Design and Evaluation of an Energy Efficient Frequency Adaptive Router

Feng Guo

This Dissertation is submitted to Dublin City University
in Candidacy for the Degree of Doctor of Philosophy

by the School of Electronic Engineering
Supervisor: Dr Xiaojun Wang

June 2016

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____

Candidate ID No.: _____

Date: _____

Abstract

Energy efficiency is becoming a prominent issue in ICT networks. Many approaches have been proposed to reduce the power consumption of ICT network devices. Among those green approaches, dynamic frequency scaling (DFS) offers an elegant solution for improving the energy efficiency of processors. To evaluate the impact of different DFS techniques on energy efficiency of real network devices, this work designs a prototype of a novel energy-aware Frequency Adaptive Router (FAR) that dynamically scales the operating frequency of core logic FPGA processor among five different processing capacities in response to traffic load, rather than leaving the network devices running on its maximum processing capacity all the time.

Three dynamic frequency adaptation control policies are introduced into the FAR to balance the trade-off between performance and power consumption. Based on statistics monitoring and preset thresholds, the proposed dynamic frequency adaptation control policies can manage the FAR to always operate at the lowest processing capacity required to handle instantaneous traffic load without affecting the quality of service (QoS). The implementation of these frequency adaptation control policies involves assessing an associated traffic throughput threshold beyond which the router will begin to lose packets for each of the five operating frequencies, and then adaptively scaling the operating frequency in response to the instantaneous traffic load to save energy without compromising end-to-end QoS.

The energy efficiency and performance of the FAR is evaluated at the five different operating frequencies with different number of active ports, traffic bit rates and packet sizes. The evaluation results show that when in idle state, the

FAR can significantly save power of up to 52%. Experiments with synthetic traces indicate that 46% of power can be saved while maintaining required QoS. Similar results can be expected when these general power-saving principles are applied in future commercial routers.

Acknowledgements

This work is co-funded by the China Scholarship Council (No. 2010647020) and the European FP7 ECONET project (No. 258454). I would especially like to acknowledge my supervisor Dr Xiaojun Wang who have spent much time reading drafts of this thesis and offering insightful suggestions, which led me to improve this work. I sincerely acknowledge Dr Martin Collier and Dr Olga Ormond for their guidance, encouragement and constructive criticism throughout the ECONET project.

I would like to thank my colleagues and friends Xiaowei Ma, Xiangjun Shi and Wenliang Fu from Beijing Institute of Technology (BIT), and Jie Jin, Xing Zheng and Lianghua Li from Hangzhou Dianzi University (HDU) for having detailed NetFPGA hardware Verilog code discussions with me. I would also like to thank research interns Yashank Sakhardande from National Institute of Technology Karnataka (NITK), Tejas Patel from Vellore Institute of Technology (VIT), Kevin Pierrot, Clément Aillet and Simon Fétigny from Ecole Nationale Supérieure Electronique Electrotechnique Informatique Hydraulique Télécommunications (ENSEEIHT) for their assistance on experimental setup and power consumption measurements.

Finally, I would like to honor the memory of my grandfather Zhiqi Zhang who passed away last year.

