheme (CS-CLP) is proposed to improve the channel efficiency. A channel efficiency of $\approx 75\,\%$ is observed from analyses and simulations when users are backlogged and the packet size is fixed. A K-port wireless switch proposed in this dissertation with each port providing 20 MHz bidirectional bandwidth yields about $2K$ times as high capacity as a wireless access point with 20 MHz per channel.

As a physical layer solution to alleviate the effect of the adjacent channel interference and to guarantee reliable synchronization, a low SINR synchronization system with DSSS modulation is presented. A prototype system is implemented on a testbed and then experiments are conducted to show the performance of the implemented system. The experimental results show that the prototype is able to correctly synchronize more than 95% of the received packets at -102 dBm.

Finally, this dissertation presents a scheduling scheme with automatic load balancing to provide fair service to a set of users served by multiple ports. In this scheduling scheme, an SRR-based algorithm is used for local scheduling and a dynamic load balancing coupled with a compensation scheme is added to ensure global fairness across ports. The proposed load balancing scheme is compared to the LLF scheme. Simulation results show that the proposed scheme, unlike the LLF, achieves fairness without degradation in channel utilization, even when user loads cannot be evenly divided into K ports. With this scheduling algorithm in conjunction with the CS-CLP, the proposed wireless switching system can achieve an overall channel utilization efficiency of $\approx 75\,\%$.

In conclusion, the higher bandwidth demand in today's local wireless networks system can be satisfied by the wireless switching system equipped with the proposed multi-channel wireless switching architecture: The potential interference due to the geographical proximity of adjacent channels in the same network is tackled by the proposed physical layer solutions; fairness issues for multiple users sharing the switch, which is a typical problem in multi-server scheduling system, is resolved by the dynamic load balancing algorithm.

5.2 Future Work

This dissertation focused mainly on the essential functionalities that characterize the proposed wireless switching system. Additional functionalities are needed for a complete wireless switching system; and further analytical work would help solidify the premise fo this research as well. The detailed descriptions are listed below:

- OFDM based physical layer: Chapter 2 presented a physical layer solution for the wireless switch based on a DSSS modulation, which has a good noise and interference immunity. Since, however, the proposed wireless switching system architecture is independent of physical layer modulation schemes, the OFDM modulation, which is a popular scheme for the current and the next generation wireless communication systems due to its promise for a higher data rate communication, should be considered.

- MAC layer protocol: Chapter 3 presented only a part of the MAC layer protocol to support the operation of the scheduler. Many missing functionalities, such as polling scheme, power management and user on-/off-line management, and relevant packet formats, need to be more thoroughly investigated.

- Optimal load balancing policy: In Chapter 4, it was analytically shown that the compensation scheme successfully compensates lagging services of users served by the same port and the dynamic load balancing ensures global fairness across ports. However, the optimality of the dynamic load balancing algorithm is still a open problem to investigate.

- Prototyping wireless switching system: the synchronization part of wireless communication system was implemented on a prototyping platform and tested. Remaining implementations that would allow a more complete verification of the proposed architecture are: (i) an implementation of the rest of physical layer functional blocks related to channel coding and source coding as well as implementation of the CS-CLP will complete the hardware part of the system; (ii) a software implementation of the MAC and upper layer protocols on both the wireless switch and the user nodes; (iii) the software implementation of the scheduling algorithm on the wireless switch.

Bibliography

1. J. Shim, M. Amde, K. Yun, and R. Cruz, "Synchronization at low sinr in asynchronous direct-sequence spread-spectrum communications," *Proceedings of the Second International Conference on Systems and Networks Communications*, Jan 2007.

2. M. Amde, J. Shim, J. Marciano, K. Yun, and R. Cruz, "A low sinr synchronization system for direct-sequence spread-spectrum communications: radio prototype, verification testbed, and experimental eesults," *Proceedings of the 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, Jan 2008.

3. J. Shim, K. Yun, and R. Cruz, "An efficient wireless switching architecture," *Proceedings of the Ninth Annual Wireless Telecommunication Symposium*, Apr 2010.

