Syracuse University

SURFACE

Dissertations - ALL SURFACE

December 2015

Design of a High Capacity, Scalable, and Green Wireless Communication System Leveraging the Unlicensed Spectrum

Mohammad Riaz Khawer
Syracuse University

Follow this and additional works at: https://surface.syr.edu/etd

 Part of the Engineering Commons

Recommended Citation

Khawer, Mohammad Riaz, "Design of a High Capacity, Scalable, and Green Wireless Communication System Leveraging the Unlicensed Spectrum" (2015). *Dissertations - ALL*. 393.
https://surface.syr.edu/etd/393

This Dissertation is brought to you for free and open access by the SURFACE at SURFACE. It has been accepted for inclusion in Dissertations - ALL by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

Abstract

The stunning demand for mobile wireless data that has been recently growing at an exponential rate requires a several fold increase in spectrum. The use of unlicensed spectrum is thus critically needed to aid the existing licensed spectrum to meet such a huge mobile wireless data traffic growth demand in a cost effective manner. The deployment of Long Term Evolution (LTE) in the unlicensed spectrum (LTE-U) has recently been gaining significant industry momentum. The lower transmit power regulation of the unlicensed spectrum makes LTE deployment in the unlicensed spectrum suitable only for a small cell. A small cell utilizing LTE-L (LTE in licensed spectrum), and LTE-U (LTE in unlicensed spectrum) will therefore significantly reduce the total cost of ownership (TCO) of a small cell, while providing the additional mobile wireless data offload capacity from Macro Cell to small cell in LTE Heterogeneous Networks (HetNet), to meet such an increase in wireless data demand. The U.S. 5 GHz Unlicensed National Information Infrastructure (U-NII) bands that are currently under consideration for LTE deployment in the unlicensed spectrum contain only a limited number of 20 MHZ channels. Thus in a dense multi-operator deployment scenario, one or more LTE-U small cells have to co-exist and share the same 20 MHz unlicensed channel with each other and with the incumbent Wi-Fi.

This dissertation presents a proactive small cell interference mitigation strategy for improving the spectral efficiency of LTE networks in the unlicensed spectrum. It describes the scenario and demonstrate via simulation results, that in the absence of an explicit interference mitigation

mechanism, there will be a significant degradation in the overall LTE-U system performance for LTE-U co-channel co-existence in countries such as U.S. that do not mandate Listen-Before-Talk (LBT) regulations. An unlicensed spectrum Inter Cell Interference Coordination (usICIC) mechanism is then presented as a time-domain multiplexing technique for interference mitigation for the sharing of an unlicensed channel by multi-operator LTE-U small cells. Through extensive simulation results, it is demonstrated that the proposed usICIC mechanism will result in 40% or more improvement in the overall LTE-U system performance (throughput) leading to increased wireless communication system capacity.

The ever increasing demand for mobile wireless data is also resulting in a dramatic expansion of wireless network infrastructure by all service providers resulting in significant escalation in energy consumption by the wireless networks. This not only has an impact on the recurring operational expanse (OPEX) for the service providers, but importantly the resulting increase in greenhouse gas emission is not good for the environment. Energy efficiency has thus become one of the critical tenets in the design and deployment of Green wireless communication systems. Consequently the market trend for next-generation communication systems has been towards miniaturization to meet this stunning ever increasing demand for mobile wireless data, leading towards the need for scalable distributed and parallel processing system architecture that is energy efficient, and high capacity. Reducing cost and size while increasing capacity, ensuring scalability, and achieving energy efficiency requires several design paradigm shifts.

This dissertation presents the design for a next generation wireless communication system that employs new energy efficient distributed and parallel processing system architecture to achieve these goals while leveraging the unlicensed spectrum to significantly increase (by a factor of two) the capacity of the wireless communication system. This design not only significantly reduces the upfront CAPEX, but also the recurring OPEX for the service providers to maintain their next generation wireless communication networks.

Design of a High Capacity, Scalable, and Green Wireless Communication System Leveraging the Unlicensed Spectrum

By

Mohammad R. Khawer

B.E. N.E.D. University of Engineering & Technology, Karachi, Pakistan, 1992

M.S. Syracuse University, New York, U.S., 1995

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer and Information Science and Engineering (CISE)

Syracuse University

December 2015

Copyright © Mohammad R. Khawer 2015
All Rights Reserved

“If you can dream - and not make dreams your master;

If you can think - and not make thoughts your aim;

If you can meet with Triumph and Disaster

And treat those two impostors just the same;”

— **Rudyard Kipling, “A Father’s Advice to his Son”**

To my family, and the loving memory of my father, late Dr. Mohammad Ilyas Akhter, who taught and inspired me to remain steadfast in the face of adversity

Acknowledgement

I would like to express my deepest and everlasting gratitude to all my teachers and mentors who were extremely kind and generous with their time and knowledge, without which I would not be where I am today.

To my advisor Professor Jian Tang, who continuously provided me with guidance and support, my greatest appreciation and thanks for his thought provoking insights and guidance that helped me tremendously in completing this dissertation work.

My deepest appreciation also goes to Professor Chilukuri K. Mohan (former Department Chair of EECS), and Dr. Nancy Cantor (former Chancellor of Syracuse University) for their help, and guidance in accomplishing this milestone.

I would like to thank my dissertation & research committee for their time, effort, consideration, and support in this entire endeavor.

I would also like to acknowledge the following Alcatel-Lucent colleagues for their guidance and support: Feng Han, Mugur Abulius, Lina So, and Shriram K. Easwaran.

My final and deepest gratitude goes to my parents, wife, kids, and siblings who stood by me with their love and encouragement without which nothing would have been possible. I dedicate this dissertation to you all.

Contents

Abstract i
Copyright v
Acknowledgement vii
List of Figures x
1 Introduction 11
1.1 Background and Motivations 11
1.2 Related Work and Literature Gap 13
1.3 Contributions and Claims of Originality 20
1.4 Dissertation Outline 24
2 Interference Mitigation for the LTE Networks in the Unlicensed Spectrum 26
2.1 Introduction 26
2.2 System Model and Problem Description 29
2.3 Unlicensed Spectrum Inter Cell Interference Coordination (usICIC) Mechanism 33
2.3.1 Overview 33
2.3.2 Workflow of the Proposed usICIC Mechanism 35
2.4 Performance Evaluation 41
2.4.1 Simulation Setup 41
2.4.2 Simulation Scenarios 43
2.4.3 Comparative Analysis of the Simulation Results 45
3 High Capacity, Scalable, and Energy Efficient Design 48
3.1 Use of Open source Real Time Operating System (RTOS) 50
3.2 Use of a Multicore processor 52
3.2.1 Supervised Asymmetric Multi-Processing (S-AMP) Configuration 54
3.2.2 Symmetric Multi-Processing (SMP) Configuration 55
3.2.3 Deterministic Symmetric Multi-Processing (D-SMP) Configuration 57
3.3 Portability and Scalability Considerations 59
3.3.1 Core Abstraction Layer (CAL) 61
3.4 Application Server Recovery Mechanism 64
3.5 Energy Efficiency Considerations 67
3.5.1 Energy Efficiency as a result of significant reduction in device count 68
3.5.2 Energy Efficiency as a result of Multicore Dynamic Power Control 69
3.5.3 Energy Efficiency as a result of Disabling the LTE-U Service 75

3.6 Testing and Performance Validation Considerations 76
4 Conclusions 81
4.1 Conclusions and Summary 81
4.2 Future Research Directions 85
5 Appendix 86
5.1 Appendix A – VxWorks 86
5.2 Appendix B – Open Source Linux with PREEMPT_RT 87
5.3 Appendix C – Critical Section of Code, and software locks 89
5.4 Appendix D – P4080 BMAN Initialization 92
5.5 Appendix E – Standard P4080 DPAA configuration for IP packet processing using the Protocol Stack 95
5.5.1 Ingress Control Plane IP Packet processing 95
5.5.2 Egress Control Plane IP Packet processing 96
5.6 Appendix F – P4080 DPAA configuration for IP packet processing without the Protocol Stack [36] 98
5.6.1 Ingress Data Plane IP Packet processing 98
5.6.2 Egress Data Plane IP Packet processing 101
5.7 Appendix G – P4080 QMAN Configuration [37] 103
6 Bibliography 105
7 Abbreviations, Acronyms, and Terms 113
8 Curriculum Vitae 116

List of Figures

Figure 1: Single Operator LTE-U Deployment Scenario in a Wireless HetNet 28
Figure 2: TCSAT Gating Cycle for LTE-U Channel Sharing with Wi-Fi 30
Figure 3: Multi-Operator LTE-U Co-channel Interference Scenario 31
Figure 4: Event Trace Diagram for the proposed usICIC Mechanism 35
Figure 5: LTE-U SINR Distribution Comparison 45
Figure 6: LTE-U User Throughput Comparison 46
Figure 7: LTE-U Small Cell Average Throughput Comparison 47
Figure 8: Supervised Asymmetric Multi-Processing (S-AMP) Configuration 54
Figure 9: Symmetric Multi-Processing (SMP) Configuration 56
Figure 10: Deterministic Symmetric Multi-Processing (D-SMP) Configuration 57
Figure 11: Core Abstraction Layer Functional Architecture 62
Figure 12: DPC State machine for the dedicated data plane cores 71
Figure 13: DPC State machine for the shared control plane core 75
Figure 14: Throughput Testing Rate over a Period of Time 78
Figure 15: Processor Occupancy Load Corresponding To Throughput Rate 79
Figure 16: P4080 BMAN Buffer Pool Seeding 93
Figure 17: Ingress Control Plane IP Packet processing using Protocol Stack 96
Figure 18: Egress Control Plane IP packet Processing using the Protocol Stack 97
Figure 19: Ingress Data Plane IP Packet Processing 101
Figure 20: Egress Data Plane IP Packet Processing 103

Chapter 1

1 Introduction

1.1 Background and Motivations

The ever increasing demand for mobile wireless data requires system design that is low cost, high capacity, flexible, scalable, and energy efficient with smaller size or foot print for building the next generation wireless communication systems. Even though the demand for mobile wireless data is increasing exponentially in recent years, but it is not translating into proportional increase in revenue for the service providers. It is therefore imperative for the service providers to look for energy efficient system design solutions that are aimed at reducing the average cost per bit of mobile wireless data transmitted by leveraging the unlicensed spectrum, to reduce the upfront capital expanses (CAPEX), and the recurring operational expenses (OPEX) for the wireless communication systems.

The use of unlicensed spectrum to aid the available licensed spectrum, allow a significant reduction in the total cost of ownership (TCO) of the wireless communication system for the service provider by increasing the overall system capacity in a cost effective manner. The licensed spectrum is not only limited, but extremely costly typically of the order of multi-billion dollars for each market segment that adds significantly to the upfront CAPEX for the service provider while deploying a wireless communication network.

The actual size or foot print of the wireless communication system has a direct impact on the recurring site leasing cost for the service provider for

the location where the wireless communication system is physically mounted or deployed. The overall power consumption of the wireless communication network is another key aspect of the recurring OPEX for the service providers, as tens of thousands of wireless communication systems are deployed to meet the ever growing mobile wireless data demand. These are some key design aspects that are driving the market trend towards system miniaturization using scalable distributed and parallel processing architecture that is low cost, high capacity, and energy efficient.

This dissertation presents in chapter 2, the research work related with improving the spectral efficiency of LTE networks in the unlicensed spectrum using a proactive small cell interference mitigation mechanism. More specifically an unlicensed spectrum Inter Cell Interference Coordination (usICIC) mechanism is presented, that significantly enhance the overall LTE-U system performance, and capacity for the wireless communication system.

This dissertation also presents in chapter 3, the research work related with system design using a new scalable distributed and parallel processing architecture that significantly reduce the device count by a factor of 14 to reduce the overall power consumption of the wireless communication system while providing higher system capacity (by a factor of two) by leveraging the unlicensed spectrum. It also employs a dynamic power control mechanism to continuously modules the clock speed of the individual processing cores of the multi-core processor used in the design, throughout the day depending upon the traffic load, in an attempt to reduce the power consumption to make the system even more energy efficient. Additionally while handling low traffic demand, the LTE-U service, and the radios corresponding to the unlicensed 5

GHz band may be completely turned off by the new design to achieve even higher energy efficiency.

Thus, the research work presented in this dissertation facilitate the design for building a commercial grade high capacity, scalable, and Green wireless communication system that leverages the unlicensed spectrum to reduce the upfront CAPEX, and the recurring OPEX for the service providers.

1.2 Related Work and Literature Gap

In this section, we describe below the state of the art related work and discuss the literature gap for the two research areas presented in the dissertation:

- **Interference mitigation for the LTE networks in the unlicensed spectrum**

  The LTE and Wi-Fi coexistence in the unlicensed spectrum has been studied recently. In [6], the benefits of deploying LTE in the unlicensed spectrum and its potential co-existence with Wi-Fi were discussed. In [8], the need for proper co-existence mechanisms was explored to enable Wi-Fi, and LTE co-existence in the unlicensed spectrum. In [9], a Wi-Fi, and LTE co-existence mechanism that consisted of blanking some LTE sub-frames to improve Wi-Fi performance was discussed. In [10], the performance of LTE and Wi-Fi was compared in the unlicensed spectrum to demonstrate that in general, LTE outperforms Wi-Fi in similar scenarios. In [11], the use of LTE UL power control was presented to improve LTE and Wi-Fi coexistence. In [12], the performance of LTE and WLAN in a shared frequency band was evaluated. The study demonstrated that the

co-existence has a negative impact on WLAN system performance but it was reasoned that the severity of the impact can be controlled by restricting LTE activities. [12] Called for more careful co-existence mechanism studies to address this issue. [5] [24]-[27] provided in-depth analyses for LTE and Wi-Fi co-existence using carefully designed Carrier Sense Adaptive Transmission (CSAT) gating cycle mechanism for LTE-U transmissions in the unlicensed spectrum. These work that were done under the umbrella of the LTE-U Forum [23] demonstrated via simulation results that the CSAT mechanism will allow LTE to be a fair neighbor to Wi-Fi while sharing the same unlicensed channel by ensuring that the performance of the Wi-Fi AP is no worse than when it shares the same channel with another Wi-Fi AP.

The difference between the work presented in this dissertation and the other research works has been that the other research works have not studied and performed analysis for multi-operator LTE-U co-existence in the unlicensed spectrum while sharing the same unlicensed channel. The LTE-U Forum work [25]–[27] assumed that for the multi-operator LTE-U deployment scenario, the two LTE-U small cells will use different channels in the unlicensed spectrum. Since there are a limited number of channels in the 5 GHZ unlicensed spectrum, the probability of two LTE-U small cells sharing the same unlicensed channel in a dense multi operator deployment scenario is very high. We have demonstrated via detailed LTE system level simulations that overall system performance degrades significantly when two LTE-U Small cells share the same unlicensed channel without any CSAT gating cycle coordination. We have proposed

an unlicensed spectrum Inter Cell Interference Coordination (usICIC) mechanism to allow the multi-operator LTE-U small cells to negotiate orthogonal (non-overlapping) CSAT gating cycles for their respective LTE-U transmissions while sharing an unlicensed channel to drastically improve (by 40% or more) the overall LTE-U system performance as validated by the simulation results. This work has resulted in one IEEE publication [45], and one LTE-U Forum technical contribution [46].

Interference migration has been studied in the context of cognitive radios. In [28], the authors surveyed the fundamental capacity limits and associated transmission techniques for different wireless network design paradigms using cognitive radios. Power allocation strategies were proposed in [29], which could be employed for the coexistence of one or more cognitive radios with a primary radio where the cognitive radios transmit in a spectrum allocated to the primary radio. In [30], the authors proposed an underlay spectrum sharing policy for a point-to-point link of cognitive radios to leverage uplink spectrum resource of the primary network, and demonstrated that such an uplink band sharing mechanism could enhance the throughput performance of point-to-point link, while the interference to the primary users could be regulated under the allowed level decided by the primary users. In [47], the authors studied the green cognitive mobile networks with small cells in the smart grid environment. Unlike most existing studies on cognitive networks, where only the radio spectrum is sensed, their proposed cognitive networks sense not only the radio spectrum environment, but also the smart grid environment, based on which power allocation and interference management for multimedia

communications are performed to significantly reduce the operational expenditure and CO2 emissions.

Unlike cognitive radios, the LTE-U small cell operates in the unlicensed spectrum where there is no concept of the primary user. Upon detecting another node's (Wi-Fi or LTE-U) presence on the unlicensed channel, the LTE-U small cell does not necessarily switch its operation to another unlicensed channel, but rather attempts to co-exist with the other nodes and share the unlicensed channel in TDM fashion.

Interference mitigation has been an important area of research for the licensed spectrum as well. In [48], the authors have concluded that Inter Cell Interference Coordination (ICIC) must be effectively used to reduce the mutual interference of edge users in heterogeneous network (Het-net), where the interference scenario becomes more complicated They analyzed the limitation of traditional ICIC, and proposed a novel ICIC method based on User' position for the improvement of the downlink ICIC efficiency. In [49], the authors presented a decentralized subcarrier collision arbitrating algorithm among adjacent cells, and proved it to be distributed and collision free. Based on this algorithm, the authors proposed an ideal synchronous ICIC method and a more practical asynchronous ICIC method, which utilized the information exchange capability among adjacent cells provided by the new systems. In [50], the authors have analyzed the limitation of the traditional ICIC method, and proposed an improved priority based ICIC method in order to improve the efficiency in LTE-Advanced systems.

Interference mitigation in the licensed spectrum is mainly focused on enhancing the cell edge spectrum efficiency, and deals with the same operator cells in a wireless HetNet. In the unlicensed spectrum since the same channel is being shared by multiple operator's LTE-U Small Cell, the interference mitigation has to be achieved by coordination across the Small Cells of multiple operator such as our proposed usICIC mechanism, that enhances the system performance not just for the edge users, but rather for all the users that are within the overlapping coverage area of the multi-operator LTE-U Small Cells.

- **High Capacity, Scalable, and Energy Efficient Design**

A multi-core processor is typically used to replace several individual processors for a high performance embedded distributed, and parallel processing system architecture design. A lot of research work has been published that evaluated the two possible multi-core processing configurations, and highlighted the superiority of asymmetric multi-processing (AMP) configuration that has deterministic execution behavior but lacks scalability, over non deterministic but scalable symmetric multi-processing (SMP) configuration, while designing high performance embedded distributed, and parallel processing systems. In [51], the characteristics and implementation of asymmetric multi-processing (AMP) framework in humanoid robot is discussed. The proposed framework was used to replace computer network as the high level processing unit for the humanoid robot that was designed to perform object localization based on visual and auditory information. In this AMP framework, a multi-core

computer was divided into several smaller virtual machines that owned a part of the physical resources including processing core, memory and input/output (I/O) devices. Each virtual machine executed a guest operating system (OS) and dedicated applications. Xen paravirtualization technology was used to conveniently manage these guest operating systems. In [52], the authors provided a discussion on the issues related with the design of embedded multi-processor systems, with the focus on DSP-based systems. Issues that were discussed in this paper included indicators of when applications would scale into multi-core processors designs, Central Processing Unit (CPU) benchmarking, algorithm partitioning and the applicability of symmetric multi-processing (SMP) and asymmetric multi-processing (AMP) architectures on embedded DSP based multiprocessor design. In [53], the authors presented the case that embedded appliances such as high end cell phones require not only high performance, but also a performance guarantee. The authors demonstrated a performance guarantee framework using asymmetric multi-processing (AMP) approach using a multi-core processor, and backed it by providing evaluation results to show that the framework improved the performance guarantee while maintaining software compatibility. In [54], the authors provided a detailed study of the performance issues associated with symmetric multi-processing (SMP) configuration on a multi-core processor. These issues included memory bandwidth restrictions for memory intensive applications; the use of software locks by existing SMP operating system to manage complex shared data structures. In [55], the authors highlighted the fact that writing

parallel software effectively for embedded systems is not an easy task. The authors believed that a new approach was needed to maximize the performance speed-up, and proposed a layered top-down model for parallel embedded software, based on their pattern language for high-performance computing. Several case studies were developed to demonstrate the strength of the proposed model and in one such case study a telecommunication system was migrated from a naive symmetric multi-processing (SMP) setup to an asymmetric multi-processing (AMP) set-up to demonstrate the performance enhancements.

The difference between the work presented in this dissertation, that has also resulted in an IEEE publication [40], and the other published research works has been that prior to our research work there was no precedence of combining the non real time control plane, and the real time sensitive data planes of one or more application servers of a wireless communication system on a single partition comprising of one or more processing cores of a multi-core processor, to be served by a single instance of an open source real time operating system. Our proposed new high performance distributed and parallel processing system configuration for multi-core processors that we called “Deterministic Symmetric Multiprocessing (D-SMP)” configuration, provides the desired deterministic execution behavior of the AMP configuration while offering the equally desired scalability feature of the SMP configuration where the number of processor partitions and consequently the number of operating system instances do not vary with the number of application servers configured in the system. We have also demonstrated in this dissertation how our

proposed D-SMP configuration may be used for the design of a commercial grade high capacity next generation wireless communication system.

1.3 Contributions and Claims of Originality

The material presented in this dissertation has so far resulted in two IEEE publications [40] [45], LTE-U Forum technical contribution [46], and several patent applications. Our specific contributions towards each of the research work presented in this dissertation are as follow:

- **Interference mitigation for the LTE networks in the unlicensed spectrum**

  Our contribution (IEEE publication [45], LTE-U Forum technical contribution [46]) for this research work is summarized as follows:

  - We present the LTE and Wi-Fi co-existence mechanism as proposed by LTE-U Forum [23], and define the deployment scenario that will result in overall degraded LTE-U system performance as a result of multi-operator LTE-U co-channel co-existence in non-LBT regions.
  - We propose an unlicensed spectrum Inter Cell Interference Coordination (usICIC) mechanism to significantly improve the overall LTE-U system performance for multi-operator LTE-U co-channel co-existence in non LBT regions.
  - We demonstrate by extensive simulation results that the proposed usICIC mechanism will result in 40% or more improvement in overall LTE-U system performance.

- The LTE-U service and the associated radios corresponding to the 5 GHz unlicensed spectrum may be opportunistically turned off to make the communication system energy efficient while dealing with low traffic demand.