Contents

List of Tables

List of Figures

List of Abbreviations

AFIFO **A**synchronous **F**irst **I**n **F**irst **O**ut

ALR **A**daptive **L**ink **R**ate

ARP **A**ddress **R**esolution **P**rotocol

BUFG **G**lobal clock **BUF**fer

BUFGMUX **G**lobal clock **BUF**fer **MU**ltiple**X**er

CMOS **C**omplementary **M**etal **O**xide **S**emiconductor

CO_2 **C**arbon **d**i**O**xide

CPU **C**entral **P**rocessing **U**nit

DAQ **D**ata **A**c**Q**uisition

DCM **D**igital **C**Lock **M**anager

DFS **D**ynamic **F**requency **S**caling

DMA **D**irect **M**emory **A**ccess

DNS **D**omain **N**ame **S**ystem

DTP **D**ouble **T**hreshold **P**olicy

DVD **D**igital *V*ersatile **D**isk

DVFS **D**ynamic **V**oltage and **F**requency **S**caling

EC **E**uropean **C**ommission

EMAC **E**thernet **M**edia **A**ccess **C**ontroller

EPS **E**thernet **P**ort **S**hutdown

ETSI **E**uropean **T**elecommunications **S**tandards **I**nstitute

EU **E**uropean **U**nion

FAR **F**requency **A**daptive **R**outer

FCFS **F**irst **C**ome **F**irst **S**erved

FIFO **F**irst **I**n **F**irst **O**ut

FPGA **F**ield **P**rogrammable **G**ate **A**rray

FTP **F**ile **T**ransfer **P**rotocol

GAL **G**reen **A**bstract **L**ayer

GCLK **G**lobal **CL**oc**K** input pad

GND **G**rou**ND**

GTE **G**reen **T**raffic **E**ngineering

HTTP **H**yper**T**ext **T**ransfer **P**rotocol

IBUFG **G**lobal **I**nput clock **BUF**fer

ICMP **I**nternet **C**ontrol **M**essage **P**rotocol

ICT **I**nformation and **C**ommunications **T**echnology

IP **I**nternet **P**rotocol

LPI **L**ow **P**ower **I**dle

MAC **M**edia **A**ccess **C**ontrol

NGN **N**ext **G**eneration **N**etworks

NI **N**ational **I**nstruments

OSPF **O**pen **S**hortest **P**ath **F**irst

PC **P**ersonal **C**omputer

PCI **P**eripheral **C**omponent **I**nterconnect

PG **P**acket **G**enerator

PLAP **P**acket **L**oss **A**ware **P**olicy

PON **P**assive **O**ptical **N**etwork

QoS **Q**uality **o**f **S**ervice

RAM **R**andom **A**ccess **M**emory

RR **R**eference **R**outer

RTT **R**ound **T**rip **T**ime

SNMP **S**imple **N**etwork **M**anagement **P**rotocol

STP **S**ingle **T**hreshold **P**olicy

TCP **T**ransmission **C**ontrol **P**rotocol

TWh **T**ara **W**att **h**our

UDP **U**ser **D**atagram **P**rotocol

USB **U**niversal **S**erial **B**us

WDM **W**avelength-**D**ivision **M**ultiplexing

Chapter 1

Introduction

1.1 Research Motivation

Traditionally, network performance is measured using metrics including bandwidth, throughput, latency, jitter and error rate. These parameters are the primary concern for network designers and manufacturers, and optimized at every step of the design and manufacture process to enhance user experience. While energy consumption, energy cost, greenhouse gas emissions and other metrics were previously brushed aside.

Recently, reducing energy consumption is universally recognized as equally important as performance improvement [1]. This was motivated by economics to reduce power cost mainly for operators but also for users facing increasing energy price and network traffic, while still maintaining end-to-end quality of service (QoS). This was also motivated by environment protection to decrease resource wastage and greenhouse gas emissions by 20% by 2020 from European Union (EU) '20-20-20' targets launched by European Commission (EC) [2].

The 21st century is in an era of information explosion. As the sharp increase in Internet traffic, broadband accesses, complex services and end users, energy consumption in ICT industry is increasing exponentially at an alarming rate. A detailed report [3] estimated that a huge increase in overall network power consumption is predicted for European Internet Service Providers (ISPs) from 21.4 terawatt hours (TWh) in 2010 to 35.8 TWh in 2020 if no green technologies are adopted. Such increase in power consumption could result in millions of energy costs. Table 1.1 summarizes the annual energy consumption of major telecommunications companies worldwide.