4. R. Gallager, "Low-density parity-check codes," *Information Theory*, Jan 1962.

5. C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes." *IEEE International Conference on Communications*, Jan 1993.

6. A. Paulraj, R. Nabar, and D. Gore, *Introduction to space-time wireless communications.* Cambridge University Press, 2003.

7. D. Tse and P. Viswanath, *Fundamentals of wireless communication.* Cambridge University Press, 2005.

8. I. Aad, C. Castelluccia, and R. INRIA, "Differentiation mechanisms for ieee 802.11," *IEEE INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, vol. 1, 2001.

9. "IEEE standards association, IEEE 802.11, 1999 edition." [Online]. Available: http://standards.ieee.org/getieee802/802.11.html

10. T. Rappaport, K. Blankenship, and H. Xu, "Propagation and radio system design issues in mobile radio systems for the glomo project," *Virginia Polytechnic Institute and State University*, Jan 1997.

11. A. Otyakmaz, R. Schoenen, S. Dreier, and B. Walke, "Parallel operation of half-and full-duplex fdd in future multi-hop mobile radio networks," *Wireless Conference*, Jan 2008.

12. T. Liu, M. Rong, Y. Xue, L. Wang, and E. Schulz, "User partitioning based resource assignment in half-duplex fdd relaying cellular networks," *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pp. 985 – 989, Mar 2007.

13. J. Andrews, A. Ghosh, and R. Muhamed, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking.* Prentice-Hall, 2007.

14. R. L. Pickholtz, L. B. Milstein, and D. L. Schilling, "Spread spectrum for mobile communications," *IEEE Transactions on Vehicular Technology*, Jan 1991.

15. R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of spread-spectrum communications–a tutorial," *IEEE Transactions on Communications*, Jan 1982.

16. J. G. Proakis, *Digital Communications 2nd ed.* McGraw-Hill, 1989.

17. R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," *DEC Research Report TR-301*, Jan 1984.

18. M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round-robin," *IEEE/ACM Transactions on Networking (TON)*, vol. 4, no. 3, Jun 1996.

19. L. Kleinrock, *Queueing Systems: Volume 2: Computer Applications.* Wiley, 1976.

20. A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," *SIGCOMM '89*, Jan 1989.

21. A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Transactions on Networking (TON)*, vol. 1, no. 3, pp. 344 – 357, Jun 1993.

22. ——, "A generalized processor sharing approach to flow control in integrated services networks: the multiple node case," *IEEE/ACM Transactions on Networking (TON)*, vol. 2, no. 2, pp. 137 – 150, Apr 1994.

23. J. Bennett and H. Zhang;, "Wf2q: worst-case fair weighted fair queueing," *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, vol. 1, pp. 120 – 128 vol.1, Feb 1996.

24. H. Zhang. . . , "Service disciplines for guaranteed performance service in packet-switching networks," *Proceedings of the IEEE*, vol. 83, no. 10, Oct 1995.

25. S. Golestani, "A self-clocked fair queueing scheme for broadband applications," *INFOCOM '94. Networking for Global Communications. 13th Proceedings IEEE*, pp. 636 – 646 vol.2, May 1994.

26. J. Bennett and H. Zhang;, "Hierarchical packet fair queueing algorithms," *IEEE/ACM Transactions on Networking (TON)*, vol. 5, no. 5, pp. 675 – 689, Oct 1997.

27. M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," *SIGCOMM '95: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, Oct 1995.

28. H. Adiseshu, G. Parulkar, and G. Varghese, "A reliable and scalable striping protocol," *ACM SIGCOMM Computer Communication Review*, Jan 1996.

29. S. Kanhere, H. Sethu, and A. Parekh, "Fair and efficient packet scheduling using elastic round robin," *IEEE Transactions on Parallel and Distributed Systemsistributed Systems*, vol. 13, no. 3, pp. 324 – 336, Mar 2002.