- **High Capacity, Scalable, and Energy Efficient Design**

  An IEEE publication [40] presents the research work related with the new scalable distributed and parallel processing system architecture that may be employed in designing the next generation Green wireless communication system. This research work encompasses the following design artifacts:

  - For the very first time in the history of wireless industry, the control plane, and the time sensitive data planes of all configured application servers including the ones leveraging the unlicensed spectrum are consolidated under a single Deterministic-SMP partition comprising of all the cores of a multicore processor, and being served by a single open source real time operating system (RTOS) instance. A patent [33] has been granted by the U.S. patent office for this radical and innovative design idea.
  - To design a communication system of varying sizes the choice of the multicore processor with more or less cores may vary depending upon the specific communication system capacity, and cost targets. Thus software portability becomes an important design consideration especially for the application software. The Core abstraction layer (CAL) has been introduced that abstracts or hides the hardware

specific details from the application software to address this design goal.

- In a single partition Deterministic-SMP configuration the control plane (non-real time threads/processes) and the data plane (real time threads/processes) share the same operating system instance even though they are bound to run on different cores. In such a configuration, a system software lock taken by a non-real-time process or thread on one core may cause a latency spike for a real time thread or process waiting for that lock to be released on a different core. Thus there was a need for a custom Lock-less Zero copy Buffer management service as an alternative to the native RTOS buffer management service provided by the open source RTOS, to meet the performance criteria essential for commercial deployment.
- The protocol stack of the chosen RTOS SMP Linux with PREEMPT_RT is not designed to be 100% guaranteed lock-less. This may result in serious performance implications for the time sensitive process or threads of the application server running on the data plane cores to use the Linux protocol stack to send or receive TCP/UDP IP packets. Since the non real time processes, and threads of the control plane also use this protocol stack, the performance implication in an SMP environment result from the fact that a spinlock taken by a non-real-time process or thread on one core (such as found on the control plane core) may cause an unbounded latency spike or busy waiting for a real time thread or process waiting for that lock to be released on a different core (such as found on the data plane core). These

unbounded latency spikes are handled well by the control plane applications (OA&M etc), but result in the breakdown of the data plane applications that have very stringent real time performance requirements. Thus there was a need for a custom lock-less zero copy messaging service in a multicore SMP environment for use by the real time process or threads running on the data plane cores to send or receive TCP/UDP IP packets without the use of the SMP Linux protocol stack.

- A performance optimization scheme on the data plane cores was needed that significantly reduced the number of interrupts, and the resulting context switch overheads on the data plane cores. This enhancement increased the processing capabilitites of the application servers on the data plane cores for increased system capacity.
- High system availability is a key performance criterion required for the commercial deployment. Since multiple application servers share the same multicore processor, the recovery mechanism for a failed application server could no longer use a processor reboot. An elaborate application server recovery mechanism has thus been defined, that performs application server recovery without affecting the other operational configured application servers in the communication system that shared the same single partition D-SMP configuration. A patent [38] has been granted by the U.S. patent office for this design artifact.
- This design drastically reduces the device count used in wireless communication systems by a factor of 14 to make it energy efficient.

Additionally a method and system for dynamic power control has been defined that further reduces the power consumption of the multicore processor by up to 30% throughout the day by modulating the clock speed of the individual cores and operating them at less than maximum clock speed depending upon the system load on each configured application server.

1.4 Dissertation Outline

The rest of the dissertation is organized as follows:

In chapter 2, we present our research work related with improving the spectral efficiency of LTE networks in the unlicensed spectrum using a proactive small cell interference mitigation mechanism. The technical content of this chapter has been captured in a research paper [45] that has been accepted for publication in an upcoming edition of the IEEE Transactions on Wireless communications. This chapter describes our proposed unlicensed spectrum Inter Cell Interference Coordination (usICIC) mechanism for enhancing the overall LTE-U system performance by 40% or more.

In Chapter 3, we present the research work that deals with defining a distributed and parallel processing system design that leverages the unlicensed spectrum, and results in a low cost, high capacity, flexible, scalable, and energy efficient architecture for building the next generation communication systems. A number of viable system designs are presented, and the rationale behind the chosen and the rejected design are discussed in this chapter. This chapter also discusses the portability, scalability, application server recovery, and energy efficiency considerations associated with the proposed system design.

In chapter 4, a discussion on the conclusion and summary for the dissertation research works, and future research directions are presented.

In Chapter 5, a number of appendixes with supporting information have been provided. **Appendix A** provides some high level information about the industry leading real time operating system VxWorks. **Appendix B** provides some high level information about Open Source Linux with PREEMPT_RT. **Appendix C** provides some high level information about Critical Section of Code, and software locks. **Appendix D** provides a discussion on the multicore processor P4080 hardware acceleration engine Buffer Manager (BMAN) buffer Initialization. **Appendix E** provides a discussion on the multicore processor P4080 Data Path Acceleration Architecture (DPAA) configuration for IP packet processing using the Protocol Stack as defined by the Board Support package (BSP) provided by the OS vendor. **Appendix F** provides a discussion on the multicore processor P4080 Data Path Acceleration Architecture (DPAA) configuration for IP packet processing without using the Protocol Stack by augmenting the Board Support package (BSP) provided by the OS vendor. **Appendix G** provides a discussion on the multicore processor P4080 hardware acceleration engine Queue Manager (QMAN) configuration.

Chapter 6 provides a list of the references used in this dissertation. Chapter 7 provides a list of the abbreviations, acronyms, and terms used in this dissertation.

Chapter 2

2 Interference Mitigation for the LTE Networks in the Unlicensed Spectrum

2.1 Introduction

The demand for mobile wireless data is growing at an exponential rate in recent years. Industry is preparing for what is being labeled as the 1000x data traffic growth in the coming years. To meet such a stunning demand a several fold increase in spectrum [15] [16] is needed. The use of unlicensed spectrum is thus critically needed to aid the existing licensed spectrum to meet such a huge mobile wireless data traffic growth demand in a cost effective manner. Currently the unlicensed 5 GHZ band is used by Wi-Fi [13] [18]-[22] capable devices such as smart phones, laptops and tablet PCs. Wi-Fi 802.11 n/ac protocols are suitable for interference free environment, but offer unpredictable Quality of Service (QoS) and degraded performance in interference limiting environment, which are considered far inferior to those offered by licensed cellular wireless technologies such as LTE [14] [17] that offers higher spectral efficiency [10].

The support for LTE in the unlicensed 5 GHZ band where it has to co-exist with Wi-Fi, and where the LTE carrier is transmitted according to the lower transmit power requirements of the unlicensed spectrum, makes it suitable only for Small Cell deployments. A small cell utilizing LTE-L (LTE in licensed spectrum), and LTE-U (LTE in unlicensed spectrum) will therefore significantly reduce the total cost of ownership (TCO) of the small cell for the

service provider. LTE-U capable small cell will have a lower cost per bit of wireless data transmitted, while providing the additional mobile wireless data offloading capacity from Macro Cell to small cell in LTE Heterogeneous Networks (HetNet).

LTE-U is a solution developed based on 3GPP LTE Rel-10/11/12, with implementation-based approach at the eNodeB to properly address the coexistence issue. This solution applies to regions such as U.S., Korea, and India that do not mandate Listen-Before-Talk (LBT) regulations. To harmoniously co-exist with the incumbent Wi-Fi, LTE-U employs channel selection and LTE-U ON/OFF period in a carrier sense adaptive transmission (CSAT) duty cycle co-existence mechanism to share the unlicensed channel in TDM manner with Wi-Fi and other operators' LTE-U deployments. The LTE-U Technical Specifications were defined and published in March 2015 by the LTE-U Forum that includes as key participating members Verizon, Alcatel-Lucent, Ericsson, LG Electronics, Qualcomm Technologies, and Samsung. The commercial deployment of LTE-U based solution is expected to happen during the second half of 2016.

The basic mode of LTE operations in the unlicensed spectrum is the Supplemental Downlink (SDL) [5]. In this mode of operations the Best Effort (BE) user data in downlink is carried over the unlicensed secondary carrier also known as SCell. All control and uplink user data is carried over the anchor primary LTE licensed carrier also known as PCell. The licensed PCell and the unlicensed SCell are employed by the small cell, and the User Equipments (UEs) in the Carrier Aggregation (CA) mode.

Figure 1 depicts the single operator wireless HetNet deployment scenario comprising of a Macro Cell, and three Wi-Fi, and LTE-U capable small cells. Wi-Fi coverage area is uplink (UL) limited due to the lower device uplink transmit power requirement in the unlicensed spectrum. On the other hand, LTE-U transmission is Supplemental Downlink (SDL) only, and all uplink control and critical data is transmitted via the anchor licensed LTE-L carrier (PCell), thereby increasing the LTE-U effective coverage area relative to the Wi-Fi coverage area for the same small cell.

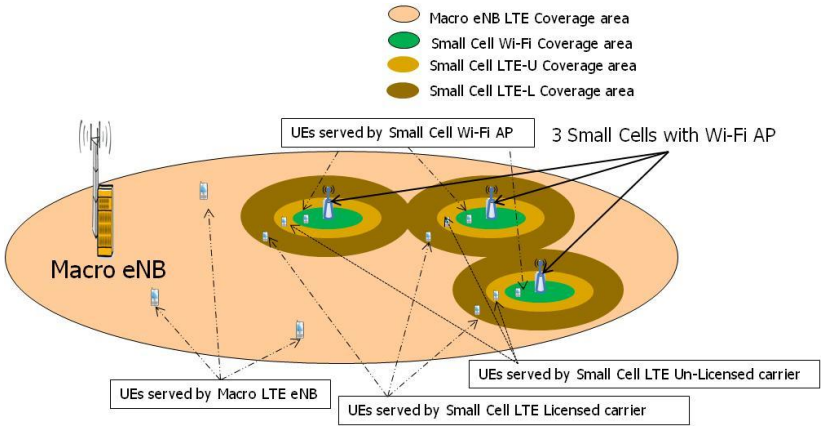


Figure 1: Single Operator LTE-U Deployment Scenario in a Wireless HetNet

The limited number of available 20 MHZ channels in the 5 GHZ unlicensed spectrum will therefore require channel sharing among Wi-Fi APs, and LTE-U small cells in a dense multi operator deployment scenario.

Studies have been published [5]-[7] to demonstrate not only the viability of LTE deployment, but also its harmonious co-existence with Wi-Fi in the unlicensed spectrum. The published LTE and Wi-Fi coexistence performance studies [5] [6] [9] [11] have demonstrated that using carefully designed co-existence mechanism, LTE can be a fair neighbor to Wi-Fi that

does not degrade the Wi-Fi AP performance more than when the Wi-Fi AP share and co-exist with another Wi-Fi AP on the same unlicensed channel.

However, none of the published literature has so far examined the impact on the overall LTE-U system performance due to the interference resulting from multi-operator LTE-U channel sharing. In this dissertation, we study this key aspect, and quantify and highlight the need for an explicit interference mitigation mechanism for multi-operator LTE-U channel sharing for countries such as U.S., that do not mandate the LBT requirements for operations in the unlicensed spectrum.

The rest of the chapter is organized as follows. In Section 2.2, we present the system model and problem description for multi-operator LTE-U co-channel co-existence scenario that employs the LTE and Wi-Fi coexistence mechanism in the unlicensed spectrum as proposed by the LTE-U Forum. In Section 2.3 we present our proposed usICIC mechanism for enhancing multi-operator LTE-U co-channel co-existence scenario performance. In Section 2.4, we provide performance evaluation by describing the simulation setup, scenarios, results, and the corresponding analysis.

2.2 System Model and Problem Description

As part of the technical specifications released for the deployment of LTE-U, the LTE-U Forum [23] has proposed CSAT [5] [24] [25] to be used as the non LBT co-existence mechanism for the harmonious co-existence of LTE with Wi-Fi in the unlicensed spectrum. The CSAT mechanism allows the LTE-U small cell share an unlicensed channel with Wi-Fi in a Time-Division-Multiplexing (TDM) manner. In this adaptive transmission scheme, the LTE-U

small cell employs a transmission gating cycle T_{CSAT} with a LTE-U ON/OFF duty cycle (Figure 2) to co-exist with Wi-Fi APs in a fair manner on the same unlicensed channel. The T_{CSAT} gating cycle is divided into Transmission Time Interval (TTI) or timeslots, each of which is a millisecond or sub-frame wide. Thus N timeslots or sub-frames for LTE-U ON (Z_1 msec), and another M timeslots or sub-frames for the LTE-U OFF (Z_2 msec) pattern (Figure 2). For the case of T_{CSAT} gating cycle with 50% duty cycle, N = M.

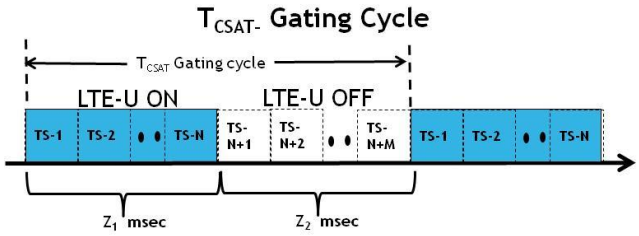


Figure 2: TCSAT Gating Cycle for LTE-U Channel Sharing with Wi-Fi

The CSAT mechanism works well for LTE and Wi-Fi co-existence, because the co-channel Wi-Fi backs off during the LTE-U ON transmission period of the T_{CSAT} gating cycle due to the Carrier Sense Multiple Access (CSMA) nature of its Medium Access Control (MAC) layer. By using an appropriate duty cycle (e.g., 50%) for the CSAT transmission gating cycle (T_{CSAT}), LTE-U small cell ensures that it leaves at least 50% of the channel for Wi-Fi use thereby ensuring a proper TDM sharing of the channel with Wi-Fi. However, an uncoordinated use of CSAT transmission gating cycle does not work well for multi-operator LTE-U small cells co-channel co-existence scenario, mainly because none of the LTE-U small cells sharing the channel backs off to each other's LTE-U transmissions during their respective LTE-U ON period of the gating cycle. This situation is exacerbated when each of

them use overlapping T_{CSAT} gating cycles for their LTE-U operations. If there is no coordination between multi-operator LTE-U small cells for the use of the LTE-U transmission gating cycle T_{CSAT} for co-channel LTE-U co-existence, it will result in overall degraded LTE-U system performance. The two LTE-U small cells that are either co-located or non-co-located with overlapping coverage area may each independently select an overlapping LTE-U transmission gating cycle. This will result in severe interference to their LTE-U users in the overlapping coverage area of the two LTE-U small cells for the duration of the overlapped LTE-U transmission gating cycle.

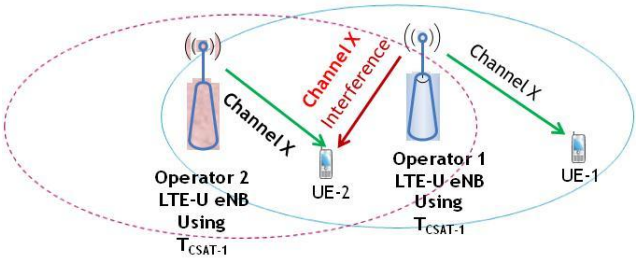


Figure 3: Multi-Operator LTE-U Co-channel Interference Scenario

Figure 3 depicts such a multi-operator LTE-U small cell deployment scenario, where Operator 1 LTE-U small cell and operator 2 LTE-U small cell each select the same unlicensed 5 GHZ Channel X and also use non-coordinated overlapping LTE-U transmission gating cycle for their LTE-U SDL operations. UE-1 is being served by operator 1 LTE-U small cell, and UE-2 is being served by operator 2 LTE-U small cell. The Operator 1 LTE-U small cell SDL transmission to UE-1 will result in interference to UE-2 during its simultaneous SDL operations with its serving Operator 2 LTE-U small cell for

the duration of the overlapped LTE-U T_{CSAT} gating cycle, and will subsequently impact the LTE-U system throughput performance.

Let's denote $\mathbf{S}_{i,j,k,t}$ as the received signal power at the i^{th} user from its serving LTE-U small cell *j*, of operator *k* at TTI *t* of the T_{CSAT} gating cycle, and $\mathbf{N}_{thermal}$ is the thermal noise at i^{th} user. Let's use Ø(t) to denote the set of interfering nodes (Wi-Fi or LTE-U) transmitting during a TTI *t* of the T_{CSAT} gating cycle on the unlicensed channel X. A node *l* is part of the interfering nodes set Ø(t) only if it is transmitting at TTI *t.* The aggregate interference $\mathbf{I}_{i.j.k.t}$ or the received signal power from the interfering nodes *l* in set Ø(t) at the i^{th} user that is being served by LTE-U small cell *j* of operator *k* at TTI *t* can be represented by $\sum_{l\in\emptyset(t)} I_{i,j,k,l,t}$.

Instantaneous signal to interference plus noise (SINR) ratio [24] for a given TTI *t* reflects instantaneous received signal powers from different transmitting nodes. Unlike Wi-Fi, LTE-U SINR distribution is independent of decoding. Thus, the SINR for the i^{th} user served by LTE-U small cell *j* of operator *k* for a TTI *t* is defined by:

$$\boldsymbol{SINR}_{\boldsymbol{i,j,k,t}} = \frac{\mathbf{S}_{i,j,k,t}}{\mathbf{N}_{thermal} + \sum_{l\in\emptyset(t)} I_{i,j,k,l,t}} \qquad \text{---------- (1)}$$

Thus for the multi-operator LTE-U co-channel interference scenario depicted in Figure 3, the higher interference level $\mathbf{I}_{i.j.k.t}$ during the overlapping portion of the LTE-U ON period of the T_{CSAT} gating cycle of the nearby LTE-U small cells will result in lower $\boldsymbol{SINR}_{\boldsymbol{i,j,k,t}}$, leading to degraded LTE-U system performance.

To address this co-channel LTE-U interference issue, and to enhance the overall LTE-U system performance (throughput gains) in non-LBT unlicensed spectrum, there is a need for an explicit LTE-U Inter Cell Interference Co-ordination mechanism. We present such a mechanism in the next section that will facilitate the coordination and usage of orthogonal non-overlapping LTE-U transmission gating cycles between adjacent multi-operator small cells that co-exist on the same unlicensed channel X (e.g., channel 149 in the 5GHZ band), for their LTE-U SDL operations.

2.3 Unlicensed Spectrum Inter Cell Interference Coordination (usICIC) Mechanism

2.3.1 Overview

The main goal of our proposed usICIC mechanism is to reduce the inter-cell interference and enhance the overall system performance for the multi-operator LTE-U co-channel co-existence scenario. When two or more LTE-U small cells with overlapping coverage area, share an unlicensed channel for their respective LTE-U SDL operations, the usICIC mechanism enables them to negotiate with each other, over the backhaul X2 eNodeB signaling interface [3], the use of non-overlapping CSAT transmission gating cycle of appropriate duty cycle. Thus, when one LTE-U small cell is transmitting during its LTE-U ON period of the CSAT transmission gating cycle, the other LTE-U small cells sharing the same channel will be quiet (or OFF) leading to significant reduction in interference ($\mathbf{I}_{i.j.k.t}$) to their users (i^{th} user of serving small cell j of operator K at TTI t of the CSAT gating cycle) throughout the overlapping coverage area of the LTE-U small cells.

When the LTE-U transmission gating cycles of two co-located or near-by multi-operator small cells that share the same channel X, is perfectly aligned, it will result in the worst case interference for LTE-U co-channel co-existence scenario. However, when the small cells employ orthogonal non-overlapping LTE-U transmission gating cycle via an inter cell coordination mechanism such as usICIC it will result in the best case interference mitigation scenario for multi operator LTE-U co-channel co-existence.

Since the small cells of different operators will not be time synchronized across the same sub-frame boundary, during the CSAT gating cycle coordination process the adjacent small cells will also establish the start of each other's transmission gating cycle relative to their own sub-frame boundaries in an attempt to achieve as orthogonal a transmission gating cycles as possible with minimum overlap. For the non-coordinated scenario that does not employ usICIC mechanism, the level of LTE-U transmission gating cycle overlap between the two small cells will be a random distribution between the worst case (perfectly aligned and completely overlapping) and best case (orthogonal non-overlapping) scenarios. Consequently, the actual interference level and the resulting LTE-U system performance will be somewhere within the two extreme boundary conditions of worst case, and best case co-ordination scenarios.

2.3.2 Workflow of the Proposed usICIC Mechanism

Figure 4 depicts the event trace diagram of our proposed usICIC mechanism with high level description for each step involved in the usICIC mechanism.

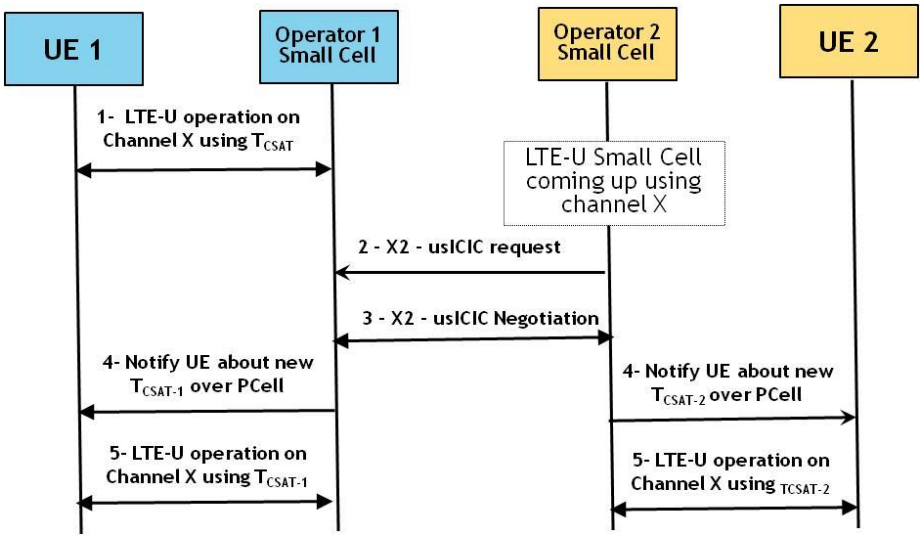


Figure 4: Event Trace Diagram for the proposed usICIC Mechanism

- In **Step 1**, operator 1 small cell is performing LTE-U SDL operations with UE 1 on unlicensed channel X (e.g., channel 149 in the 5GHZ band).
- In **Step 2**, as operator 2's small cell starts to initiate its LTE-U operations, its channel selection algorithm picks channel X for its LTE-U operations. Upon detecting the presence of other operator (operator 1) LTE-U small cell on the same channel X, operator 2 small cell initiates the usICIC request with Operator 1 small cell over the backhaul eNodeB X2 signaling interface.
- In **Step 3**, the two small cells then exchange relevant information (such as number of Wi-Fi AP nodes presence on the channel X, number of LTE-U small cell nodes presence on Channel X, the timing information to notify

each other about the start of their sub-frame boundaries etc) over the backhaul X2 eNodeB signaling interface, to negotiate and determine as orthogonal a T_{CSAT} gating cycle as possible of appropriate duration for their LTE-U operations to share the channel X in time domain.