Table 1.1: Annual energy consumption of major telecommunications companies worldwide

Company Name	Energy Consumption (TWh)				
	2011	2012	2013	2014	2015
British Telecom [4]	2.75	2.67	2.61	2.52	2.41 [1]
Telecom Italia [5] [6]	1.93	1.91	2.40	2.51	2.63
Deutsche Telecom [7] [8]	8.30	8.40	8.60	9.10	8.90 [2]
Telefonica [9]	-	-	-	6.35	6.43
France Telecom [10]	3.98	4.16	4.31	4.22	4.45
Verizon [11]	10.00	10.47	-	-	-
AT&T [12]	14.10	14.60	14.80	14.90	14.80
NTT [13]	5.66	5.22	4.88	4.83	4.91 [3]
China Mobile [14] [15]	12.93	14.30	15.06	17.18	20.09

In United Kingdom (UK), the energy consumption for British Telecom's network and estate in 2010 was 3.12 TWh in total [4], including 2,281 GWh energy consumption for maintaining networks, data centers and offices in UK, 417 GWh energy consumption by gas, heating oil and generator fuel in UK, and 425 GWh

[1] British Telecom has monitored and regulated energy consumption using its network of around 79,000 smart meters to optimise energy efficiency across 3,850 of its sites by responding to operational needs, weather forecast information and the number of people in the building. The annual energy consumption kept decreasing from 2011 to 2015.

[2] Due to the fact that Deutsche Telecom has made progress in energy efficiency including network migration to IP technology, improved network utilization, and consolidation of data centers, annual energy consumption slightly decreased by approximately 3% in 2015.

[3] With the adoption of energy saving measures, NTT reduced its annual energy consumption in 2012, 2013 and 2014.

energy consumption in countries outside UK. To operate networks, data centers and offices, British Telecom was one of UK's largest energy consumers, with around 0.7% of the entire UK's energy consumption [16]. As indicated in [17], approximately 10% of the UK's entire energy consumption was related to ICT equipments.

In Italy, the energy consumption of Telecom Italia was 2.64 TWh in 2015, which was approximately 0.89% of Italy's entire energy consumption [6]. Compared to 2.51 TWh in 2014 and 1.93 TWh in 2011, it increased by 4.56% and 26.89% respectively [5]. In this energy consumption increase from 2011 to 2015, the energy consumption of network infrastructures is the major source of the energy consumption increase. The energy consumption of network infrastructures contributed 70%. While, data centers contributed 10% and other sources (e.g., offices, shops, etc.) contributed the remaining 20%.

In Germany, the energy consumption of Deutsche Telecom also followed an increasing trend from 2011 to 2014, except for a slight decrease in 2015. The energy consumption of Deutsche Telecom was 8.90 TWh in 2015 [8]. Compared to 8.30 TWh in 2011 [7], it increased by 6.74%. The Deutsche Telecom attributed this energy consumption increase to increasing transmission volumes and network expansions. In Spain, the energy consumption of Telefonica was 6.43 TWh in 2015, which was about 2.70% of Spain's entire energy consumption [9]. Compared to 6.35 TWh in 2014, it increased by 1.33%. In France, the energy consumption of France Telecom also increased from 3.98 TWh in 2011 to 4.45 TWh in 2015 [10].

Similar trends are observed in the rest of the world. In United States (US), an early work [18] investigated and examined annual energy usage for office and network equipments. This work also estimated that the total energy used by office and network equipments was 74 TWh, which was 2% of entire US's energy

usage in 1999. In 2011, the energy consumption of Verizon (an American telecommunications company) was 10.00 TWh, which was about 0.24% of entire US's energy consumption [11]. The energy consumption of Verizon increased to 10.47 TWh in 2012. While, the energy consumption of AT&T (another American telecommunications company) was 14.60 TWh in 2012 [12], which was of the same magnitude as Verizon.