30. S. Kanhere and H. Sethu, "Fair, efficient and low-latency packet scheduling using nested deficit round robin," *High Performance Switching and Routing*, Jan 2001.

31. A. Shiravi, Y. Kim, and P. Min, "Proportional nested deficit round robin: Improving the latency of packet scheduler with an o(1) complexity," *First International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications, 2005. (AAA-IDEA 2005)*, pp. 34–41, 2005.

32. Y. Zhang and P. Harrison, "Performance of a priority-weighted round robin mechanism for differentiated service networks," *Computer Communications and Networks*, Jan 2007.

33. Y. Zhou, M. Hosaagrahara, and H. Sethu, "Opportunity-based deficit round robin: a novel packet scheduling strategy for wireless networks," *High Performance Switching and Routing*, Jan 2002.

34. S. Mohanty and L. Bhuyan, "Fair scheduling over multiple servers with flow-dependent server rate," *Local Computer Networks*, Jan 2006.

35. D. Raz, B. Avi-Itzhak, and H. Levy, "Fairness considerations of scheduling in multi-server and multi-queue systems," *Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*, 2006.

36. H. XIAO and Y. JIANG, "Analysis of multi-server round robin scheduling disciplines," *IEICE TRANSACTIONS on Communications*, vol. E87-B, no. 12, Dec 2004.

37. A. Ganti, E. Modiano, and J. Tsitsiklis, "Optimal transmission scheduling in symmetric communication models with intermittent connectivity," *IEEE Transactions on Information Theory*, vol. 53, no. 3, pp. 998 – 1008, Mar 2007.

38. S. Kittipiyakul and T. Javidi, "Delay-optimal server allocation in multiqueue multiserver systems with time-varying connectivities," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2319 – 2333, May 2009.

39. S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Transactions on Networking (TON)*, Jan 1999.

40. T. Nandagopal, S. Lu, and V. Bharghavan, "A unified architecture for the design and evaluation of wireless fair queueing algorithms," *Wireless Networks*, Jan 2002.

41. Y. Bejerano, S.-J. Han, and L. Li, "Fairness and load balancing in wireless lans using association control," *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, Sep 2004.

42. R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of spread-spectrum comunications - a tutorial," *IEEE Transactions on Communications*, vol. COM-30, no. 5, pp. 855–884, May 1982.

43. R. Price and P. Green, "A communication technique for multipath channels," *Proceedings of the IRE*, Jan 1958.

44. A. J. Viterbi, *Code Division Mulple Access - Princibles of Spread Spectrum Communication.* Addison-Wesley, 1995.

45. H. Meyr and G. Ascheid, *Synchronization in Digital Communications. Volume 1 : Phase-, Frequency-Locked Loops, and Amplitude Control.* Wiley-Interscience, 1989.

46. T. Rappaport, K. Blankenship, and H. Xu, *Tutorial : Propogation and Radio System Design Issues in Mobile Radio Systems for the GloMo Project.* Virginia Tech, 1997.

47. S. W. Golomb and G. Gong, *Signal Design for Good Correlation: For Wireless Communication, Cryptography and Radar.* Cambridge University Press, 2005.

48. D. Sarwate and M. Pursley, "Crosscorrelation properties of pseudorandom and related sequences," *Proceeding of the IEEE*, vol. 68, no. 5, pp. 593–619, May 1980.

49. H. L. V. Trees, *Detection, Estimation, and Modulation Theory: Detection, estimation, and linear modulation theory, Part I.* John Wiley and Sons, 2001.

50. M. Amde, J. Marciano, R. Cruz, and K. Yun, "Code acquisition at low SINR in spread spectrum communications," in *ISSSTA '06: The 9th International Symposium on Spread Spectrum Techniques and Applications*, Manaus, Brazil, Aug. 2006.

51. E. Grayver, J. Frigon, A. M. Eltawil, A. Tarighat, K. Shoarinejad, A. Abbasfar, D. Cabric, and B. Daneshrad, "Design and VLSI implementation for a WCDMA multipath searcher," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 889–902, 2005.