- In **Step 4**, the two small cells notify their respective serving UEs about the beginning and end of the negotiated LTE-U ON period of the CSAT gating cycle from step 3, by the activation/deactivation of the unlicensed secondary carrier (SCell) using existing 3rd Generation Partnership Project (3GPP) standards based MAC (Medium Access control) channel element signaling over the primary carrier that is anchored in the licensed spectrum (PCell). This allows the UEs to know when to expect the LTE-U SDL transmission (LTE-U ON period of the T_{CSAT} gating cycle) form their serving LTE-U small cell.
- In **Step 5**, the two small cells initiate their LTE-U SDL operations with their respective users (UEs) while causing minimal to no interference to each other's LTE-U SDL operations with users in their overlapped coverage area.

If channel X is only shared by two LTE-U small cells, the negotiated orthogonal (non-overlapping) T_{CSAT} gating cycles for the two LTE-U small cells cannot use the entire 100% time duration of channel X. Some portion of the channel usage time has to be left free of any LTE-U activity to allow a nearby Wi-Fi AP to sneak in, and share the channel as well. In such a case, the number of nodes sharing channel X will change from 2 to 3 (Two LTE-U small cells, and one Wi-Fi AP). This change in channel node usage will trigger another round of usICIC negotiations between the two LTE-U small cells that

would result in the two LTE-U small cells using orthogonal CSAT gating cycle that use one third of the available channel time.

The usICIC mechanism applies equally for the scenario where the same unlicensed channel is shared by more than two LTE-U small cells, such as the case where an unlicensed channel is shared e.g., by three LTE-U small cells, and a Wi-Fi AP. Using the usICIC mechanism the three LTE-U small cells may each negotiate the use of their respective orthogonal non overlapping CSAT gating cycle that use one fourth of the available channel time, while leaving 25% of the channel time exclusively for use by the Wi-Fi AP, thereby reducing interference for all nodes sharing the unlicensed channel.

The goal of our proposed interference mitigation usICIC mechanism is to minimize $\mathbf{I}_{i.j.k.t}$ by reducing the interference caused by two or more co-channel LTE-U small cells to each other's LTE-U SDL operations. The use of the orthogonal non- overlapping CSAT transmission gating cycle by each co-existing LTE-U small cell on the same unlicensed channel will result in better SINR distribution, which will lead to higher user (50^{th} percentile, Average, 95^{th} percentile) and small cell system throughputs on the unlicensed carrier. The improvement in SINR distribution results from the fact that when one LTE-U small cell *j* is transmitting during a TTI *t* of its LTE-U ON transmission gating cycle, the other LTE-U small cells that share the same unlicensed channel X, will not be simultaneously transmitting, and will thus not cause interference to the users served by the LTE-U small cell *j* of operator *k*.

The usICIC handshake mechanism is triggered only when a new LTE-U small cell starts to share an unlicensed channel with another LTE-U small cell, or if there is a change in the number of nodes sharing the unlicensed channel.

Therefore, once the orthogonal CSAT duty cycle duration, and start, and end time is negotiated, the LTE-U small cell may continue to use it for its LTE-U SDL transmission. If there is a change in the CSAT duty cycle duration, and start, and end time as a result of a new usICIC handshake mechanism that is being done in parallel with the on-going LTE-U SDL transmissions, the LTE-U small cell will start using the new settings once they become available after the successful conclusion of the new usICIC handshake mechanism.

The SINR distribution, user (50^{th} percentile, Average, 95^{th} percentile) and small cell average throughput are the three key performance indicators (KPIs) that we have utilized in our LTE system level simulations to evaluate the performance of our proposed usICIC mechanism for the multi-operator LTE-U co-channel co-existence scenario.

The LTE-U small cells average throughput is dependent upon Channel Resource Utilization [24], Channel Loading [24], and Channel Congestion [24] metrics. Better SINR distribution resulting from the use of our proposed usICIC mechanism will result in better channel conditions so the small cell will be able to complete its LTE-U SDL transmissions to its UEs much faster, thereby reducing the channel loading and congestion metrics and improving the LTE-U small cell average throughput.

Channel Congestion metric $\boldsymbol{C_{j,k}}$ for the LTE-U small cell *j* of operator *k*, is a function of the Channel Resource Utilization $U_{j,k}$ and Channel Loading $\boldsymbol{L_{j,k}}$, and is defined as follows:

$$\boldsymbol{C_{j,k}} = 1 - \left(\frac{U_{j,k}}{L_{j,k}}\right) \qquad \text{---------- (2)}$$

If T is the set of TTIs over a given period of time, such that

$$T = \{t_1, t_2, \ldots, t_n\} \qquad \text{---------- (3)}$$

Let **ʊ** denote the set of users served by LTE-U small cell *j* of operator *k* within its coverage area of the 5GHz unlicensed spectrum. Then the Channel Resource utilization $\boldsymbol{U_{j,k}}$ for the LTE-U small cell *j* of operator *k* is defined as the fraction of time the LTE-U small cell is transmitting to one of its users in set ʊ over the unlicensed channel X. It is thus represented as follows:

$$\boldsymbol{U_{j,k}} = \left(\frac{\sum_t 1(P_{j,k,t})}{T}\right) \qquad \text{---------- (4)}$$

Where the indicator function **1 ($P_{j,k,t}$)** is defined as follows:

$$\mathbf{1}(\boldsymbol{P_{j,k,t}}) = \begin{cases} 1 & If\ small\ cell\ \ j,k\ is\ transmitting\ in\ TTI\ t \\ 0 & If\ small\ cell\ \ j,k\ is\ Not\ Transmitting\ in\ TTI\ t \end{cases} \qquad \text{---------- (5)}$$

LTE-U small cell unlicensed Channel Loading $\boldsymbol{L_{j,k}}$, may be defined as the fraction of the time LTE-U small cell *j* of operator *k* has data to transmit (non empty queues) to its users defined by set **ʊ** within its coverage area. Let $\mathbf{q_{i,t}}$ be the size of the queue for the i^{th} user in the set $\boldsymbol{ʊ_{j,k}}$ at Transmission Time Interval (TTI) *t*.

An indicator function **1($ʊ_{i,t}$)** is defined on the set ʊ such that the value of the indicator function is 1, if the small cell *j* of operator *k* has a non empty queue at TTI *t* for its i^{th} user or element of user set **ʊ**. The value of the indicator function **1($ʊ_{i,t}$)** will otherwise be 0.

$$1\ (ʊ_{i,t}) = \begin{cases} 1 & if\ q_{i,t} > 0\ at\ TTI\ t \\ 0 & if\ q_{i,t} = 0\ at\ TTI\ t \end{cases} \qquad \text{---------- (6)}$$

Then, the Channel Loading $L_{j,k}$ over the unlicensed 5 GHZ unlicensed spectrum for each LTE-U small cell *j* of operator *k* may therefore be represented as follows:

$$L_{j,k} = \left(\frac{\sum_t \sum_{i \in \upsilon} 1\,(\upsilon_{i,t})}{T} \right) \quad \text{---------- (7)}$$

So using equations (2) - (7), the overall congestion metric of an unlicensed channel is defined as follows:

$$C_{j,k} = 1 - \left(\frac{\sum_t 1\left(P_{j,k,t}\right)}{\sum_t \sum_{i \in \upsilon} 1\,(\upsilon_{i,t})} \right) \quad \text{---------- (8)}$$

The LTE-U small cell continues to monitor the channel conditions (Wi-Fi activity, number of co-existing Wi-Fi, and LTE-U Small Cell) of other unlicensed channels during its LTE-U OFF period of the CSAT transmission gating cycle. This is done to identify a candidate set of unlicensed channels to switch the LTE-U operation if the current unlicensed channel condition continues to yield degraded system performance. Thus by monitoring the Congestion metric $C_{j,k}$ in addition to the system throughput, the LTE-U small cell may easily decide when to switch its LTE-U operations to an alternative cleaner unlicensed channel if the degraded system performance persists for a pre-defined period of time. Additionally the small cell may utilize these metrics to determine when to turn its LTE-U SDL operations off while dealing with light traffic that could easily be handled by the licensed LTE carrier. This opportunistic turning off of the small cell LTE-U SDL operations [24] will make it a fair neighbor to other nodes sharing the unlicensed channel, because by doing so it makes the unlicensed channel available to other transmitting nodes to use completely.

2.4 Performance Evaluation

In this section, we present the simulation setup, scenarios, results, and the corresponding analysis.

2.4.1 Simulation Setup

We performed LTE system level simulations that were based on the scenario #1 of 3GPPTR 36.872 [4].The MATLAB based LTE system level simulator employed for this simulation work has been developed in Bell Labs, Alcatel-Lucent, which provides LTE protocol stack support at the PHY (layer 1), and the MAC (layer 2) with complete scheduling functions. All Simulation parameters and assumptions are summarized in Table 1.

The Simulations employed two operators, one small cell per operator, and 30 users per operator in 21 macro cell layout with Inter Site Distance (ISD) of 500 meters. Three carriers were employed in the simulation: 2 licensed carriers at 2GHz (one licensed carrier or PCell per operator), and 1 unlicensed carrier SCell for LTE-U SDL operations with a bandwidth of 10 MHz at 5GHz (shared by the small cells of two operators). One transmit and one receive antenna configuration was used at each small cell with transmit power of 30 dBm used for licensed carrier (PCell) and 24 dBm for unlicensed carrier (SCell).

In the LTE system level simulation, the 21 macro cell layout was achieved by 7 macro cell sites, with 3 sectors per site. Within each of these 21 macro cells coverage areas a cluster of small cells was deployed. In each cluster, one small cell was deployed for each operator, so that there were two small cells per macro cell cluster for evaluating the performance of a multi-

operator unlicensed spectrum Inter Cell Interference coordination (usICIC) mechanism.

Table 1: Simulation Parameters & Assumptions

Parameter	Value
Layout	21 Macro Cells layout
Macro Cell Inter Site Distance (ISD)	500 m
Number of Licensed Carriers	2 (2 GHz for licensed carriers)
Number of Unlicensed Carrier	1 (5 GHz for unlicensed carrier)
LTE-U System Bandwidth	10 MHz
Number of Operators	2
Number of small cell Clusters Per Macro Cell	1
Number of Small Cells Per Operator per cluster	1
Number of UEs	30 per macro cell per operator
UE Dropping	2/3 UEs randomly and uniformly dropped within the clusters, 1/3 UEs randomly and uniformly dropped throughout the macro cell coverage area that contained the cluster
Tx Power on Unlicensed	30 dBm over licensed carrier, and 24 dBm over unlicensed carrier
Cell Selection or Association Criteria	Reference Signal Received Power (RSRP) based on the licensed carrier
Antenna Configuration	1Tx, 1 Rx
Traffic Model	Full Buffer
LTE-U T_{CSAT} Duty Cycle	50%
LTE-U ON Period	100 msec
LTE-U OFF Period	100 msec

The simulation employed 30 UEs for each operator's Small cell dropped within each of the 21 macro cell coverage area. Two-thirds of these UEs were randomly and uniformly dropped within the cluster, and the remaining one-third UEs were randomly and uniformly dropped throughout the macro cell coverage area that contained the cluster. Cell selection or association was based on the licensed carrier reference signal received power (RSRP) criteria. A full buffer traffic model was employed in the simulation to ensure that the

small cell utilized the entire duration of its LTE-U ON period of its T_{CSAT} transmission gating cycle. For performance evaluation, two types of multi-operator small cell layouts within a macro cell cluster were considered:

- Non co-located multi operator small cells that were 80 meters apart.
- Co-located multi operator small cells that were 3 meters apart.

2.4.2 Simulation Scenarios

Four scenarios were simulated to evaluate the performance of our proposed usICIC mechanism compared to the non coordinated case using the three key performance indicators (SINR distribution, user throughput, and small cell average throughput). In each scenario the T_{CSAT} gating cycle with 100 msec LTE-U ON, and 100 msec LTE-U OFF intervals (i.e. 50% duty cycle) was employed. In this section, we first describe the simulation scenarios, and then provide simulation results and comparison analysis of the three key performance indicators for the four scenarios, to highlight and quantify the LTE system performance enhancements achieved by our proposed usICIC mechanism.

- **Scenario # 1 (Worst Coordination) Non Co-Located small cells –** Non Co-Located (80 meters apart) Multi-operator LTE-U small cells sharing the same unlicensed channel X, and using the same fully time aligned T_{CSAT} gating cycle for their LTE-U SDL transmission.
- **Scenario # 2 (Worst Coordination) Co-Located small cells –** Multi-operator Co-located (3 meters apart) LTE-U small cells sharing the same unlicensed channel X, and using the same fully time aligned overlapping T_{CSAT} gating cycle for their LTE-U SDL transmission.

- **Scenario # 3 (Best Coordination) Co-Located small cells –** Multi-operator Co-located (3 meters apart) LTE-U small cells sharing the same unlicensed channel X and employing our proposed usICIC mechanism to use orthogonal non overlapping T_{CSAT} gating cycle for their LTE-U SDL transmission.
- **Scenario # 4 (Random No Coordination) Co-Located small cells –** Multi-operator co-located (3 meters apart) LTE-U small cells sharing the same unlicensed channel X, and using randomly partially aligned overlapping T_{CSAT} gating cycle for their LTE-U SDL transmission. Typically the small cells of two operators are not time synchronized and aligned on the same sub-frame boundary. So in the case of non coordinated multi-operator small cells (Scenario #4), the level of T_{CSAT} gating cycle (LTE-U ON interval) overlap will be a random distribution between [0,..,100] msec. Therefore, for Scenario #4 (Random No Coordination case), the simulations were repeated 10 times with randomly generated T_{CSAT} overlapping gating cycle patterns for each simulation run, and the LTE-U performance averaged across these 10 runs to achieve statistically meaningful result.

2.4.3 Comparative Analysis of the Simulation Results

The LTE-U SINR distributions comparison for the four simulated scenarios is shown in Figure 5. Scenario #3 (Co-Located small cells - Best Coordination using our proposed usICIC mechanism) has the best SINR distribution that is even better than that of Scenario #1 (Non Co-located small cells), where the cells were non co-located, and thus had smaller overlapping LTE-U coverage area that limited the impact of interference each small cell LTE-U transmissions would have caused to their respective users. This highlights the fact that the multi-operator small cells that are either co-located or non co-located, and share the same channel X with overlapping coverage area for their LTE-U SDL operations will benefit with better LTE-U SINR distribution leading to improved overall LTE-U system performance from using our proposed usICIC mechanism.

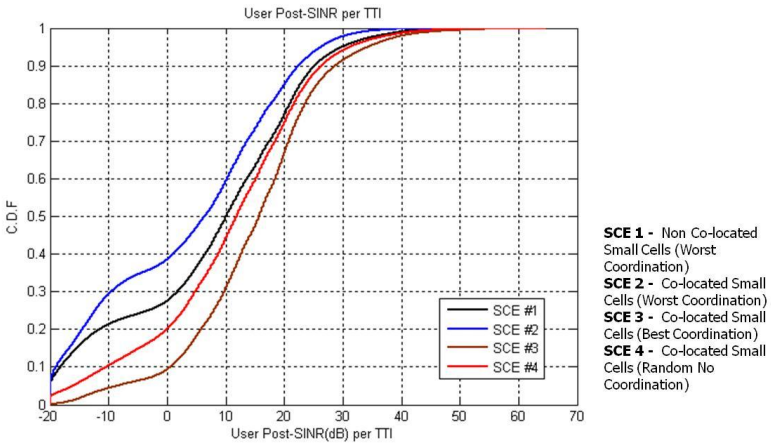


Figure 5: LTE-U SINR Distribution Comparison

The LTE-U user throughput (50th percentile, Average, and 95th percentile) comparison for the four simulated scenarios is shown in Figure 6. Scenario #3 (Co-Located small cells – Best Coordination that employed our proposed usICIC mechanism) has the best user throughput results, and

demonstrate that the use of our proposed usICIC mechanism will result in substantial LTE-U system performance enhancement.

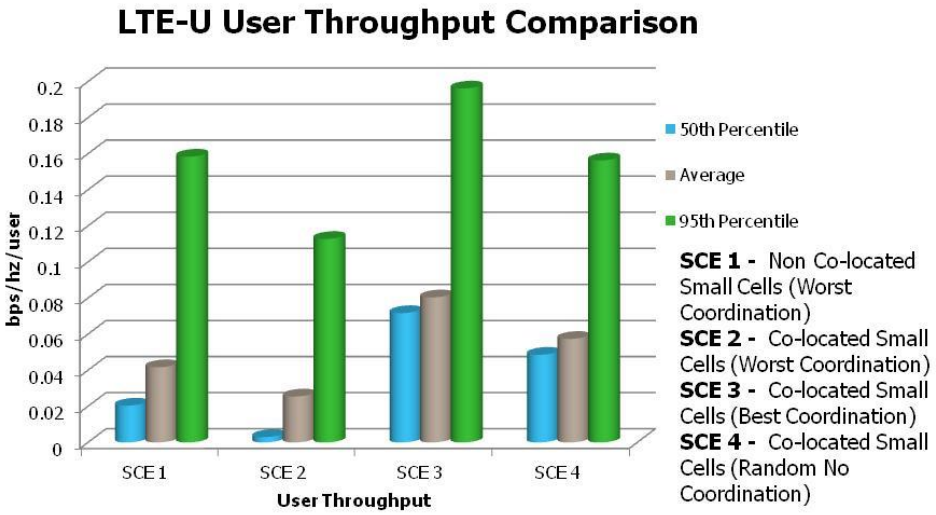


Figure 6: LTE-U User Throughput Comparison

The LTE-U small cell average throughput comparison for the four simulated scenarios is shown in Figure 7. The results demonstrate that performance of the Scenario # 4 (random no co-ordination) will lie somewhere between [Worst Coordination Performance (Scenario #2), Best Coordination Performance (Scenario # 3)]. Scenario # 3, that employed our proposed usICIC mechanism had the best overall LTE-U system performance, and yielded a performance improvement by a factor of 1.4 (or 40%) compared to Scenario # 4 (Random No multi-operator coordination), and a performance improvement by a factor of 3.15 (0r 215%) compared to that of Scenario # 2 (Worst coordination).

The simulation results also demonstrate that Scenario #3 (Co-located Cells) that employed our proposed usICIC mechanism has a performance that

is 2.05 times (or 105%) better than that of the Scenario #1 (Non Co-located Cells) that employed no coordination. Thus, even the overall LTE-U system performance of non co-located multi-operator small cells that share the same channel X and have some overlapping coverage area will also benefit from the use of our proposed usICIC mechanism.

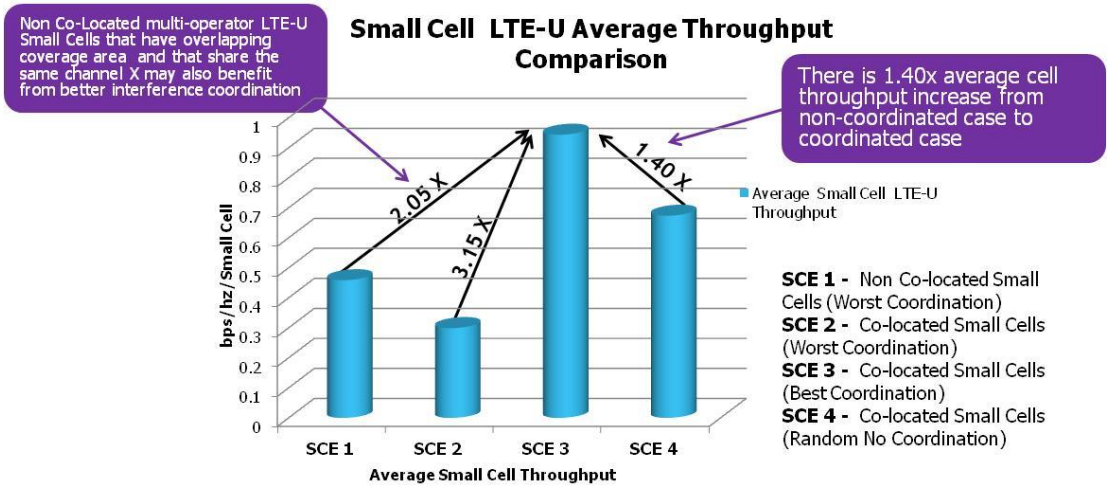


Figure 7: LTE-U Small Cell Average Throughput Comparison

Chapter 3

3 High Capacity, Scalable, and Energy Efficient Design

The demand for mobile wireless data has been stunning in recent years, and is expected to continue to grow exponentially in coming years. This chapter presents the research work related with the design for a new distributed, and parallel processing system architecture for next generation wireless communication systems, that is driven by requirements such as lower cost, high capacity, flexibility, scalability, and energy efficiency to reduce the capital expanse (CAPEX), and recurring operational expanse (OPEX) for the service providers. A wireless base-station is a practical real world example of the next generation communication systems that may benefit from the design strategies presented in this dissertation.

By leveraging the unlicensed spectrum, the capacity of the next generation wireless communication system could be significantly increased by a factor of two. Unlike the licensed spectrum that is extremely costly and typically of the order of multi-billion dollars for each market segment, there is no similar upfront CAPEX implications for the service providers to leverage the unlicensed spectrum, and significantly increase their network capacity to not only meet the ever increasing mobile wireless data demand in a cost effective manner, but also to significantly enhance the overall user Quality of Experience (QoE).

The next generation communication system will require the use of multiple application servers including the ones that manage data transmissions in the unlicensed spectrum, to handle different classes of traffic. An LTE Layer 2

(L2) scheduler is a real world example of an application server. Each application server requires a management or control plane and a time-sensitive data plane. Since the control plane handles and deals with jitters and latency spikes much better than the time-sensitive data plane, different system configuration treatments are needed for each. Typically, the support of each application server requires the use of two processors to achieve physical separation of the control plane and the time-sensitive data plane. Each of these processors in turn requires additional dedicated devices such as external double data rate (DDR) memory, and flash memory devices to store the application generic and the operating system image, and other devices necessary for proper operation. The single core processor that serves the control plane hosts the non-real time threads and processes such as the operation, administration, and maintenance (OA&M) process. The processor hosting the control plane may use any operating system, even one that is open source. The single core processor that serves the data plane hosts the real time threads and processes. The processor hosting the data plane has stringent real time performance requirements, and is traditionally served by a dedicated hard core real time operating system. These real time performance constraints stem from the fact that the real time process typically has to run once every millisecond (msec), and needs to complete the necessary processing well within this one msec time interval. Exceeding this critical boundary for real time execution may result in throughput degradation, and even worse, a breakdown in system behavior leading to overall system instabilities.

Traditionally the same system design and configuration is replicated for each configured and operational instance of the application server. Thus the number of processors and operating system instances used in the system vary with the number of configured and operational application servers. Typically a deployment royalty and licensing fee is paid to a third party tier-1 operating system (OS) supplier such as Wind River Systems. The fee is dependent upon the number of OS instances used in the communication system. This not only makes scalability an issue for next-generation communication systems, but the increase in device count also contributes significantly to the energy efficiency, size/footprint of the system and the cost of goods sold (COGS).

The rest of the chapter is organized as follows: Section 3.1 provides the rationale behind the use of open source real time operating system (RTOS). Section 3.2 describes the various multi-core processor system configurations and provides the reasoning behind the chosen configuration. Section 3.3 presents the portability, and scalability aspect of the design. Section 3.4 discusses the critical aspect of application server recovery mechanism. Section 3.5 presents the energy efficiency consideration for the proposed design. Finally, section 3.6 provides a discussion on the testing aspect for the new design.

3.1 Use of Open source Real Time Operating System (RTOS)

Traditionally a third party proprietary hard core real time operating system (RTOS) such as vxWorks [42] [44] made by Wind River Systems is used to serve the data plane that hosts the real time processes and threads.

The control plane hosts the non-real time processes and threads, and its performance requirements are easily met by using an open source OS such as Linux.

To reduce the overall system cost of goods sold (COGS), a design alternative we chose was to use an open source RTOS such as symmetric multi-processing (SMP) Linux with PREEMPT_RT patch [41] [42] to serve the data plane instead of the industry-wide popular proprietary RTOS such as vxWorks [42] [44]. The upfront development licensing cost of using open source SMP Linux with PREEMPT_RT, provided by a tier-1 OS supplier such as Wind River or MontaVista, is significantly less than using any of their proprietary real time operating systems. These companies provide extensive testing of the open source Linux code and also provide patch management, which reduces concerns around indemnification issues that may result from the use of open source SMP Linux with a PREEMPT_RT patch. In addition, no deployment royalty/license fee has to be paid to the OS supplier for using an open source RTOS on each communication system or gateway that is sold and deployed commercially.

The use of the open source RTOS even though help reduce the system COGS, but opens the door for serious performance challenges to meet the stringent real time performance requirement of the data plane application software process/threads. A number of high performance optimization services and schemes were needed to help overcome the performance issues that are posed by the use of a non hard core open source RTOS.

3.2 Use of a Multicore processor

Our architectural design required the use of a powerful multicore processor that has the processing powers to support multiple high capacity application servers. We selected Freescale Semiconductor's P4080 multicore processor [31], mainly because it had eight e500mc processing cores, a data path acceleration architecture (DPAA) that contained hardware acceleration engines that provide high performance packet processing, and forwarding capabilities with minimal to no impact to the processing cores. This critical packet processing offload to hardware acceleration engines means that more processing power becomes available on the cores that may be used to increase the system capacity. The processing horsepower is one of the limiting factors in determining the system capacity. The combination of the more powerful e500mc processing cores and the use of the hardware acceleration engines available on the multicore processor increased processing capacity by a factor of 1.5 compared to the previous generation single core processor. The DPAA hardware acceleration engines we employed in our design include a buffer manager (BMan), queue manager (QMan), and two frame managers (FMans). The BMan provides data buffer management in hardware for the buffers originally created by software. The QMan provides queuing and quality of service (QoS) scheduling of frames to the appropriate cores. The FMan supports multiple external Ethernet interfaces, and provides the capability to perform inline packet parsing, general classification to enable policing, and QoS based packet distribution to central processing unit (CPU) cores for further processing.

At a high level, each P4080 processor core has a software portal that has a specified number of channels. Each channel is assigned to a particular end point from where the frames may be de-queued. Each core portal is an end point and will have a dedicated channel. The FMan is also an end point, and will have a dedicated channel for each configured external Ethernet interface it supports. Each channel has eight priority based work queues (WQs) to implement QoS, and each WQ contains a plurality of first-in first-out (FIFO) frame queues (FQs) that contain the FIFO frames, where a frame represents a single buffer or list of buffers that hold data, e.g., packet payload.

In the egress direction where an application process may try to send packets over the external Ethernet interface, the application with help from a kernel driver will queue the packets onto the FQ of an FMan channel associated with the external Ethernet interface. The FMan, with help from Qman, de-queue frames from the channel's FQ to transmit packets on the corresponding external Ethernet interface.

In the ingress direction, the FMan will receive packets on the external Ethernet interface, and based on the programmed parsing, classifying, and distributing (PCD) rules, route the packet to the appropriate core for further processing by queuing it onto the FQ associated with the appropriate channel assigned to the core portal.

The rationale behind using a multicore processor in our design was to consolidate the control and data planes of all the configured application servers onto a single multicore processor. Doing so significantly reduced the device count needed in the new system configuration for distributed computing and parallel processing. This not only drastically reduced the

COGS and the size of the system, but also reduced extra hardware and software complexity, and potential failure points. The next few subsections describe the multicore configurations that were considered, and the rationale behind the chosen configuration.

3.2.1 Supervised Asymmetric Multi-Processing (S-AMP) Configuration

The most obvious choice for multiple application server support using a multi-core processor is the supervised asymmetric multi-processing (S-AMP) configuration [32] [43] as shown in Figure 8.

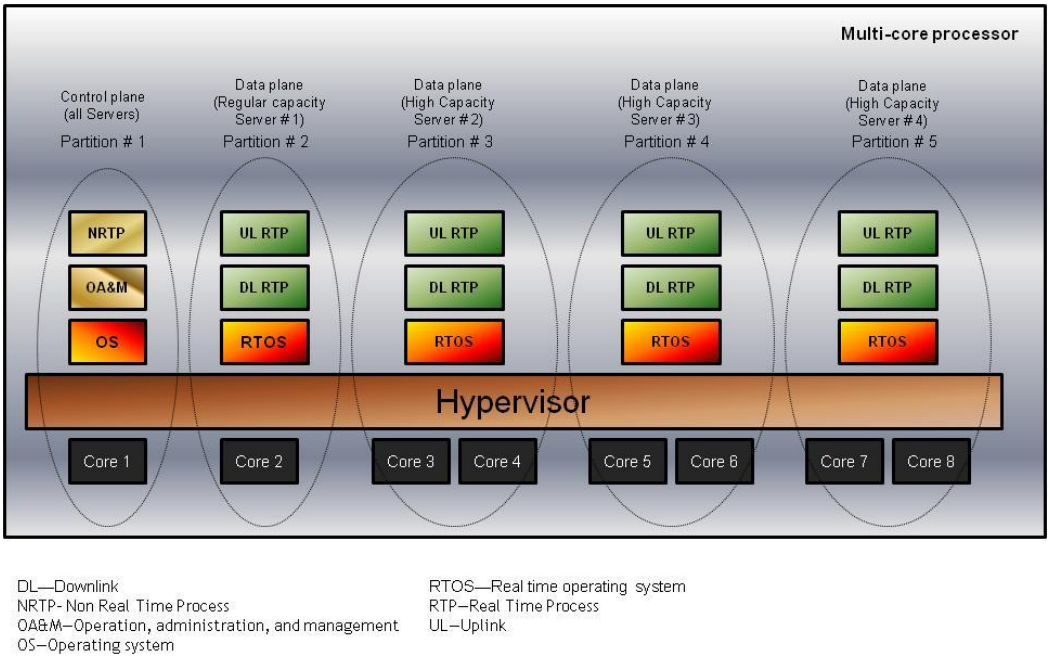


Figure 8: Supervised Asymmetric Multi-Processing (S-AMP) Configuration

A high capacity application server will require two cores, whereas a regular capacity application server may require just one processing core for its data plane. Each partition of the S-AMP configuration has its own dedicated operating system instance, and therefore mandates the use of another system

supervisory software entity called a Hypervisor [32] [43] for proper system operation. The Hypervisor ensures that one OS instance will not corrupt other partitions of the multi-core processor.

A system using S-AMP configuration that supports three high capacity, and one regular capacity application server as shown in Figure 8, will therefore consists of five partitions: a common control plane partition, and four dedicated data plane partitions one for each configured application server.

The use of a Hypervisor also adds significantly to the system COGS, as per instance licensing fee has to be paid to the Hypervisor vendor for its usage. Scalability is a serious drawback of this configuration (the total number of partitions and the RTOS instance needed vary with the number of configured application servers in the system).

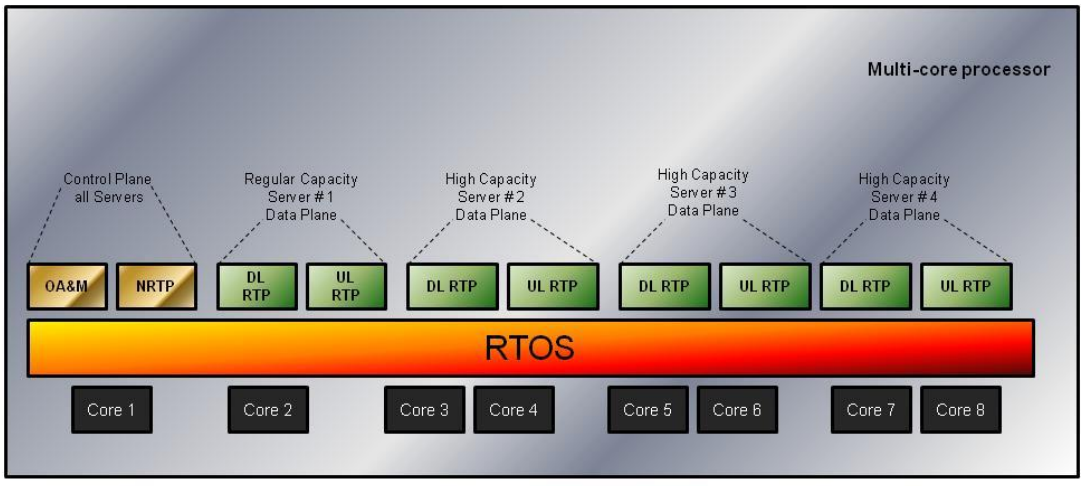
The cost and scalability issues associated with the S-AMP configuration forced us to look for an alternative cost effective and scalable configuration solution for our system design.

3.2.2 Symmetric Multi-Processing (SMP) Configuration

A less intuitive choice for multiple application server support on a multicore processor is the symmetric multi-processing (SMP) configuration [32] [43] with a single partition that comprised of all the processing cores.

This configuration, shown in Figure 9 does not use a hypervisor [32] [43] which results in a highly scalable architecture since a single partition with one OS instance serves all control and data planes, irrespective of the number of application servers that are supported by the system. The SMP configuration, however, loses the desired deterministic behavior of the

supervised AMP (S-AMP) configuration which offers clear segregation of the control and data planes. The consolidation of control and data planes under the same SMP partition using a single OS instance may result in mixed execution of non-real time control plane threads/processes and the real time data plane threads/processes on the same cores. This consolidation of control and data planes under one partition coupled with the use of an open source RTOS such as SMP Linux with PREEMPT_RT poses a serious challenge to meet the stringent real time performance needs of the data plane real time threads/processes.



DL—Downlink
NRTP- Non Real Time Process
OA&M–Operation, administration, and maintenance
OS–Operating system
RTOS—Real time operating system
RTP–Real Time Process
UL–Uplink

Figure 9: Symmetric Multi-Processing (SMP) Configuration

Thus the performance issues and the non-deterministic execution behavior of the SMP configuration makes it unsuitable for our system design.

3.2.3 Deterministic Symmetric Multi-Processing (D-SMP) Configuration

The shortcomings of the SMP configuration may be addressed by employing core reservation and core affinity constructs provided by SMP Linux with PREEMPT_RT to achieve AMP-like system behavior in a SMP configuration. We call this the deterministic symmetric multi-processing (D-SMP) configuration [33]. In this configuration, all non-real time processes/threads such as OA&M are bound using core affinity to a core that is dedicated for the control plane activities. A default affinity mask to the control plane core is also defined to ensure that any process/thread that has no specific core bindings will default to the control plane core for execution. This ensures that the cores that are dedicated for the data plane activities are reserved to run only real time processes/threads.

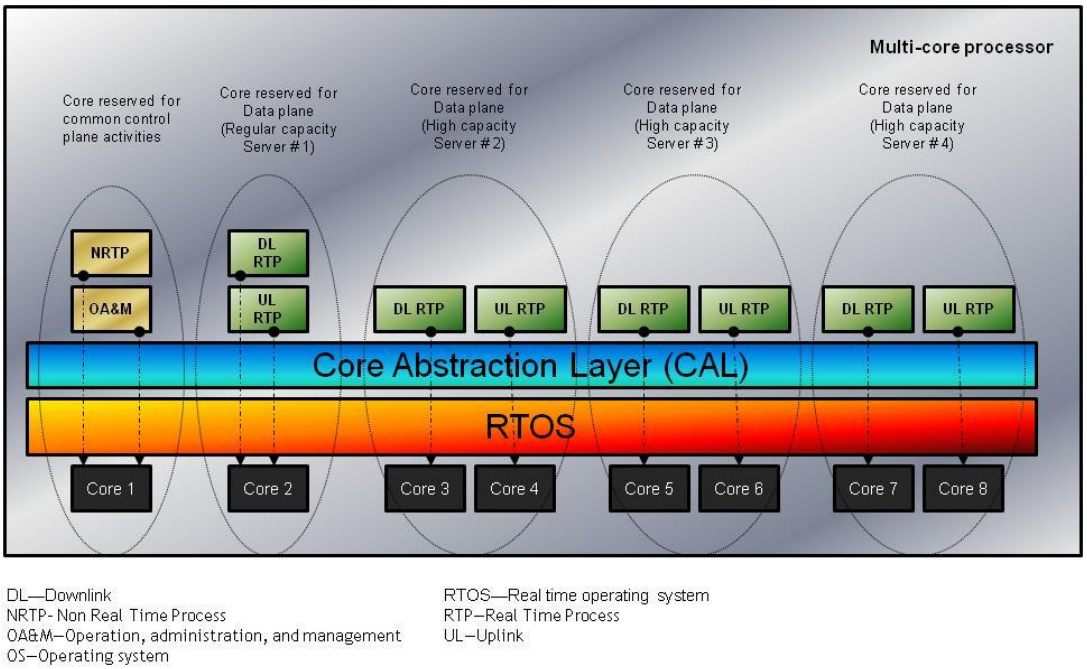


Figure 10: Deterministic Symmetric Multi-Processing (D-SMP) Configuration

Figure 10 shows the D-SMP configuration that support three high capacity and one regular capacity application server. The real time data plane process for each application server is bound to its own dedicated data plane core using core affinity and the core reservation construct offered by SMP Linux with PREEMPT_RT. In this configuration the non-real time processes/threads will not have to compete with higher priority real time processes/threads for processing time as they have their own dedicated control plane core for execution. Consequently, they will also not take any valuable processing time away from the real time processes/threads that are executing on the data plane cores.

A core abstraction layer (CAL), described in more detail in the next section, is a middleware layer that is defined in our architecture with an eye towards future software portability as it hides the core and hardware-specific details from the data plane application software. CAL additionally provides high performance system services such as buffer management, and messaging for the data plane processes/threads. Without these high performance services, it is otherwise impossible to use an open source RTOS such as SMP Linux with PREEMPT_RT in a single partition D-SMP configuration that serves all the control and data planes using the same OS instance, and still meet the stringent real time performance needs of data plane processes/threads.

The D-SMP configuration is highly scalable as it uses just a single SMP partition and a single RTOS instance irrespective of the number of supported application servers in the system.

Additionally, the non-essential interrupts such as software direct memory access (DMA) interrupts are disabled on the data plane cores and instead these interrupts are programmed to occur on the control plane core [37]. Doing so will ensure that no unnecessary context switch from a user space application to an interrupt service routine (ISR) running in kernel space, will occur on the data plane cores as a result of these interrupts.

To support high capacity it is important to reduce all unnecessary processing overhead required to handle each incoming packet on the processing cores. To achieve this goal, the hardware acceleration engines on the multi-core processors may be configured such that each incoming packet arriving for handling on a data plane core does not result in an interrupt but rather polled from the hardware queues by the application software. Each context switch resulting from an interrupt, incurs an appreciable processing overhead. By eliminating this unnecessary processing overhead, more processing time and power will become available on the data plane cores to serve the real time processes and threads for increased capacity. Please refer to appendix G for more optimization details.

3.3 Portability and Scalability Considerations

In our system configuration, we have chosen Freescale Semiconductor's P4080 multicore processor [31] with eight processing cores. However, to design a system of varying sizes, the choice of the multicore processor with more or less cores may vary depending upon the specific system capacity and cost targets.

Thus software portability and scalability becomes an important design

consideration, especially for the application software. It is imperative to have a middleware layer that abstracts or hides the hardware-specific details from the application software. This middleware layer may also provide performance optimized services such as buffer management and messaging to the application software in case any performance related issues are identified during testing while using the system services provided by the native open source RTOS. Performance issues are triggered by unbounded latency spikes that can cause the data plane processes/threads to exceed the critical boundary of one msec for execution, and result in throughput degradation that eventually leads to system instabilities which can cause an application server outage. These random un-bounded latency spikes manifestation is mainly due to the use of system services and the protocol stack of the open source RTOS such as SMP Linux with PREEMPT_RT, as their implementation is not guaranteed to be lock-less. In concurrent programming, a critical section is a piece of software code that accesses a shared resource such as a device or a data structure that must not be concurrently accessed by more than one process or thread of execution . The critical section is guarded by a software lock that may be typically implemented via semaphore to achieve synchronization at the entry and exit of the critical section. As critical section usually terminates in fixed time, a process or thread has to wait a fixed time to enter it and thus experiences a bounded latency. However if the semaphore is acquired by a low priority non-real time process or thread, and then preempted by an interrupt prior to exiting the critical section, other high priority real time processes or threads that need access to the same critical section will have to endure a random unbounded latency spike since they will have to

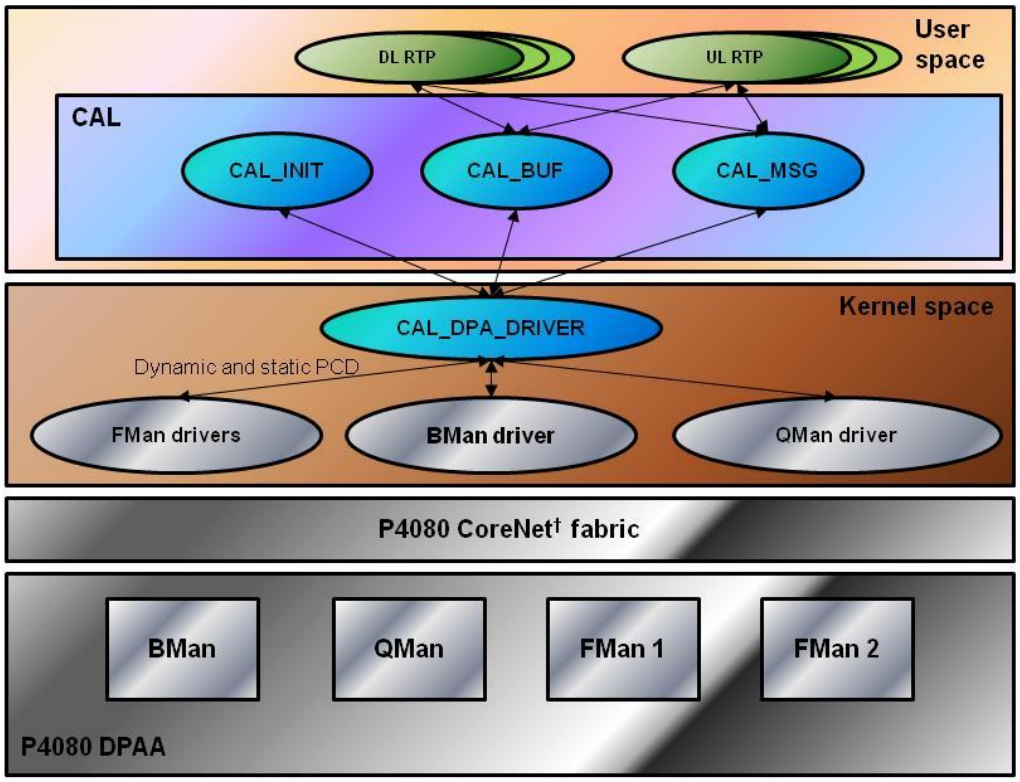
wait until the non-real time process or thread is scheduled to run again, and thus releases the software lock or semaphore after finishing the execution of the critical section. Thus when the same system services are used by real time processes/threads on the data plane cores as well as the non-real time processes/threads on the control plane core, a software lock taken by a non-real time process or thread on the control plane core may cause a latency spike for a real time process or thread on the data plane core waiting for the release of the same lock. Therefore, data plane processes/threads should avoid the direct use of any system services that are offered by the native operating system, and instead should rely on the alternative high performance services offered by the custom middleware layer CAL. This is done to ensure that not only the performance criterion is met, but also to help facilitate the future portability of software to new platforms with different multicore processors.

3.3.1 Core Abstraction Layer (CAL)

Figure 11 illustrates the functional architecture of the core abstraction layer (CAL) [34] which is one of the integral building blocks of our proposed system configuration architecture. This middleware layer hides the core and hardware-specific details of the multicore processor from the user space applications that will make it easier for software portability to different multicore processors based architecture in the future.

The BMan, FMan, and QMan Drivers shown in Figure 11, are part of the standard board support package (BSP) provided by the OS vendor. The CAL framework consists of three user space modules: the CAL initialization

module (CAL_INIT), the CAL buffer module (CAL_BUF), and the CAL messaging module (CAL_MSG) that provide a user space application programming interface (API) for use with data plane user space applications, and a custom kernel space driver, the CAL_DPA_DRIVER, that serves as an interface to the DPAA hardware acceleration engines.



†Trademark of Freescale Semiconductor, Inc.

BMan—Buffer manager (hardware acceleration engine)
CAL—Core abstraction layer
CAL_BUF—CAL buffer module
CAL_INIT—CAL initialization module
CAL_MSG—CAL messaging module
DL—Downlink
DPAA—Data path acceleration architecture
FMan—Frame manager (hardware acceleration engine)
QMan—Queue manager (hardware acceleration engine)
P4080—Freescale multi-core processor
PCD—Parsing, classifying, and distributing rule for packet routing within the P4080 CoreNet Fabric
RTP—Real Time Process
UL—Uplink

Figure 11: Core Abstraction Layer Functional Architecture

The CAL_INIT module is responsible for setting up the CAL infrastructure necessary to support buffer management and messaging services and to initialize the CAL_DPA_DRIVER that is used to manage the

P4080 [31] data path acceleration architecture (DPAA) resources. This module also provides a network interface configuration file as well as parsing, classification, and distribution (PCD) rules that the CAL_DPA_DRIVER can use to initialize and configure the PCD component of the FMan. Once configured, it allows routing of incoming frames on external Ethernet interfaces to appropriate cores with no involvement from the cores.