52. L.-L. Yang and L. Hanzo, "Serial acquisition of DS-CDMA signals in multipath fading mobile channels," *IEEE Transactions on Vehcular Technology*, vol. 50, no. 2, pp. 617–628, Mar. 2001.

53. O. Shin and K. B. Lee, "Utilization of multipaths for spread-spectrum code acquisition in frequency-selective rayleigh fading channels," *IEEE Transactions on Communications*, vol. 49, no. 4, pp. 734–743, Apr. 2001.

54. S. H. Won and Y. J. Kim, "Performance analysis of multi-path searcher for mobile station in W-CDMA system employing transmit diversity," *Electronics Letters*, vol. 39, no. 1, pp. 137–139, Jan. 2003.

55. E. Sourour and S. C. Gupta, "Direct-sequence spread-spectrum parallel acquisition in nonselective and frequency-selective rician fading channels," *IEEE Journal on Selectec Areas in Communications*, vol. 10, no. 3, pp. 535–544, Apr. 1992.

56. R. R. Rick and L. B. Milstein, "Optimal decision strategies for acquisition of spread-spectrum signals in frequency-selective fading channels," *IEEE Transactions on Communications*, vol. 46, no. 5, pp. 686–694, May 1998.

57. A. Polydoros and C. L. Weber, "Analysis and optimization of correlative code-tracking loops in spread-spectrum systems," *IEEE Transactions on Communications*, vol. COM-33, no. 1, pp. 30–43, Jan. 1985.

58. G. Fock and J. B. et al, "Channel tracking for rake receivers in closely spaced multipath environment," *IEEE Journal on Seleted Areas in Communications*, vol. 19, no. 12, pp. 2420–2431, Dec. 2001.

59. A. M. Eltawil and B. Daneshrad, "A low power DS-CDMA RAKE receiver utilizing resource allocation techniques," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 8, pp. 1321–1330, Aug. 2004.

60. W.-H. Sheen and C.-H. Tai, "A noncoherent tracking loop with diversity and multipath interference cancellation for direct-seqeunce spread-spectrum systems," *IEEE Transactions on Communications*, vol. 46, no. 11, pp. 1516–1524, Nov. 1998.

61. M. K. Simon and M.-S. Alouini, *Digital Communication over Fading Channels.* Wiley, 2005.

62. Xilinx Virtex-4. [Online]. Available: http://www.xilinx.com/virtex4

63. Agilent N5110B Baseband Studio for Waveform Capture and Playback : Technical Overview. [Online]. Available: http://cp.literature.agilent.com/litweb/pdf/5989-2095EN.pdf

64. N5102A Baseband Studio Digital Signal Interface Module : Technical Overview. [Online]. Available: http://cp.literature.agilent.com/litweb/pdf/5988-9495EN.pdf

65. Agilent E4438C ESG Vector Signal Generator : Data Sheet. [Online]. Available: http://cp.literature.agilent.com/litweb/pdf/5988-4039EN.pdf

66. Cisco Aironet 802.11a/b/g Wireless CardBus Adapter. [Online]. Available: http://www.cisco.com/en/US/products/hw/wireless/ps4555/products_data_sheet09186a00801ebc29.html

67. A. R. S. Bahai, B. R. Saltzberg, and M. Ergen, *Multi-carrier digital communications: theory and applications of OFDM.* Springer, 2004.

68. O. Kremien and J. Kramer, "Methodical analysis of adaptive load sharing algorithms," *IEEE Transactions on Parallel and Distributed Systems*, Jan 1992.

69. Khalighi, J. Brossier, G. Jourdain, and K. Raoof, "Water filling capacity of rayleigh mimo channels," *PIMRC 01*, Sep 2001.

70. "MAX2829 dual-band 802.11a/b/g gransceiver." [Online]. Available: http://datasheets.maxim-ic.com/en/ds/MAX2828-MAX2829.pdf