CAL_BUF module provides performance-optimized buffer management services to be used exclusively by the “fast path” which is the transmission path taken by the ingress and egress data plane traffic of data plane process/threads. CAL_BUF collaborates with CAL_DPA_DRIVER to provide lock-less zero copy buffer management service for buffers that the CAL_DPA_DRIVER creates, but which are managed by the BMan [31] hardware acceleration engine. BMan manages up to 64 pools of buffers on behalf of the software for both hardware (hardware acceleration engines and network interfaces) and software use, and provides hardware arbitration for simultaneous access for a buffer from multiple sources on various cores without the need for a software lock. Since no software locks are required to acquire or release a buffer, no process or thread will be blocked while waiting for a lock to be released. Please refer to appendix D for more details.

CAL_MSG provides performance-optimized messaging services for the “fast path.” The fast path does not use the Linux protocol stack to send or receive Transmission Control Protocol/User Datagram Protocol (TCP/UDP) Internet Protocol (IP) packets. CAL_MSG interfaces with the CAL_DPA_DRIVER to provide lock-less zero copy messaging services via

the P4080 multicore processor's DPAA. Please refer to appendix E, and appendix F for more details.

3.4 Application Server Recovery Mechanism

Since the system supports multiple application servers on a single multicore device, application server recovery [38] can become complex. In the previous implementation, where each application server had its own dedicated processors, the recovery procedure following a software crash was simple. In that case, the associated processors were rebooted, and all of the hardware and software components properly initialized, ready to reconfigure the application server.

However, in our proposed architecture, rebooting the multicore processor to recover or rescue an application server is no longer a viable option, since it will cause the common control plane and the remaining active application servers configured on the multicore processor to reboot as well. High system availability is a critical service provider's requirement, and bringing down all of the operational application servers when just one application server has to be restarted or reconfigured after a software crash is not an acceptable option.

Thus the general requirement for the recovery procedure is that the recovery of an application server should not impact the functionalities, behavior, and performances of the other application servers on the same board that is hosting the failing application server. Furthermore, application server recovery should not introduce instability or resource shortages that may critically affect the other active application servers on the board.

The P4080 system resources allocated to an application server are managed in several ways. For example, some resources are allocated from system-wide shared pools. When such resources are allocated (i.e., owned by an application server) they cannot be reused by other components until they are explicitly released. In the case of a failure of the software owning resources from shared pools, the platform software must clean up these resources on behalf of the owner software otherwise the corresponding resources will be lost forever.

To clean up and recover the shared resources owned by a failed application server, the platform needs a way to identify which application server owns them. For that reason each individual resource from a shared pool has an associated tag specifying the owner of the resource. The owner tag should be updated each time an application server resource is allocated and each time it is released to/from the shared pool.

Ingress buffer pools of the Ethernet interface used on the ingress data plane path are an example of system-wide shared resources. They are dedicated to receive data plane packets, and allocated by the FMan driver. However releasing these buffers back to the shared pool is the responsibility of the user space process or application server. Egress buffer pools used on the egress data plane path are an example of application server specific private resources, as these are allocated by the configured application server. Every time an application server goes down, there is a strong likelihood that it may have owned some buffers from the shared pool. If the recovery mechanism does not recover the buffers from the shared pool, after some time, and a few more occurrences of application server failure later, the

shared pool may not have enough buffers left to service the needs of the remaining operational application servers in the system.

The memory management subsystem of an operating system such as Linux manages several virtual memory address spaces: one address space for each process and one address space for the kernel. When a process terminates, the operating system automatically releases all resources allocated from the process address space. Consequently, there is no need for an explicit cleanup of resources that are part of the process address space. The application server specific private resources that are allocated on behalf of a failing application server from the kernel address space should be cleaned up by the platform software driver, e.g., the CAL_DPA_DRIVER. It should be noted that the termination of a process does not automatically clean up the application server resources from kernel modules.

At a high level, upon detecting an application server failure, the recovery mechanism performed the following steps:

- Notified the management module to get the application server's logical IP address removed from the board
- Changed the mapping between the ingress frame queues from the current data plane core back to the control plane core to ensure that all incoming packets destined for the particular core are handled until a new application server is configured again later on this core
- Disabled the interrupts that were needed for the application server process
- Disabled the direct memory access (DMA) transfers associated with the impacted application server

- Explicitly invoked the platform kernel driver CAL_DPA_DRIVER to release the application server specific private resources that were allocated on behalf of a failing application server from the kernel address space since the operating system automatically releases only the resources that are allocated from the process address space
- Explicitly invoked the platform kernel driver CAL_DPA_DRIVER to release any shared resources owned by the failed application server
- Stopped all of the software components and then re-launch all of the software components associated with the failing application server.

After performing the cleanup steps, the application server recovery mechanism sets up the execution environment to host a new application server by performing the following steps:

- Configuring the application server logical IP address on the board
- Setting up the dynamic PCD rule by binding the frame queue (FQ) with the application server's logical IP address to allow FMan to route the incoming data plane packets from the external Ethernet interface to the appropriate core

3.5 Energy Efficiency Considerations

Energy efficiency is an important design consideration for the next generation wireless communication systems as it lowers the recurring operational expenses (OPEX) incurred by the service provider.

3.5.1 Energy Efficiency as a result of significant reduction in device count

As mentioned earlier traditionally, the support of each application server requires the use of two processors to achieve physical separation of the control plane and the time-sensitive data plane. Each of these processors in turn requires additional dedicated devices such as external double data rate (DDR) memory, and flash memory devices to store the application generic and the operating system image, and other devices such as Digital Signal Processors (DSP) necessary for proper operation.

Thus to support seven regular capacity application servers, the previous generation communication system would require the use of 14 processors (two per application servers), and their accompanying devices (DDR memory, Flash memory, DSP etc) per processor. Traditionally the physical hardware design for each application server is on its own dedicated board for plug and play functionality. Alternatively, a board may have support for multiple application servers with the same hardware design replicated multiple times on the same physical board for each application server. Nonetheless such a design configuration would require either seven boards (one application server per board), or more than one board (multiple application servers per board) with backplane board connectors. This not only increases the actual size or footprint of the overall communication system but also increases the power consumption of the communication system as well.

The chosen D-SMP configuration [33] presented earlier in this dissertation utilizes the P4080 multicore processor [31] with 8 cores that is capable of supporting seven regular capacity application servers. Thus with

this design a single multicore device is able to replace 14 individual processors to provide support for seven application servers. This configuration also allows us to use the economy of scale, and use single bigger size memory devices (DDR memory, Flash memory) instead of smaller size devices that were previously needed for each of the 14 processors.

Thus to support seven regular capacity application servers where previously each application server consumed “x” watts of power, the new system design configuration has been able to reduce the overall power consumption by the seven application servers by roughly 85.7% (at a high level a single hardware circuit design has replaced the previous 7 hardware circuit designs that were needed one for each of the application servers).

3.5.2 Energy Efficiency as a result of Multicore Dynamic Power Control

The multicore devices are extremely powerful but consume a lot of power. The FSL P4080 processor with eight processing cores [31] that is used in the proposed system configuration consumes 27 watts while the cores are running @ 1500 MHZ. These sophisticated multicore devices provide hardware support to modulate the operating clock speed of each individual core. e.g., some of the cores may be run at half the operating clock speed depending upon the system load.

The FSL P4080 chipset provides the Dynamic Frequency Scaling (DFS), and core Doze/NAP features. When the Dynamic Frequency Scaling (DFS) mode is triggered for a core, the core is run at less than maximum operating frequency in order to reduce the power consumption. e.g., the core could be run at 800 MHZ instead of 1500 MHZ, resulting in a 10% savings in

overall power consumption as claimed by the multicore device manufacturer. When the cores are transitioned from DFS to “Doze” mode, no further instruction fetching is conducted. The cores are halted, but the clocks remain active. When the cores are transitioned from “Doze” to “Nap” mode, the clocks are gated off. Single core Active/Nap power consumption @ 1500 MHZ is 1.6W/0.8 Watts respectively. This could result in an additional 20% overall power savings.

The application servers do not run at full capacity at all times throughout the day. The communication system may be made even more energy efficient by taking into account the system load of the application servers at any given instance of time, and employing a dynamic power control mechanism [9] to modulate the operating clock speed of each individual core.

A static time based method (e.g. from mid night to 4:00 A.M.) to determine when the application server may be running at off peak capacity may not always yield optimal result. In fact this static approach may cause serious system performance issues because a change in traffic pattern due to any reason may pose a serious problem for the communication system while it is operating in off peak capacity mode with lower clock speed.

A dynamic power control mechanism may be introduced that employs two distinct state machines (one for dedicated data plane cores, and the second one for the shared control plane core) to modulate the clock speed of each of the cores of the multicore processors depending upon the system load to dynamically lower the power consumption of the multicore device. Thus for a communication system containing seven application servers, there will be eight dynamic power control state machines in concurrent operations

(one for the shared control plane core, and seven for the dedicated data plane cores for each of the seven application servers). It is important to note that the data plane of a high capacity application server contains two cores, so in this case the single dynamic power control state machine will modulate the clock speed of the two associated data plane cores of the high capacity application server. A system configuration such as the one shown in Figure 10 that contains three high capacity, and one regular capacity application servers will have five dynamic power control state machines in concurrent operations (one for the shared control plane, and four for the four dedicated data planes core partitions). To avoid the ping-pong effect in state transition, the system load is measured over a number of consecutive polling cycles prior to making the state transition decision where each polling cycle consists of a pre-defined time interval such as 10 seconds.

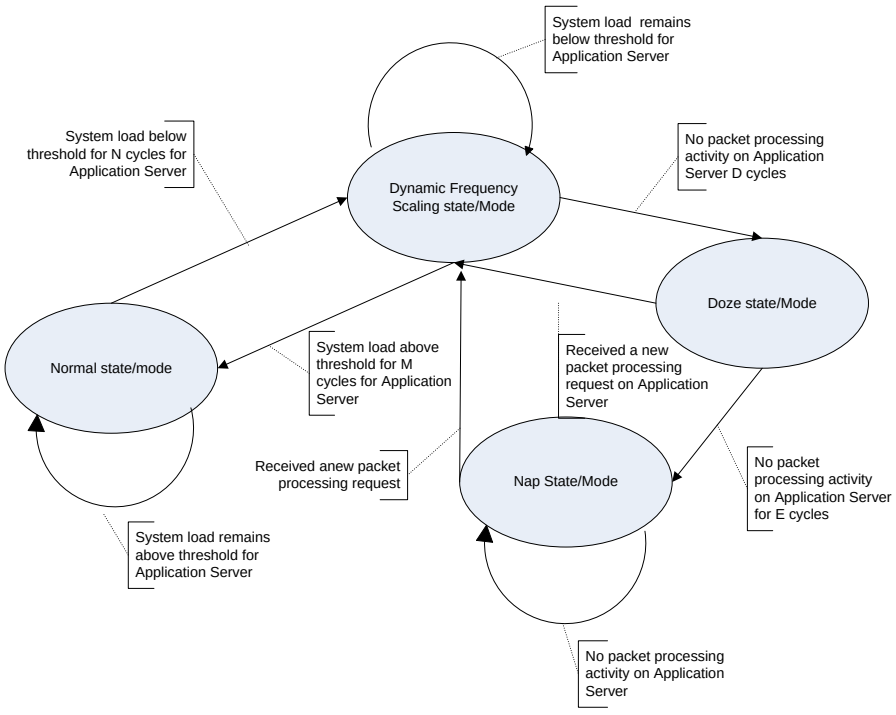


Figure 12: DPC State machine for the dedicated data plane cores

Figure 12 depicts a Dynamic Power Control (DPC) state machine for cores that are dedicated to serve the data plane of each configured application server in the system. The state machine for dedicated cores in the multicore processor includes four states: a normal state, a dynamic frequency scaling (DFS) state, a Doze state, and a Nap state.

The dynamic power control state machine for each dedicated data plane core as shown in Figure 12 operates in the following manner:

a) The state machine for a particular dedicated data plane core stays in the normal state if the system load of the associated application server is above a certain pre-defined threshold (e.g., half the full system capacity of the application server). In the normal state the core will be running at full system frequency and result in no additional power savings

b) The state machine for a particular dedicated data plane core transitions from Normal state to the DFS state when the system load of the associated application server is less than or equal to a pre defined threshold (such as half the full system capacity of the application server) for a number of consecutive polling cycles *N*. When the Dynamic Frequency Scaling (DFS) mode is triggered for a core, the core is run at less than maximum operating frequency in order to reduce the power consumption. e.g., the core could be run at 800 MHZ instead of 1500 MHZ

c) While the state machine is in DFS state it transitions back to Normal state when the system load goes above the pre-defined threshold

(such as half the full system capacity of the application server) for a number of consecutive polling cycles *M*

d) While the state machine for the dedicated data plane core is in DFS state, it transitions to the doze state when there is no system load for the associated application server for a number of consecutive polling cycles *D*. The cores are halted, but the clocks remain active

e) While the state machine of the dedicated data plane core is in Doze state it transitions back to DFS state if a new request for packet processing is received by the application server

f) While the state machine of the dedicated data plane core is in Doze state, it transitions to the nap state when there is no system activity for the associated application server for a consecutive number of polling cycles *E.* Clock to the core is gated off, and result in power savings

g) While the state machine of the dedicated data plane core is in Doze state, it will remain in this state as long as no new request for packet processing is received for the associated application server, otherwise the state machine transition to DFS state

Figure 13 depicts a dynamic power control state machine for shared core that serves the control plane of all configured application servers in the system. The state machine for shared control plane core in the multicore processor includes just two states: a normal state, and a dynamic frequency scaling (DFS) state. Since the control plane is shared by all the configured application servers, its associated DPC state machine does not include the Doze, and NAP states. The state machine is triggered by the computation of

system load that is a cumulative load of all application servers that are operational in the system. The dynamic power control state machine for shared control plane core operates in the following manner:

a) The state machine for the shared control plane core stays in the normal state if the cumulative system load of all associated application server is above a certain pre-defined threshold (e.g., half the full system capacity of all the application servers). In the normal state the core will be running at full system frequency and result in no additional power savings

b) The state machine for the shared control plane core transitions from Normal state to the DFS state when the cumulative system load of all the application servers is less than or equal to a pre defined threshold (such as half the full system capacity of all the application servers) for a number of consecutive polling cycles *N*. When the Dynamic Frequency Scaling (DFS) mode is triggered for a core, the core is run at less than maximum operating frequency in order to reduce the power consumption. e.g., the core could be run at 800 MHZ instead of 1500 MHZ

c) While the state machine is in DFS state it transitions back to Normal state when the cumulative system load goes above the pre-defined threshold (such as half the full system capacity of all the application servers) for a number of consecutive polling cycles *M,* otherwise it stays in DFS state

Without the dynamic power control mechanism the same maximum power is consumed by the multicore device throughout the day irrespective of the

system load or usage. Thus utilizing a dynamic power control mechanism may result in 10%-30% reduction in the multicore device power consumption during time periods when the system is operating at below capacity due to traffic patterns. Such a savings may make the wireless communication system even more energy efficient and good for the environment as every little bit of energy conservation matters when there are tens of thousands of wireless communication systems deployed to meet the ever growing demands of the wireless mobile data.

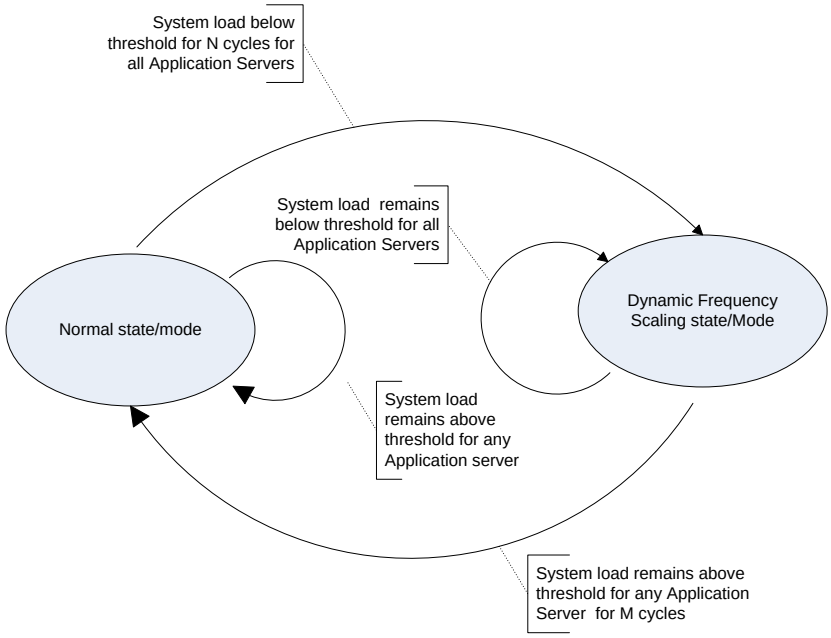


Figure 13: DPC State machine for the shared control plane core

3.5.3 Energy Efficiency as a result of Disabling the LTE-U Service

Depending upon the traffic pattern that does not necessitate the use of the unlicensed spectrum for additional system capacity, the LTE-U SDL service and the corresponding radios associated with the 5 GHz unlicensed spectrum may be opportunistically turned off throughout the day to reduce the

power consumption and make the wireless communication system even more energy efficient.

3.6 Testing and Performance Validation Considerations

High system availability is a key performance criterion which service providers require for commercial deployment. A system that performs adequately only some of the time is not acceptable. An end-to-end black box stress testing of the system design is essential to validate that the radical design alternatives chosen and presented here are capable of meeting the performance requirements.

Off-the-shelf simulator test equipment may be employed to conduct end-to-end stress testing of the system. Such a tool provides the capability to inject into the system the maximum capacity data rate the system is configured to support, and also provides a means to validate the uplink and downlink throughput for the duration of the stress test.

The testing philosophy employed was simple: The system utilizing the architectural design concepts presented here should support the maximum data rate allowed by the given system configuration for an extended period of time lasting up to several days with no degradation in throughput or system instabilities leading to a system outage.

As part of the new system capabilities developed, a service was also created to allow monitoring of the processing load on a per-core basis for the multicore processor. Thus the processing loads on the data plane cores where the application servers reside could be closely monitored throughout the duration of the stress test. Such a monitoring was deemed essential to

identify the presence of any random unbounded latency spikes in the system caused by spin locks, so that once identified, they may be eliminated by employing appropriate optimization techniques. A spinlock is a lock which causes a thread trying to acquire it to simply wait in a loop or spin, while repeatedly checking if the lock is available. Since the thread remains active but is not performing a useful task, the use of such a lock is a kind of busy waiting. Once acquired, spinlocks will usually be held until they are explicitly released. Because Spinlock avoid overhead from operating system process re-scheduling or context switching, spinlocks are efficient if threads are only likely to be blocked for a short period. For this reason, spinlocks are often used inside operating system kernels including SMP Linux with PREEMPT_RT. However, spinlocks become wasteful if held for longer durations especially in a SMP configuration, as they may prevent other threads from running on the same or different core and require re-scheduling. The longer a lock is held by a thread, the greater the risk that the thread will be interrupted by the OS scheduler while holding the lock. If this happens, other threads will be left spinning (repeatedly trying to acquire the lock), while the thread holding the lock is not making progress towards releasing it. The result is an indefinite postponement until the thread holding the lock can finish and release it. Thus the use of spinlocks by non hard core RTOS such as SMP Linux with PREEMPT_RT in its kernel and implementing the various native system services, and protocol stack pose a serious performance issue for real time processes or threads on a multicore processor using SMP configuration.

The design goal has been to ensure that the processing load per core should not exceed 80 percent at the full rate system configuration, thereby leaving a 20 percent capacity for headroom to deal with any possible variations in processor occupancy load related to such a complex system design. The complexity arise from the use of a single partition SMP configuration that is served by a single instance of non hard core open source SMP Linux with PREEMPT_RT while supporting the data, and control planes of all configured application servers in the communication system on the same multicore processor.

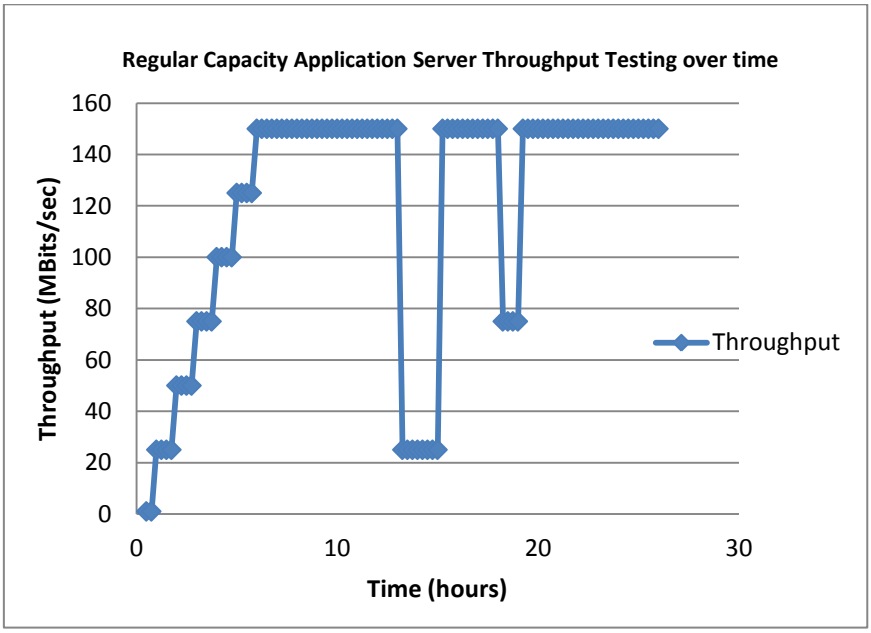


Figure 14: Throughput Testing Rate over a Period of Time

Figure 14 shows the throughput rate that a regular capacity application server was subjected to over a period of 26 hours of continuous stress testing. The rate was varied throughout this testing interval to simulate varying load conditions at different intervals of time. The rate was varied gradually from an

initial rate of 1 Mbits/sec (low system load) and varied gradually for each hourly run till it reaches a peak rate of 150 Mbits/sec (high system load) around the 6th hour, where it was kept steady for over 6 hours till it was varied up and down again.

Figure 15 shows the corresponding processing or processor occupancy load on the data plane core associated with the regular capacity application server under test in response to the system load or throughput it is supporting at that given instance of time. The key focus behind this kind of stress testing was to validate that the application server is capable of supporting varying system load with no degradation in throughput, and with high system availability for prolonged periods of time with no system instabilities such as loss of communication buffers, processor occupancy reaching 100% due to random unbounded latency spikes etc leading to application server outage.

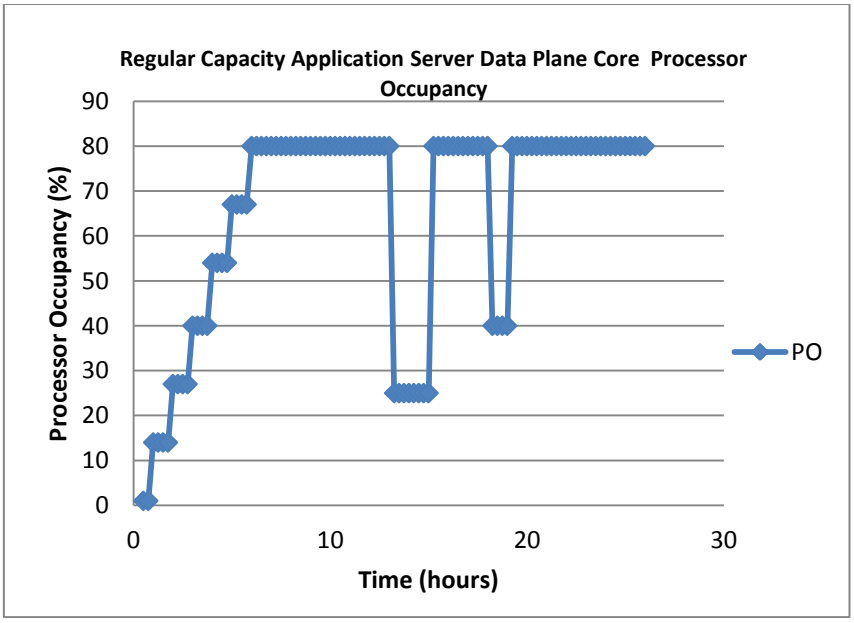


Figure 15: Processor Occupancy Load Corresponding To Throughput Rate

Extensive testing was also performed involving application server recovery scenarios to validate that an operational application server may be taken out of service and brought back up with no impact to other operational application servers configured in the system that shares the same single partition SMP configuration.

Chapter 4

4 Conclusions

4.1 Conclusions and Summary

In this dissertation, we have presented a study of the CSAT co-existence mechanism proposed by the LTE-U Forum for the harmonious co-existence of LTE and Wi-Fi in the unlicensed spectrum. We proposed a proactive small cell interference mitigation mechanism to improve the spectral efficiency of LTE networks in the unlicensed spectrum. We defined and presented a deployment scenario that highlighted the fact that even though using this CSAT co-existence mechanism will make LTE a fair neighbor to Wi-Fi, it will result in significant degraded overall system performance when multi-operator LTE-U small cells share the same unlicensed channel without any CSAT gating cycle co-ordination for LTE-U transmissions.

We proposed an unlicensed spectrum Inter Cell Interference Coordination (usICIC) mechanism to address this issue. We demonstrated via extensive simulation results that our proposed usICIC mechanism will result in 40% or more improvement in overall LTE-U system performance.

To make the LTE-U capable small cell energy efficient, an opportunistic LTE-U SDL mode of operation is used, where the 5 GHz LTE-U radios on the small cell are powered off, if the traffic pattern for the small cell is below a certain threshold that could easily be handled by the licensed spectrum. This is done to not only achieve a fair usage of the unlicensed spectrum (use only when needed), but to reduce the overall power consumption resulting in

recurring CAPEX savings, especially when there will be tens of thousands of LTE-U small cells deployed in the wireless HetNet of each service provider.

This dissertation has also presented the design for distributed and parallel processing system architecture for building a low cost, high capacity scalable, and energy efficient next generation communication system that leverages the unlicensed spectrum, and would result in significant CAPEX, and OPEX reductions for the service provider. The use of the unlicensed spectrum significantly increased (by a factor of two) the capacity of the wireless communication system in an extremely cost effective manner. The architecture utilized some radical design ideas that were never attempted before such as combining the control, and data planes of multiple application servers under a single D-SMP partition on a single multi-core processor to be served by a single instance of an open source RTOS. This resulted in significant reduction in device count (by a factor of 14), system size/footprint reduction, and lead to a scalable cost effective energy efficient system configuration solution that resulted in CAPEX and OPEX reduction for the service provider.

The drastic reduction in device count, and the use of dynamic power control mechanism that continuously modulate the clock speed of each processing core depending upon the traffic pattern of the wireless communication system throughout the day, makes the system extremely energy efficient. The resulting reduction in overall power consumption by the tens of thousands of wireless communications systems deployed in each of the service provider's network, not only help reduce their recurring OPEX, but

also help reduce the greenhouse gas emissions as a result of lower energy consumption, making it good for the environment.

The use of the more powerful processing cores on the multi-core processor, along with the use of the hardware acceleration engines for packet processing offload, allowed for an overall capacity increase by a factor of around 1.5 compared to the use of the previous generation single core processors.

Several system configurations using a multi-core processor were considered. Our proposed single partition D-SMP configuration offered the desired deterministic execution behavior of the S-AMP configuration, and the scalability, and cost savings of the SMP configuration, where a single open source RTOS instance is used irrespective of the number of application servers configured on the multi-core processor. This resulted in a reduction by a factor of 14 in the number of OS instances used for the seven application server configuration.

The decision to use a tier-1 OS supplier for the open source RTOS, reduced concerns about any indemnification issues that may result from the use of open source RTOS in the commercial product.

The use of open source RTOS even though contributed towards reducing the system COGS by eliminating the need to pay a tier-1 OS supplier a per instance RTOS deployment royalty fee, but opened the door for serious performance issues for the time sensitive data plane processes/threads. Consequently, the communication system was not able to provide high system availability for a prolonged period of time while maintaining the architected maximum system throughput. High performance

lock-less zero copy buffer management, and messaging services were introduced for the time sensitive data plane processes/threads to address the performance issues to ensure that the communication system could be commercially deployed.

The choice of a multi-core processor with more or less cores is generally driven by the size, and capacity requirements of the communication system. From a software portability perspective (build once, use many times) it was considered important to define a middleware layer CAL, that abstracts or hide the hardware specific details from the application software thus making future portability easier by adopting to different multi-core processors in future products with minimal impact to the application software.

Since the control, and data planes of all configured application servers in the communication system shared the same SMP partition and served by a single RTOS instance, application server recovery mechanism became quite complex. The new recovery mechanism has to be done without a multi-core processor reboot, and required elaborate system resource cleanup procedures to ensure that other operational application servers are not impacted, and the high system availability requirement of the next generation communication system is not compromised.

Extensive black box testing was needed and performed to alleviate any concerns surrounding the high system availability due to the use of such radical architectural design changes to ensure that a communication system utilizing such a system configuration is not just a theoretical concept, but rather fully capable of meeting the requirements needed for commercial deployment.

4.2 Future Research Directions

The research works presented in this dissertation may serve as a stepping stone towards 5G. The 5G air interface is anticipated to provide mobile broadband connectivity with even higher data rate that is currently possible by leveraging the unlicensed spectrum, to deliver higher quality of experience (QoE) to the users, and with a signaling interface that will allow connectionless MAC for Internet of Things (IoT) with billions of connected devices.

The new 5G millimeter-wave air interface is anticipated to be a line of sight carrier to be deployed on a small cell as a secondary carrier like the LTE Unlicensed (LTE-U) carrier in carrier aggregation (CA) mode with a primary carrier that will be anchored in 4G (LTE) licensed spectrum. 3GPP standardization activities for 5G are about to start with a target of commercial 5G network deployment in 2020.

Chapter 5

5 Appendix

5.1 Appendix A – VxWorks

VxWorks, [44] is an industry leading commercial off-the-shelf (COTS) embedded real time operating system (RTOS) developed as proprietary software by Wind River Systems of Alameda, California USA. It speeds time-to-market, lowers development risk, shortens schedules, and minimizes cost with leading capabilities in multicore, safety, security, and scalability.

VxWorks provide scalability, reliability, and rock-solid real-time performance, with comprehensive multicore processor support, including asymmetric multiprocessing (AMP) and symmetric multiprocessing (SMP) OS configurations and hardware-optimized multicore acceleration capabilities. Included in each VxWorks platform is, a fully 32-bit and 64-bit capable real-time operating system (RTOS), as well as Wind River development and multicore tools.

The powerful and flexible multicore capabilities mean that VxWorks is uniquely designed to optimize size, weight, power, and performance with next-generation intelligent embedded systems, and maximize the benefits of multicore devices.

The VxWorks SMP and AMP enabled platforms allow developers to deliver higher-performance multicore-powered products with reduced risk and development investment. It is capable of dealing with the most demanding

time constraints. VxWorks is a high-performance RTOS tuned for both determinism and responsiveness.

5.2 Appendix B – Open Source Linux with PREEMPT_RT

There are many methods for achieving real-time responsiveness in operating systems. Hard core real-time operating systems (RTOS), such as the Wind River VxWorks operating system [44], were designed specifically to solve this problem. VxWorks was designed from the ground up to be an RTOS. On the other hand, Linux was originally designed to be a general purpose operating system (GPOS).

There have been many attempts over the past several years to improve the real-time performance of Linux so that the benefits of Linux, such as open-source software availability, can be leveraged for real-time applications. The two basic methods of making Linux an RTOS are PREEMPT_RT and the Kernel Preemption Patches.

In the kernel preemption approach, the Linux kernel is modified to reduce the amount of time the kernel spends in non-preemptible sections of code before it can respond to interrupts and subsequent task scheduling operations. This approach requires cooperative modification of the Linux kernel to achieve minimized non-preemptive paths. Therefore, it is an ongoing effort and potentially requires revalidation whenever a kernel modification is made. If code that exceeds the non preemptible path limit of the kernel is found, it must be restructured to function within the preemption latency goals. Moreover, the process of finding non-preemptive paths of excessive length is empirical. It also requires exhaustive traversal of all kernel paths under all

load conditions in order to guarantee discovery of the worst-case scenario. This is realistically impossible. While competitive preemption latency figures are available, the complexity of the overall kernel makes an absolute guarantee unrealistic.

Many open-source projects have been formed to create a low-latency kernel using a variety of preemption approaches. The most significant project was started by Ingo Molnar and is being mainstreamed into the standard Linux kernel over time. Wind River includes this feature in their Wind River Linux platforms.

There are certainly benefits to this approach; probably the most notable is that the PREEMPT_RT project is widely considered to be mainstream Linux and the quality and maturity is improving over time. Many code paths in the kernel (such as device drivers) are being modified in mainline to support preemption. However there are still some limitations with PREEMPT_RT support particularly in code receiving less community coverage. This includes architecture-specific code and device drivers.

Therefore, there is variability in the level of performance available among Linux target architectures, and even dependencies per target board are possible. In many cases device drivers will port seamlessly into the PREEMPT_RT model. But preemptive performance, as well as correct operation of legacy and new device drivers, present ongoing validation issues. Thus the PREEMPT_RT feature delivers the benefit of being part of the mainstream Linux kernel development and is well suited to certain types of real-time application development such as audio or video streaming.

However, the feature is incapable of yielding 100 percent guaranteed real-time performance.

5.3 Appendix C – Critical Section of Code, and software locks

Concurrent computing is a form of computing in which programs are designed as collections of interacting computational processes that may be executed in parallel. Concurrent programs (processes or threads) can be executed on a single processor by interleaving the execution steps of each in a time-slicing way, or can be executed in parallel by assigning each computational process to one of the processing cores of a multi-core processor.

In concurrent computing, a critical section refers to a portion of software code that accesses a shared resource such as a device or a data structure that must not be concurrently accessed by more than one process or thread of execution. The critical section has to be guarded by a software lock that may be typically implemented via semaphore to achieve synchronization at the entry and exit of the critical section. As critical section usually terminates in fixed time, a process or thread has to wait a fixed time to enter it and thus experiences a bounded latency. However if the semaphore is acquired by a low priority non-real time process or thread, and then preempted by an interrupt prior to exiting the critical section, other high priority real time processes or threads that need access to the same critical section will have to endure a random unbounded latency spike since they will have to wait until the non-real time process or thread is scheduled to run again, and thus

releases the software lock or semaphore after finishing the execution of the critical section.

A spinlock is a lock which causes a thread trying to acquire it to simply wait in a loop or in other words spin while repeatedly checking if the lock is available. Since the thread remains active but is not performing a useful task, the use of such a lock is a type of busy waiting. Once acquired, spinlocks will usually be held until they are explicitly released, although in some implementations they may be automatically released if the thread being waited on (that which holds the lock) blocks, or goes to sleep.

Because they avoid overhead from operating system process re-scheduling or context switching, spinlocks are efficient if threads are only likely to be blocked for a short period. For this reason, spinlocks are often used inside operating system kernels. However, spinlocks become wasteful if held for longer durations, as they may prevent other threads from running and require re-scheduling. The longer a lock is held by a thread, the greater the risk is that the thread will be interrupted by the OS scheduler while holding the lock. If this happens, other threads will be left spinning as they will repeatedly try to acquire the lock, while the thread holding the lock is not making progress towards releasing it. The result is an indefinite postponement until the thread holding the lock can finish and release it. Thus the use of spinlocks by non hard core RTOS such as SMP Linux with PREEMPT_RT in implementing the various native system services, and protocol stack pose a serious performance issue for real time processes or threads on a multi-core processor using SMP configuration. To meet the stringent real time performance requirements custom lock less system services that employs

atomic operations may be used instead of the native services offered by SMP Linux with PREEMPT_RT as they cannot be guaranteed to be 100% lock free.

An operation performed on shared memory is atomic if it completes in a single step relative to other threads of concurrent execution. When an atomic store is performed on a shared variable, no other thread can observe the modification half-complete. When an atomic load is performed on a shared variable, it reads the entire value as it appeared at a single moment in time. Non-atomic loads and stores do not make those guarantees.

Without those guarantees, lock-free programming would be impossible, since we could never let different threads manipulate a shared variable at the same time. Thus at any time two threads operate on a shared variable concurrently, and one of those operations performs a write, both threads must use atomic operations. If this rule is violated and either thread uses a non-atomic operation, it results in a data race which is an undefined behavior that results in torn reads and torn writes leading to system instabilities such as software crashes and non deterministic system behavior.

5.4 Appendix D – P4080 BMAN Initialization

The hardware acceleration engine BMan manages up to 64 pools of buffers each containing a number of buffers of varying sizes on behalf of the software for both hardware (acceleration engines, and network interfaces) and software use, and provides hardware arbitration for simultaneous access for a buffer from multiple sources on various cores without the need of a software lock. Since no software locks are taken while attempting to acquire or release a buffer, therefore no process or thread will be blocked while waiting for a lock to be released.

At system initialization time, a custom kernel driver such as CAL_DPA_DRIVER acquires a contiguous chunk of external DDR memory that is in kernel space to serve as buffer memory. To avoid data copy, the custom kernel driver performs a kernel-to-user space mapping of this contiguous chunk of memory to allow the user space applications direct access to the kernel space buffer memory.

The custom kernel driver next initializes and programs the hardware acceleration engine BMan [31] with the number of buffer pools, the number of tokens (buffers) per pool, and the size of the buffers to which the tokens point to within each pool.

The custom kernel driver then carves the contiguous chunk of memory to seed the pools with tokens (memory address/pointers). The start address of the memory chunk is used as a base pointer when calculating the seed values for the tokens in each buffer pools. The seed address [35] for any given token within a specific buffer pool may be calculated by the following formula:

$$\text{Memory Address}_{x,y} = \text{Base\_Starting\_Address} + \sum_{k=1}^{x-1} (\text{Pool\_Size}_k * \text{Pool\_Buffer\_Size}_k) + ((y-1) * \text{Pool\_Buffer\_Size}_x)$$

Where x is the pool number, y is the buffer (or token) number within pool x, Pool_Size is the number of tokens (buffers) within pool k, and Pool_Buffer_Size is the size of each individual buffer within pool k (such as 1 KB).

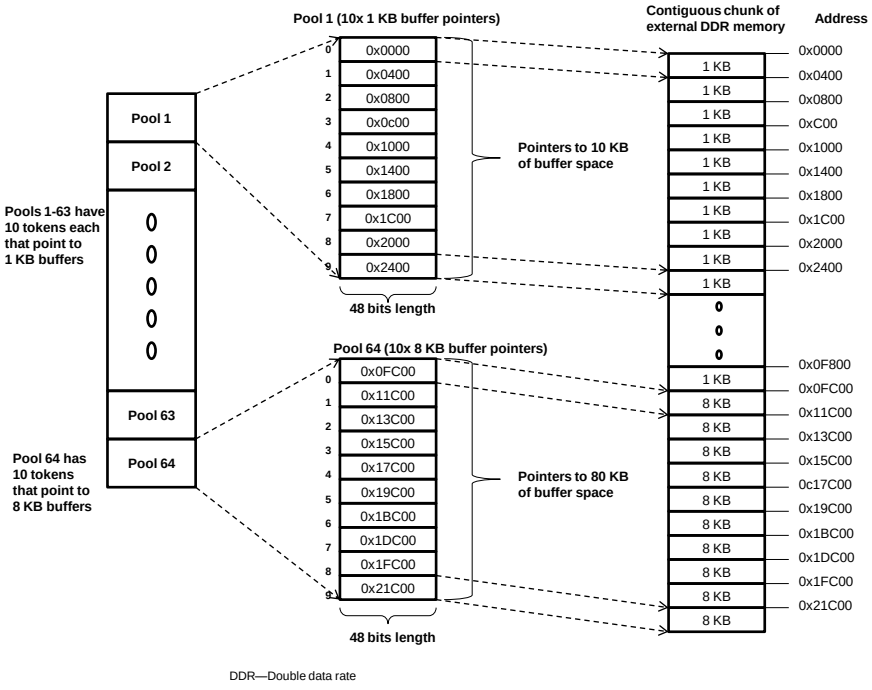


Figure 16: P4080 BMAN Buffer Pool Seeding

Figure 16 illustrates how the contiguous chunk of memory is carved into buffers for the various buffer pools that are used to seed/initialize the BMAN. For purposes of illustration it is assumed that 0x0000 is the start

address of the contiguous chunk of memory. Accordingly, using the equation listed above, the seed value for the first token (token 0) of the first pool (pool 1) that points to a buffer of 1 KB size is 0 (i.e., 0x000). The next token (token 1) of pool 1 is 1024 (i.e., 0x400), and so on. Assuming that pool 1 through pool 63 have ten tokens each that point to buffers of size 1 KB, and that pool 64 has ten tokens pointing to buffers of size 8 KB, the last pool (pool 64) will have a first token value (token 0) of 64,512 (i.e., 0xFC00) and the next token value (token 1) will have a value of 72704 (i.e., 0x11C00), and so on.

Once all the pools are seeded with the appropriate buffer tokens and the hardware acceleration engine BMan [31] properly programmed, it may then assist in implementing the lock-less zero copy multi-core buffer management scheme.

Once the initialization is done, whenever the buffer management scheme needs a buffer, such as when a packet arrives, the Bman [31] de-queues (i.e., takes out) a token (i.e., a pointer to a buffer memory) from the appropriate hardware pool. Similarly, when a buffer is released, the BMan [31] en-queues (i.e., puts back) the token (i.e., a buffer memory pointer) back onto the hardware pool.

5.5 Appendix E – Standard P4080 DPAA configuration for IP packet processing using the Protocol Stack

5.5.1 Ingress Control Plane IP Packet processing

Due to performance considerations typically the control plane, and data plane traffic are isolated to separate Ethernet interfaces. In our proposed architecture, the Linux protocol stack is bound and runs on the control plane core x. Figure 17 shows the P4080 DPAA configuration for ingress control plane IP packet processing. The CAL_INIT module sets up static PCD rules at system initialization time that allows the FMan to en-queue via the QMan API all packets arriving on the control plane Ethernet interface to an ingress frame queue (FQ) of channel X that is bound to the software portal of core X. The frame queue descriptor (FQD) for the packets destined for control plane uses the callback function “dispatcher” that is part of the standard BSP data path acceleration (DPA) driver provided by the OS vendor. This callback function when invoked as an interrupt service routine to service the de-queued entry from the front of the DQRR register, hands over the packets to the Linux protocol stack for processing, and the packet is eventually delivered to its appropriate user space destination such as OA&M (step 1 through step 7). A data copy is involved as the control plane packet crosses the kernel space to user space boundary and makes its ways through the Linux protocol stack before reaching its destination. The key take away from Figure 10 is that the ingress control plane traffic, and Linux protocol stack processing is confined to the control plane core X and the hardware acceleration engines (FMan, QMan, BMan) route the control plane packets to the protocol stack with no

performance impact to data plane cores that host the real time critical data plane process or application server.

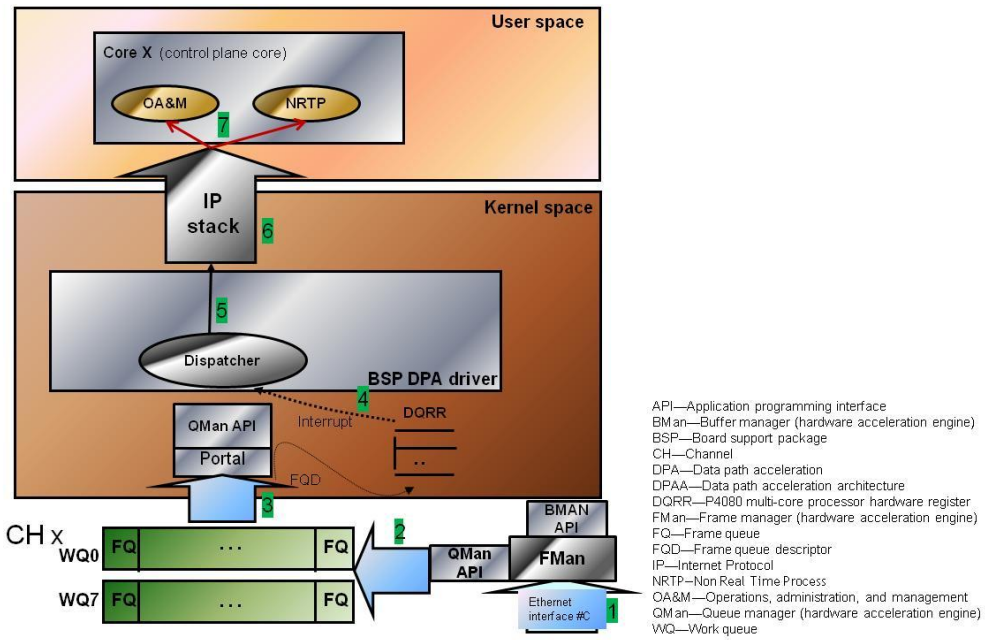


Figure 17: Ingress Control Plane IP Packet processing using Protocol Stack

5.5.2 Egress Control Plane IP Packet processing

Figure 18 illustrates the P4080 system configuration for egress control plane IP packet processing. The non real time user space applications such as OA&M running on control plane core x send TCP/UDP IP packets using the Linux protocol stack (step 1). The protocol stack after performing necessary packet processing, hands over the packet to the standard BSP DPA driver provided by the OS vendor (step 1'). The BSP DPA Driver first acquires a buffer via the BMan API (step 2 and step 3) and copies the packet it receives from the protocol stack into it, and release the kernel buffer that was acquired by the protocol stack. The BSP DPA driver, then en-queues the packet on the dedicated egress FQ of channel C via the QMan API (step 4

and step 5). The QMan determines the status of whether the en-queue operation was successful or not, and this response is sent back to the BSP DPA driver, that may try to repeat the en-queue operation if it fails for some reason (step 5'). The FMan, with help from QMan, processes all packets from the various FQ. The FMan de-queue the frames from channel C's FQ (step 6). FMan then transmits the packet onto the associated external Ethernet interface #C. Once the packet is transmitted, the FMan releases the buffer back to BMan (step 7).

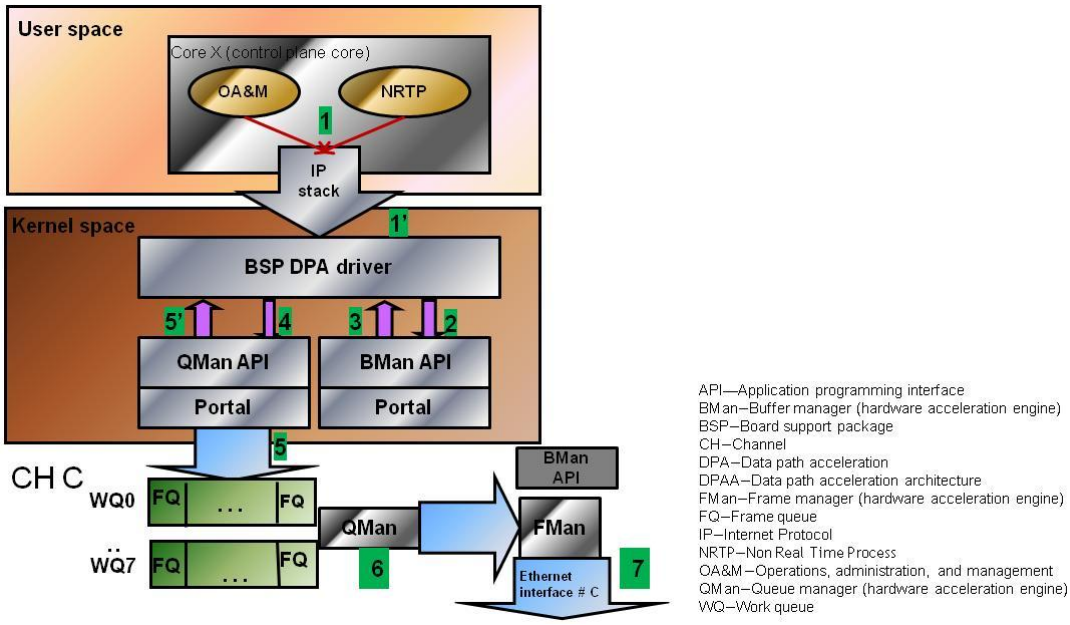


Figure 18: Egress Control Plane IP packet Processing using the Protocol Stack

5.6 Appendix F – P4080 DPAA configuration for IP packet processing without the Protocol Stack [36]

5.6.1 Ingress Data Plane IP Packet processing

Figure 19 shows a P4080 system configuration for processing IP packets while bypassing the protocol stack. The user space process “DL Real Time Process“ runs in an endless “run to completion” mode and wakes up every transmission time Interval (TTI) (such as 1 msec), performs its necessary processing, and then goes back to sleep. Each instance of the data plane process or application server is mapped to a particular data plane core, and has a kernel to user space (K-U) mapped ring of buffer descriptors (RBD) with a unique ring ID. The K-U mapping of RBD allow the user space (such as CAL_MSG), and kernel space (such as CAL_DPA_DRIVER) components of CAL to access it without any addressing problems, and the need for data copy. CAL_MSG provides an API to the application such as data plane process or application server to register itself with CAL_MSG at cell initialization time. During this time the FMan is programmed by the CAL_DPA_DRIVER with a dynamic PCD rule that binds the IP address of each data plane process or application server to a specific hardware frame queue (FQ) of a channel Y that is bound to the software portal of data plane core Y. A mapping association is also established between the same ingress FQ, and the ring of buffer descriptors (RBD) of the data plane process or application server.

At a high level as shown in Figure 19, when the FMan receives a TCP/UDP IP data packet (step 1) it acquires a suitable size kernel-user space mapped buffer from BMan and copies the received packet into the buffer, the

FMan examines the packet IP address, and determines as per the configured dynamic PCD rules which ingress FQ to en-queue the packet, and uses the QMan API to en-queue the IP packet onto the appropriate FQ (step 2). The QMan handles all the FQ for the various channels. The WQ are there to implement quality of service (QoS). So each arriving IP packet once en-queued onto a FQ goes through internal hardware queuing within the QMan. The FQ that make it to the front of the queue are ready for processing, and are en-queued into the DQRR register automatically by the QMan as space becomes available in that hardware register (step 3). The DQRR register has a depth of 15 entries, where each entry is a frame queue descriptor (FQD) that contains information about the FQ, such as the pointer to the IP packet buffer, an associated callback function, etc. As each FQD makes it to the front of the DQRR register, the front entry is automatically de-queued and a portal interrupt for the specified core (such as core Y) is generated and the associated callback function “Fast_Path_Dispatcher” which is part of our custom CAL_DPA_DRIVER is invoked as an interrupt service routine (ISR) in kernel space to process the frame queue descriptor (FQD) (step 4).

The callback function “Fast_Path_Dispatcher” determines using a mapping table, which ring of buffer descriptors (RBD) this particular frame queue descriptor (FQD) is mapped, and creates a buffer descriptor (BD) by copying the kernel-user space mapped buffer pointer (along with any book-keeping info) and en-queue the buffer descriptor on to the appropriate kernel-user space mapped ring of buffer descriptors (RBD) for later processing by the associated user space data plane process or application server on a particular data plane core Y (step 5).

The user space process or application server runs in an endless “run to completion” mode. The real time process wakes up every TTI (e.g., 1 msec), and invokes the CAL_MSG API to retrieve any packets that are en-queued on the ring of buffer descriptors (RBD) for its consumption (step 6). The CAL_MSG retrieves the buffer descriptors that are en-queued on the particular ring of buffer descriptors (RBD) (step 7), and delivers them to the user space process or application server for processing (step 8). The packet headers at the various layers (TCP/UDP/IP/Ethernet) are used for routing the packet payload to its eventual destination. Typically as the packet makes it way up the protocol stack the various protocol headers at different layers (L2, layer 3 (L3)) get stripped away from the packet, and only the packet payload is eventually delivered to the destination user waiting for it. Since the received ingress data plane TCP/UDP IP packet did not go through the standard protocol stack, the user space process or application server receives the TCP/UDP IP packet along with all the various headers (TCP/UDP/IP/Ethernet). The user space process or application server then extracts and operates on the packet payload. When done, it releases the buffer using the CAL_BUF APIs (not shown in the figure). When there are no more buffer descriptors containing IP packets en-queued on the RBD for the user space process or application server to process, the user space process or application server goes back to sleep. Since there is only one producer (CAL_DPA_DRIVER), and one consumer (a specific user space process or application server) for each ring of buffer descriptors (RBD), this scheme could be implemented with atomic operations that do not require any software locks. The use of kernel to user space mapped buffers, and ring of buffer

descriptors also ensures that there is no data copy needed as the packet traverses from the kernel space device drivers to its eventual destination in user space.

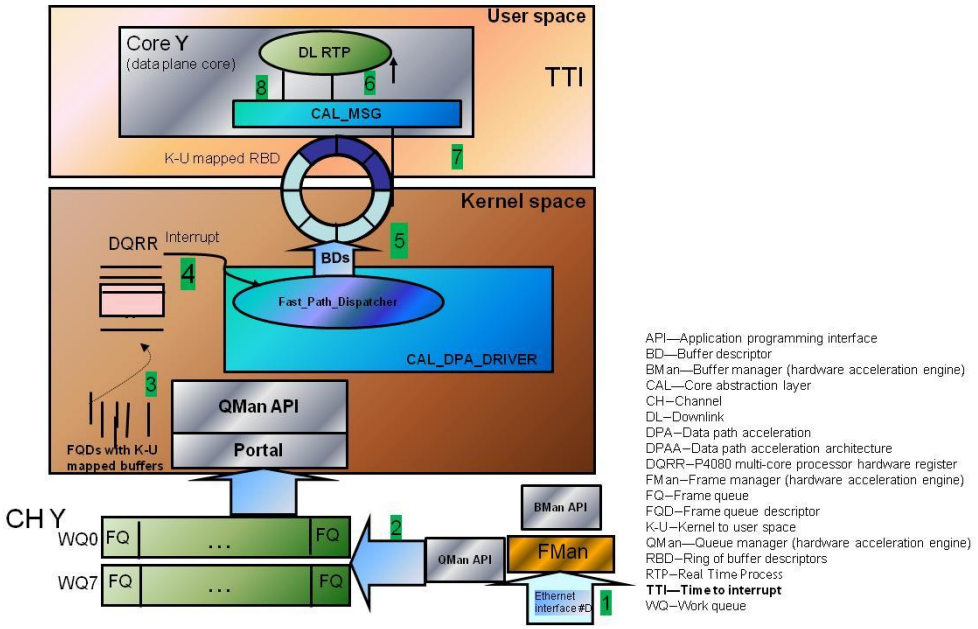


Figure 19: Ingress Data Plane IP Packet Processing

5.6.2 Egress Data Plane IP Packet processing

Figure 20 shows our proposed data plane egress fast path architecture. When the user space UL Real Time process or application server needs to transmit a TCP/UDP IP packet on the external interface it employs our CAL_MSG service that does not use the Linux IP protocol stack and thus does not subject the real time data plane application to any un-bounded latency spikes as mentioned previously.

As shown in Figure 20 when the data plane user space application needs to transmit a TCP/UDP IP packet, it first acquires a kernel to user space mapped buffer from CAL_BUF (step 1 through step 6). When it is ready

to send data it will populate the data payload and the generated appropriate headers (TCP/UDP/IP/Ethernet) into the buffer. The packet header needs to be generated, and appended because the TCP/UDP IP packet does not go through any protocol stack for its transmission, and the appropriate header has to be added to the packet payload so that it may be processed properly once it gets transmitted out of the external interface. The user space process then invokes the CAL_MSG APIs. The CAL_MSG module with help from CAL_DPA_DRIVER will en-queue the packet onto a FQ on the target channel (CH D) that is associated with the external Ethernet interface #D that is dedicated to carry only the data plane traffic (step 7 through step 10). This packet en-queuing is performed without the use of the protocol stack and instead relies on a lock-less, zero copy messaging scheme to avoid unbounded latency spikes. The FMan, with help from QMan, processes all packets from the various FQ in a hardware efficient manner that has no performance penalty for each individual core (step 11). After de-queuing frames from the channel D's FQ, FMan transmits the packet on the external Ethernet interface #D. Even though there is a context switch involved whenever the packet is traversing the user-space to kernel space boundary, there is no data copy involved. Once the packet is transmitted, the FMan releases the buffer back to BMan (step 12). Step 13 through step 15 are optional steps where the QMan determines the status of whether the en-queue operation was successful or not, and this response is sent all the way back to the user space application if it so desires to receive the status.

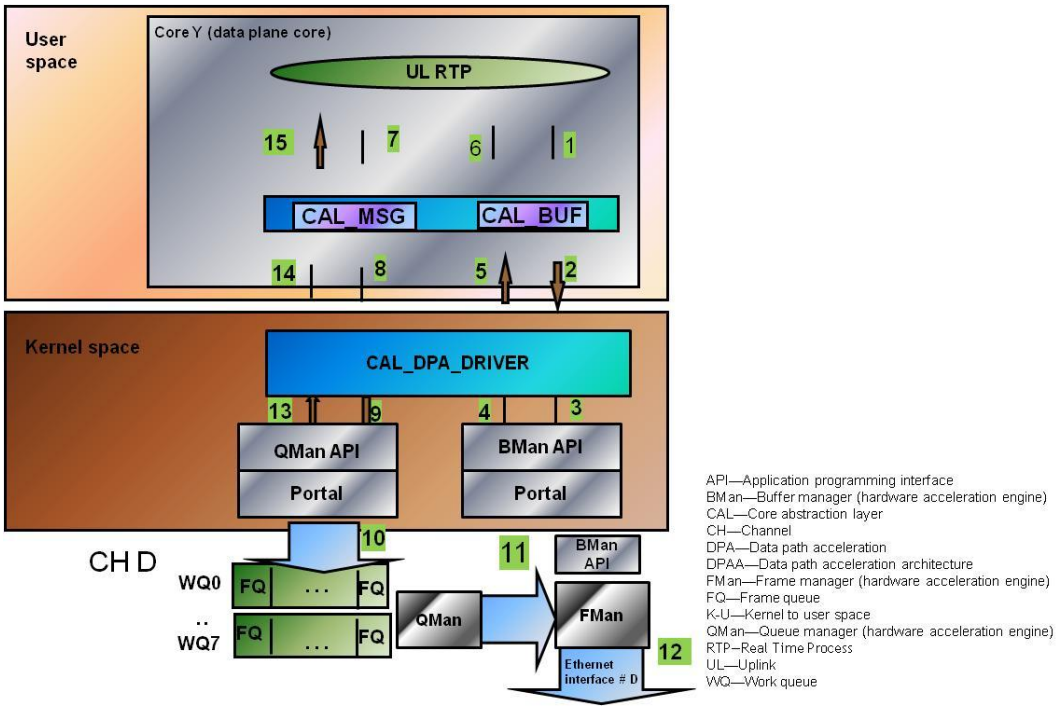


Figure 20: Egress Data Plane IP Packet Processing

5.7 Appendix G – P4080 QMAN Configuration [37]

By default the P4080 hardware acceleration engine QMAN [31] is configured to run in “interrupt mode”. In this mode whenever a frame queue descriptor (FQD) entry containing ingress TCP/UDP IP packet, is automatically de-queued from the front of the DQRR register a portal interrupt for the data plane core is generated, and the associated callback function is executed on the core in kernel space to process the FQD containing the ingress TCP/UDP IP packet. Each such interrupt may result in a context switch (user-to-kernel-to-user space) as the data plane process or application server runs in user space on the data plane core. A high number of ingress data plane packets therefore translate into an equally high number of interrupts and consequently result in an appreciable total context switching

overhead that consumes precious processing resources on the data plane core.

One design alternative to reduce this significant context switching overhead is to use “interrupt coalescing mode” where up to a maximum of fifteen (the depth of the DQRR register) data packets are handled by the callback in one interrupt service routine (ISR).

However an even better performance optimization design alternative is to configure the QMan to run in polling mode [37] as opposed to the default interrupt mode. In this mode the QMan pushes the packet through its internal hardware queue until the packet reaches the front of the hardware queue. However, in polling mode the hardware will NOT generate any portal interrupts to the data plane core. A kernel thread is created as part of the CAL_DPA_DRIVER that wakes up periodically (every ΔT time) and invokes the QMan API to poll for any queued-up frame queue descriptors (FQDs) that have reached the front of the hardware queue and subsequently get “N” FQD entries de-queued from the QMan. Where “N” represents a tunable parameter (e.g., N=100) and indicates the number of FQD entries from the QMan that the kernel thread will service every time it “wakes up,” and thus reduce context switching overhead by a factor of N compared to the default interrupt mode.

Each context switch results in an appreciable processing overhead. By eliminating this unnecessary processing overhead, more processing time and power will become available on the data plane cores to serve the real time processes and threads for increased capacity.

Chapter 6

6 Bibliography

[1] “LTE in unlicensed spectrum – 3GPP”. [Online]. Available: http://www.3gpp.org/news-events/3gpp-news/1603-lte_in_unlicensed

[2] FCC 14-30. [Online]. Available: http://www.fcc.gov/document/5-ghz-u-nii-ro

[3] “3GPP TS 36.420 V8.0.0 (2007-12)”. [Online]. Available: http://www.qtc.jp/3GPP/Specs/36420-800.pdf

[4] 3GPPTR 36.872 Version 12.1.0 [Online]. Available: http://www.3gpp.org/DynaReport/36872.htm

[5] Qualcomm Research, “LTE in Unlicensed spectrum: Harmonious coexistence with Wi-Fi”, June 2014, [Online]. Available: https://www.qualcomm.com/media/documents/files/lte-unlicensed-coexistence-whitepaper.pdf

[6] Huawei, “U-LTE: Unlicensed Spectrum Utilization of LTE”, [Online]. Available: http://www.google.com/url?url=http://www.huawei.com/ilink/en/download/HW_327803&rct=j&frm=1&q=&esrc=s&sa=U&ei=ctJpVMuRNoutyQTti4H4Dg&ved=0CCAQFjAA&sig2=daydago9X4cIOwTEANZQFA&usg=AFQjCNE1vpiFW971BnmhVJUHoJBCv-VGhg

[7] Huawei, “LTE Small Cell v.s. WiFi User Experience”, [Online]. Available: http://www.google.com/url?url=http://www.huawei.com/ilink/en/download/HW_323974&rct=j&frm=1&q=&esrc=s&sa=U&ei=XctvVLXNB6e1sAS

L04KgDw&ved=0CBQQFjAA&sig2=iYJ6GP-jEpgvCUYtwucRqA&usg=AFQjCNH7gfmfhYGYOeNibEyNwkcMn3F-mA

[8] F. M. Abinader *et al.*, “Enabling the Coexistence of LTE and Wi-Fi in Unlicensed Bands”, in IEEE Communications Magazine (Future Of WI-FI), November 2014

[9] E. Almeida *et al*., “Enabling LTE/WiFi coexistence by LTE blank subframe allocation”, in Proceedings of IEEE International Conference on Communications (ICC), June 2013

[10] A. M. Cavalcante et al., “Performance Evaluation of LTE and Wi-Fi Coexistence in Unlicensed Bands,” Proc. IEEE 77th VTC 2013-spring, Dresden, Germany, June 2013

[11] F. S. Chaves *et al*., “LTE UL power control for the improvement of LTE/Wi-Fi coexistence,” in Proceedings of IEEE Vehicular Technology Conference, Sep. 2013

[12] T. Nihtil *et al*., “System Performance of LTE and IEEE 802.11 Coexisting on a Shared Frequency Band,” IEEE Wireless Communications and Networking Conference (WCNC), Apr. 2013

[13] IEEE 802.11-2007, IEEE Standard for Local and Metropolitan Area Networks—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-2007, (Revision of IEEE Std 802.11-1999), June 12, 2007

[14] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, “LTE-Advanced: next-generation wireless broadband technology

[invited paper]," Wireless Communications, IEEE, vol. 17, no. 3, pp. 10–22, 2010

[15] J. Peha, "Approaches to spectrum sharing," Communications Magazine, IEEE, Vol. 43, no. 2, pp. 10-12, 2005

[16] R. Ratasuk, M. A. Uusitalo, N. Mangalvedhe, A. Sorri, S. Iraji, C. Wijting, and A. Ghosh, "License-exempt LTE deployment in heterogeneous networks," in Wireless Communication Systems, 2012. ISWCS 2012. Proceedings of 9th International Symposium. IEEE, 2012

[17] T. Nihtil¨a, J. Turkka, and I. Viering, "Performance of LTE self-optimizing networks uplink load balancing," in Proceedings of the 73rd IEEE Vehicular Technology Conference (VTC), Budapest, Hungary, May 2011

[18] E. H. Ong, J. Kneckt, O. Alanen, Z. Chang, T. Huovinen, and T. Nihtil¨a, "IEEE 802.11ac: Enhancements for very high throughput WLANs", in 22nd IEEE Personal Indoor and Mobile Radio Communications (PIMRC), Toronto, Canada, September 2011

[19] Z. Chang, O. Alanen, T. Huovinen, T. Nihtil¨a, E. H. Ong, J. Kneckt, and T. Ristaniemi, "Performance analysis of 802.11ac DCF with hidden nodes," in 75th IEEE Vehicular Technology Conference (VTC), Yokohama, Japan, May 2012

[20] G. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X.P. Costa, and B. Walke, "The IEEE 802.11 universe", IEEE Communications Magazine, 48(1):62–70, January 2010

[21] IEEE 802.11n 2009. Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, Amendment 5: Enhancements for higher throughput. October 2009

[22] D. Skordoulis, Q. Ni, H. Chen, A. P. Stephens, C. Liu, and A. Jamalipour. "IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs", IEEE Wireless Communications, 15(1):40–47, February 2008

[23] LTE-U Forum. [Online]. Available: http://www.lteuforum.org/

[24] "LTE-U Forum: LTE-U Technical Report". [Online]. Available: http://www.lteuforum.org/documents.html

[25] "LTE-U Forum: LTE-U SDL Coexistence Specification". [Online]. Available: http://www.lteuforum.org/documents.html

[26] "LTE-U Forum: UE Minimum Requirements for LTE-U SDL". [Online]. Available: http://www.lteuforum.org/documents.html

[27] "LTE-U Forum: eNB Minimum Requirements for LTE-U SDL". [Online]. Available: http://www.lteuforum.org/documents.html

[28] A. Goldsmith, S. A. Jafar, I. Maric´, and S. Srinivasa, "Breaking Spectrum Gridlock With Cognitive Radios: An Information Theoretic Perspective", Proceedings of the IEEE, Vol. 97, No. 5, May 2009,Pages 894 – 914

[29] A Guha, and V. Ganapathy, "Power Allocation Schemes for Cognitive Radios", Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on 6-10 Jan. 2008, Pages 51 – 56

[30] H. Lee¤, K. Han, Y. Hwang, and S. Choi, “Opportunistic Band Sharing for Point-to-Point Link Connection of Cognitive Radios”, Cognitive Radio Oriented Wireless Networks and Communications, 2009. CROWNCOM '09. 4th International Conference on Date 22-24 June 2009, Pages 1-6

[31] Freescale Semiconductor, “QorIQ™ P4080 Communications Processor Product Brief,” Doc. Number P4080PB, Rev. 1, 2008, <http://cache.freescale.com/files/32bit/doc/prod_brief/P4080PB.pdf>

[32] J. Landry, “The Freescale Embedded Hypervisor,” Freescale Semiconductor, Nov. 2010, <http://cache.freescale.com/files/training/doc/dwf/AMF_ENT_T0616.pdf>

[33] M. R. Khawer, S. Easwaran, Patent Title: “Apparatus For Multi-Cell Support In A Network”; U.S. Patent #: US8634302; Patent Filing Date:, July 30, 2011; Patent grant date: January 21, 2014

[34] M. R. Khawer, L. So, Patent Title: “Core Abstraction Layer For Telecommunication Network Applications”; Korea Patent #: JP5759006; Patent Filing Date:, September 29, 2011; Patent Grant Date: June 12, 2015

[35] M. R. Khawer, L. So, Patent Title: “Lock-Less Buffer Management Scheme For Telecommunication Network Applications”; U.S. Patent #: US8737417; Patent Filing Date:, October 28, 2011; Patent Grant Date: August 06, 2013

[36] M. R. Khawer, L. So, Patent Title: “Lock-Less And Zero Copy Messaging Scheme For Telecommunication Network Applications”;

U.S. Patent #: US8737417; Patent Filing Date:, November 12, 2011; Patent grant date: May 27, 2014

[37] M. R. Khawer, M. Abulius, Patent Title: “Method And System For Improved Multi-Cell Support On A Single Modem Board”; U.S. Patent #: US8861434; Patent Filing Date:, November 28, 2011; Patent Grant Date: October 14, 2014

[38] M. R. Khawer, M. Abulius, Patent Title: “Method And System For Cell Recovery In Telecommunication Networks”; U.S. Patent #: US8730790; Patent Filing Date:, November 18, 2011; Patent Grant Date: May 20, 2014

[39] M. R. Khawer, “Method And System For Dynamic Power Control For Base Stations”; U.S. patent publication #: US 2013/0017854 A1, Publication Date: January 17, 2013

[40] M. R. Khawer, “Design Strategies for the Application Server Architecture/Configuration (and Its Functions) in Next-Generation Communication Systems”; IEEE Communication Magazine – Design & Implementation Series (Volume: 53, Issue: 4, April 2015), Pages 227 – 233

[41] N. Litayem and S. B. Saoud, “Impact of the Linux Real-Time Enhancements on the System Performances for Multi-Core Intel Architectures,” Internat. J. Comput. Applic., 17:3 (2011), 17–23

[42] G. Seiler, “Wind River Linux and VxWorks Real-Time Capabilities: A Comparison,” Wind River Systems White Paper, 2007, <http://www.windriver.com/whitepapers/guestwp/servefile.php?f=rt_linux_vxworks_comparison_whitepaper.pdf>

[43] Wind River Systems, “Wind River Hypervisor,” 2011, <http://www.windriver.com/products/product-overviews/PO_Hypervisor_0611.pdf>

[44] Wind River Systems, “Wind River VxWorks Platforms,” 2013, <http://www.windriver.com/products/product-overviews/PO_VE_6_9_Platform_0211.pdf>

[45] M. R. Khawer, J. Tang, F. Han, “usICIC – A Proactive Small Cell Interference Mitigation Strategy for Improving Spectral Efficiency of LTE Networks in the Unlicensed Spectrum”, accepted for publication in the upcoming edition of the IEEE Transactions on Wireless Communications

[46] LTE-U Forum Technical Contribution Titled: “Multi Operator LTE-U Co-Channel Co-Existence Scenario: Need for Explicit Inter Cell Interference Co-Ordination Mechanism”, December 02, 2014, Dallas Tx

[47] S. Bu, and F. Richard Yu, “Green Cognitive Mobile Networks with Small Cells for Multimedia Communications in the Smart Grid Environment,” IEEE Transactions on Vehicular Technology., vol. 63, no. 5, pp. 2115-2126, June 2014

[48] D. Xiao, X. Yu; D. Yang, “ A Novel Downlink ICIC Method Based on User Position in LTE-Advanced Systems”, IEEE Vehicular Technology Conference (VTC Fall), 2012, Pages: 1 - 5, DOI: 10.1109/VTCFall.2012.6399351

[49] S. Wang, Y. Zhang, GuangguoBi , “A decentralized downlink dynamic ICIC method for multi-cell OFDMA system”, International

Conference on Wireless Communications and Signal Processing (WCSP), 2011, Pages: 1 - 5, DOI: 10.1109/WCSP.2011.6096797

[50] Q Li, D. Xiao; D. Yang, “A downlink ICIC method based on priority in LTE-Advanced systems”, 4th IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT), 2011, Pages: 498 - 501, DOI: 10.1109/ICBNMT.2011.6155984

[51] W. Wong, T. Neoh, C. Loo, Y. Liew, E. Lee, “Implementation of asymmetric multiprocessing framework in humanoid robot”, IEEE Conference on Robotics Automation and Mechatronics (RAM), 2010, Pages: 188 - 193, DOI: 10.1109/RAMECH.2010.5513190

[52] W. Smit, “The architectural design of DSP based embedded parallel processors”, Africon Conference in Africa, 2002. IEEE AFRICON. Pages: 323 – 326, Vol.1, DOI: 10.1109/AFRCON.2002.1146856

[53] J. Sakai, I. Hiroaki, S. Torii, M. Edahiro, “Multitasking Parallel Method for High-End Embedded Appliances”, Micro, IEEE, Volume: 28, Issue: 5, Pages: 54 - 62, DOI: 10.1109/MM.2008.75

[54] Y. Qingbo; B. Yungang; C. Mingyu; S. Ninghui, “A Scalability Analysis of the Symmetric Multiprocessing Architecture in Multi-Core System”, IEEE International Conference on Networking, Architecture, and Storage, NAS 2009. Pages: 231 - 234, DOI: 10.1109/NAS.2009.51

[55] R. Vincke. N. De Witte, S. Van Landschoot, E. Steegmans, J. Boydens, “Writing Parallel Embedded Software Effectively”, International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro , Pages: 474 - 477, DOI: 10.1109/PDP.2014.117

Chapter 7

7 Abbreviations, Acronyms, and Terms

API — Application programming interface

AP — Access Point

BD — Buffer descriptor

Bman — Buffer manager (hardware acceleration engine)

BSP — Board support package

CA — Carrier Aggregation

CAL — Core abstraction layer

CAL_BUF — CAL buffer module

CAL_INIT — CAL initialization module

CAL_MSG — CAL messaging module

CAPEX — Capital Expanse

CH — Channel

COGS — Cost of goods sold

COTS — Commercial Off The Shelf

CPU — Central processing unit

CSAT — Carrier Sense Adaptive Transmission

DDR — Double data rate

DFS — Dynamic Frequency Scaling

DL — Downlink

DMA — Direct memory access

DPA — Data path acceleration

DPAA — Data path acceleration architecture

DPC	— Dynamic Power control
DQRR	— P4080 multi-core processor hardware Register
DSP	— Digital signal Processor
D-SMP	— Deterministic SMP
FIFO	— First in first out
FMan	— Frame manager (hardware acceleration engine)
FQ	— Frame queue
FQD	— Frame queue descriptor
FSL	— Freescale Semiconductor Limited
GPL	— General Public License
GPOS	— General Purpose Operating System
IP	— Internet Protocol
ISR	— Interrupt service routine
K-U	— Kernel to user space mapped
KPI	— Key Performance Indicator
L2	— Layer 2
L3	— Layer 3
LAA	— Licensed Assisted Access
LBT	— Listen Before Talk
LTE	— Long Term Evolution (A 4G wireless interface)
LTE-U	— LTE in Unlicensed Spectrum
MAC	— Media Access Control
Msec	— Millisecond
NRTP	— Non Real Time Process
OA&M	— Operation, administration, and maintenance

OPEX — Operational Expanses

OS — Operating System

P4080 — Freescale Semiconductor’s multi-core processor

PCell — Primary Cell

PCD — Parsing, Classifying, and Distributing

Qman — Queue manager (hardware acceleration engine)

QoS — Quality of service

QoE — Quality of Experience

RBD — Ring of buffer descriptors

RTOS — Real time operating system

RTP — Real Time Process

S-AMP — Supervised AMP

Scell — Secondary Cell

SDL — Supplemental Downlink

SMP — Symmetric multi-processing

TCP — Transmission Control Protocol

TDM — Time Division Multiplexing

TTI — Transmission Time Interval

UDP — User Datagram Protocol

UL — Uplink

UE — User Equipment

usICIC — unlicensed spectrum Inter Cell Interference Coordination

WQ — Work queue

8 Curriculum Vitae

Mohammad R. Khawer
Mohammad.khawer@alcatel-lucent.com

MAJOR

An innovator with a major in computer and information science engineering (CISE), with nineteen years of wireless industry experience in designing cutting edge 4G, 3G, and 2G commercial wireless base-station products.

EDUCATION

- Ph.D., Computer & Information Science Engineering
  December 2015 Syracuse University, New York, USA
- M.S., Computer & Information Science
  May 1995 Syracuse University, New York, USA
- B.E., Computer Systems Engineering
  December 1992, N.E.D. University of Engineering & Technology, Karachi, Pakistan

PROFESSIONAL EXPERIENCE

- **Distinguished Member of Technical Staff - Wireless Product Division - Alcatel-Lucent (August 1996 – Present)**
  - Currently involved in the forward looking 5G design activities as a member of the advanced Performance group in the wireless CTO organization of Alcatel-Lucent (June 2015 – present)
  - Represented Alcatel-Lucent on the LTE-U Forum, and made significant contributions towards the standardization of LTE

deployment in the Unlicensed Spectrum (LTE-U), as well as towards the on-going Licensed Assisted Access (LAA) R13 standardization activities in 3GPP (2014 – present)

- Developed Visual Simulator for use in advanced wireless system modeling, and simulation for use by wireless CTO, Bell Labs research, and Alcatel-Lucent services teams (2013 - present)
- Served as the lead platform architect and SCRUM Master for the development of the 4G LTE wireless Outdoor Small Cell (2011-2012)
- Served as the lead platform architect for the development of the 4G LTE wireless base-station (2008-2011)
- Served as the platform architect, and Feature Engineer for the development of 3G CDMA 1xEVDO wireless base-station (2004 – 2008)
- Served as the lead platform developer, and Feature Engineer for the development of 3G CDMA 3g1x wireless base-station (2001 – 2004)
- Served as the lead platform developer, and Feature Engineer for the development of 2G TDMA Microcell base-station (1996 – 2001)

- **Lead Software Developer - Sapient Corporation (May 1995 - August 1996)**
    - Served as the architect, and development team leader at Sapient Corporation to design, implement, and deploy the first internet based banking system in the world

PATENTS

[1] M. R. Khawer, “Software Circuit Switching Routers Using anchor Channels”, U.S. patent publication #: US 6,885,743 B2, Patent Grant Date: April 26, 2005

[2] M. R. Khawer, P. B. Pillai; “Dynamic Power Control For A TDMA based Air Interface“, U.S. patent publication #: US 7,058,029 B2, Patent Grant Date: June 6, 2006

[3] M. R. Khawer, P. Lopreiato; “Method For Provisioning A Permanent Virtual Circuit in an ATM Network“, U.S. patent publication #: US 7,292,599, Patent Grant Date: November 06, 2007

[4] M. R. Khawer, M. H. Kraml, S. G. Pisano, T. S. Yang; “Multi-Link Load Balancing For Reverse Link Backhaul Transmission” , U.S. patent publication #: US 8,325, 735 B2, Patent Grant Date: December 4, 2012

[5] M. R. Khawer, S. Easwaran, Patent Title: “Apparatus For Multi-Cell Support In A Network”; U.S. Patent #: US8634302; Patent Filing Date:, July 30, 2011; Patent grant date: January 21, 2014

[6] M. R. Khawer, L. So, Patent Title: “Core Abstraction Layer For Telecommunication Network Applications”; Korea Patent #: JP5759006; Patent Filing Date:, September 29, 2011; Patent Grant Date: June 12, 2015

[7] M. R. Khawer, L. So, Patent Title: “Lock-Less Buffer Management Scheme For Telecommunication Network Applications”; U.S. Patent #: US8737417; Patent Filing Date:, October 28, 2011; Patent Grant Date: August 06, 2013

[8] M. R. Khawer, L. So, Patent Title: “Lock-Less And Zero Copy Messaging Scheme For Telecommunication Network Applications”; U.S. Patent #: US8737417; Patent Filing Date:, November 12, 2011; Patent grant date: May 27, 2014

[9] M. R. Khawer, M. Abulius, Patent Title: “Method And System For Improved Multi-Cell Support On A Single Modem Board”; U.S. Patent #: US8861434; Patent Filing Date:, November 28, 2011; Patent Grant Date: October 14, 2014

[10] M. R. Khawer, M. Abulius, Patent Title: “Method And System For Cell Recovery In Telecommunication Networks”; U.S. Patent #: US8730790; Patent Filing Date:, November 18, 2011; Patent Grant Date: May 20, 2014

[11] M. R. Khawer, “Method And System For Dynamic Power Control For Base Stations”; U.S. patent publication #: US 2013/0017854 A1, Publication Date: January 17, 2013

[12] M. R. Khawer, L. So; “Core Abstraction Layer For Telecommunication Network Application”; U.S. patent publication #: US 2012/00093047 A1, Publication Date: April 19, 2012

[13] M. R. Khawer, L. So, Patent Title: “Lock-Less And Zero Copy Messaging Scheme For Telecommunication Network Applications”; Korea Patent #: KR101529895; Patent Filing Date:, October 12, 2011; Patent Grant Date: June 12, 2015

[14] M. R. Khawer, L. So, Patent Title: “Lock-Less And Zero Copy Messaging Scheme For Telecommunication Network Applications”;

Japan Patent #: JP5726316; Patent Filing Date:, October 19, 2011; Patent Grant Date: April 10, 2015

[15] M. R. Khawer, M. Abulius, Patent Title: “A Method And System For Improved Multi-Cell Support On A Single Modem Board”; Japan Patent #: JP5726320; Patent Filing Date:, October 27, 2011; Patent Grant Date: April 10, 2015

[16] M. R. Khawer, M. Abulius, Patent Title: “A Method And System For Improved Multi-Cell Support On A Single Modem Board”, Taiwan Patent #: TWI479850; Patent Filing Date:, November 23, 2011; Patent Grant Date: April 01, 2015

[17] M. R. Khawer, M. Abulius, Patent Title: “A Method And System For Cell Recovery In Telecommunication Network”; Japan Patent #: JP5710017; Patent Filing Date:, November 01, 2011; Patent Grant Date: March 13, 2015

[18] M. R. Khawer, M. Abulius, Patent Title: “A Method And System For Improved Multi-Cell Support On A Single Modem Board”; Korea Patent #: KR101502808; Patent Filing Date:, October 27, 2011; Patent Grant Date: March 10, 2015

[19] M. R. Khawer, M. Abulius, Patent Title: “A Method And System For Improved Multi-Cell Support On A Single Modem Board”; Patent Case #: 808249-FR-EPT; France Patent #: EP2647163; Patent Filing Date:, October 27, 2011; Patent Grant Date: January 14, 2015

[20] M. R. Khawer, M. Abulius, Patent Title: “A Method And System For Improved Multi-Cell Support On A Single Modem Board”; Patent Case

#: 808249-GB-EPT; Great Britain Patent #: EP2647163; Patent Filing Date:, October 27, 2011; Patent Grant Date: January 14, 2015

[21] M. R. Khawer, M. Abulius, Patent Title: “A Method And System For Improved Multi-Cell Support On A Single Modem Board”; Patent Case #: 808249-DE-EPT; Germany Patent #: EP2647163; Patent Filing Date:, October 27, 2011; Patent Grant Date: January 14, 2015

[22] M. R. Khawer, L. So, Patent Title: “Lock-Less Buffer Management Scheme For Telecommunication Network Applications”; Japan Patent #: JP5671150; Patent Filing Date:, October 12, 2011; Patent Grant Date: December 26, 2014

[23] M. R. Khawer, S. Easwaran, Patent Title: “Apparatus For Multi-Cell Support In A Network”; Japan Patent #: JP5654678; Patent Filing Date:, June 29, 2011; Patent Grant Date: November 28, 2014

[24] M. R. Khawer, S. Easwaran, Patent Title: “Apparatus For Multi-Cell Support In A Network”; Korea Patent #: KR101467967; Patent Filing Date:, June 29, 2011; Patent Grant Date: November 26, 2014

[25] M. R. Khawer, S. Easwaran, Patent Title: “Apparatus For Multi-Cell Support In A Network”; Taiwan Patent #: TWI462018; Patent Filing Date:, July 26, 2011; Patent Grant Date: November 21, 2014

[26] M. R. Khawer, L. So, Patent Title: “Lock-Less And Zero Copy Messaging Scheme For Telecommunication Network Applications”; Taiwan Patent #: TWI451339; Patent Filing Date:, November 07, 2011; Patent Grant Date: September 01, 2014

[27] M. R. Khawer, V. Subramanian, R. Pupala;“ Visual Simulator For Wireless Systems”; U.S. Patent Filing #: 14/027687, Patent Filing Date: September 16, 2013

[28] M. R. Khawer, R. Soni, “System And Method For LTE And Wi-Fi Co-Existence On Same Small Cell In Unlicensed Band”, Patent Filing # 14/480950, Patent Filing Date: September 09, 2014

[29] M. R. Khawer, J. Ling, T. Hu, Patent Title: “User Equipment Assistance For Interference Mitigation In Unlicensed Frequency Bands”, Patent Filing # 14/476052, Patent Filing Date: September 03, 2014

[30] J. Ling, M. R. Khawer, R. Soni, Patent Title: “Occupancy Notifications For Unlicensed Frequency Bands”, Patent Filing # 14/496440, Patent Filing Date: September 25, 2014

[31] M. Baker, S. Ye, T. Hu, S. Wong, F. Han, M. R. Khawer, Patent Title: “Control Signaling For Partial Subframe Data Transmission Termination In LTE-U”, Patent Filing # PCT/CN2014/090540, Patent Filing Date : November 07, 2014

[32] M. R. Khawer, T. Hu, D. Wang, Patent Title: “Selective Activation And Deactivation Of Carriers In Unlicensed Frequency Bands”, Patent Filing # 14/561029, Patent Filing Date: December 04, 2014

[33] M. R. Khawer, Y. Cai, R. Soni, Patent Title: “LTE Small Cell Handover To Carrier-Grade Wi-Fi”, Patent Filing # 14/576405, Patent Filing Date : December 19, 2014

[34] M. R. Khawer, C. Kahn, R. Soni, Patent Title: “Allocation Of Unlicensed Frequency Bands For A Wireless Hotspot”, Patent Filing # 14/607772, Patent Filing Date: January 28, 2015

[35] M. R. Khawer, Y. Cai, R. Soni, Patent Title: “Differential Charging For Data Usage In Licensed And Unlicensed Frequency Bands”, Patent Filing # 14/607790, Patent Filing Date: January 28, 2015

[36] M. R. Khawer, T. Hu, M. Baker, S. Ye, “Network Controlled Acquisition Of Uplink Channels In Unlicensed Frequency Bands”, Patent Filing # 14/613654, Patent Filing Date: February 04, 2015

[37] M. R. Khawer, Y. Cai, R. Soni, Patent Title: “User Equipment Assisted Handover In Licensed And Unlicensed Frequency Bands”, Patent Filing # 14/607803, Patent Filing Date: January 28, 2015

[38] M. R. Khawer, Y. Cai, R. Soni, Patent Title: “Network Support For Differential Charging For Data Usage In Licensed And Unlicensed Frequency Bands”, Patent Filing # 14/631310, Patent Filing Date : February 25, 2015

[39] M. R. Khawer, Y. Cai, R. Soni, Patent Title: “A Charging Interface For Inter-Radio Access Technology Handover Of Call Sessions”, Patent Filing # 14/686066, Patent Filing Date: April 14, 2015

[40] M. R. Khawer, Patent Title: “Adaptive Subframe Puncturing For Carrier Sensing Adaptive Transmission”, Patent Filing # 14/686080, Patent Filing Date: April 14, 2015

PUBLICATIONS & STANDARDS CONTRIBUTIONS

[1] M. R. Khawer, J. Tang, F. Han, “usICIC – A Proactive Small Cell Interference Mitigation Strategy for Improving Spectral Efficiency of LTE Networks in the Unlicensed Spectrum”, accepted for publication in the upcoming edition of IEEE Transactions on Wireless Communications

[2] M. R. Khawer, “Design Strategies for the Application Server Architecture/Configuration (and Its Functions) in Next-Generation Communication Systems”; IEEE Communication Magazine – Design & Implementation Series (Volume: 53, Issue: 4, April 2015), Pages 227 – 233

[3] LTE-U Forum Technical Contribution Titled: “CSAT with Adaptive Sub-frame Puncturing”, April 23, 2015

[4] 3GPP TSG RAN WG1 LAA Meeting (Ad Hoc), Paris, France, 24th – 26th March 2015, TDOC submission #: R1-151075, “Channel Selection for LAA”, Alcatel-Lucent, Alcatel-Lucent Shanghai Bell.

[5] 3GPP TSG RAN WG1 LAA Meeting (Ad Hoc), Paris, France, 24th – 26th March 2015, TDOC submission #: R1-151072, “Measurement and RS Design for LAA”, Alcatel-Lucent, Alcatel-Lucent Shanghai Bell.

[6] 3GPP TSG RAN WG1 Meeting #79, San Francisco, 17th – 21st November 2014, TDOC submission #: R1-144702, “Channel Selection for Licensed-Assisted Access”, Alcatel-Lucent, Alcatel-Lucent Shanghai Bell.

[7] LTE-U Forum Technical Contribution Titled: “Multi Operator LTE-U Co-Channel Co-Existence Scenario: Need for Explicit Inter Cell Interference Co-Ordination Mechanism”, December 02, 2014, Dallas Tx

[8] 3GPP TSG RAN WG1 Meeting #78bis, Ljubljana, Slovenia, 6th – 10th October 2014, TDOC submission #: R1-144082, “Channel Selection for Licensed-Assisted Access”, Alcatel-Lucent, Alcatel-Lucent Shanghai Bell.

[9] LTE-U Forum Technical Contribution Titled: “LTE-U Smart Pricing”, August 6, 2014, Murray Hill New Jersey