In Japan, the energy consumption for telecommunications was 42 TWh in 2004 [19], which was about 1% of the Japan's entire energy consumption and was about 4% of the Japan's electricity generation [20]. The energy consumption of NTT (a Japanese telecommunications company) was 4.91 TWh in 2015 [13], which was about 0.53% of Japan's entire energy consumption. In China, the energy consumption of China Mobile (a Chinese telecommunications company) has been consistent growth from 2011 to 2015. The energy consumption of China Mobile was 20.09 TWh in 2015 [15]. Compared to 12.93 TWh in 2011 [14], it increased by 35.63%.

Apart from total energy consumption increases, average energy price is also increasing in the meantime. Referred to the United States Energy Information Administration (EIA) agency, average energy price has increased with a high pace except for two slight decreases in 2012 and 2015. The average energy price increased from 99.0 million per TWh (in US dollar) in 2011 to 104.1 million per TWh in 2015. Table 1.2 summarizes average energy price from 2011 to 2015 [21]. The increasing of energy consumption and price are driving Internet Service Providers (ISPs) and telecommunications companies worldwide towards energy efficient networking.

Table 1.2: Average energy price (Million per TWh in US dollar)

Year	2011	2012	2013	2014	2015
Million per TWh in $	99.0	98.4	101.0	104.4	104.1

On the other hand, carbon dioxide (CO_2) emissions from the ICT industry should never be ignored [22]. In 2002, the Global e-Sustainability Initiative (GeSI) reported that global Telecommunications infrastructure and devices contributed 152 million tons CO_2 emissions, including 66 million tons from mobile networks, 64 million tons from fixed narrowband networks, 18 million tons from Telecommunications devices and 4 million tons from fixed broadband networks [3]. From 2002, Telecommunications CO_2 emissions have grown from 152 million tons in 2002 to 300 million tons in 2007. The GeSI also estimated that the CO_2 emissions of global Telecommunications infrastructure and devices will be approximately 349 million tons in 2020, including 179 million tons from mobile networks, 70 million tons from fixed narrowband networks, 51 million tons from Telecommunications devices and 49 million tons from fixed broadband networks [3].

In 2007, Gartner estimated that the ICT industry contributed about 2% of global CO_2 emissions [23]. This work also indicated that the approximately 2% of global CO_2 emissions that the ICT industry contributed is equivalent to the entire airline industry. Such an unsustainable trend of the power consumption and the CO_2 emissions in the ICT industry should be addressed seriously in order to reduce power consumption and CO_2 emissions from growing ICT industry.

1.2 Energy Efficient Networking

In the late 20th and early 21st centuries, Internet traffic is witnessing an exponential growth in the face of technological and social change [24]. To keep pace with the traffic growth, network devices are designed to translate improvements in digital circuits and operating frequency into higher

performance, such as maximizing the routing capacity of network devices at the cost of the highest and constant power consumption.

Network traffic is very dynamic and the busy-hour traffic rate has always been much higher than the average traffic rate. The gap between peak and average traffic is growing further due to increased share of video streaming of the Internet traffic. The routing capacity of network devices is designed to accommodate peak rates rather than average rates. In backbone networks, the routing capacity is usually twice larger than rush-hour traffic volume to achieve minimum network delay and packet loss [25]. However, the peak rates only occur for a very small fraction of the time, so significant amount of power are wasted to keep the network always at full capacity. The work in [26] monitors Internet backbone utilization levels and results indicate that the routing capacity in backbone networks has actually grown faster than Internet traffic in 2008. The average and peak link utilization on major backbone networks are approximately 29% and 43% respectively.

In ICT networks, traditional design of network devices tends to primarily emphasize network performance. Power consumption, while given some attention, failed to be recognized as an important part of a family of design metrics, along with performance, reliability, security, etc. As the performance has been always a predominant consideration, most network devices never vary their capacity when they are running, even if very little or no traffic data is involved. Such phenomenon without enough attention on power consumption has caused huge energy waste in ICT network infrastructures. Power consumption measurements in the work [27] show that a 1 Gb/s Ethernet link consumes approximately 4 W more power than a 100 Mbp/s Ethernet link, and a 10 Gb/s Ethernet link consumes even more power ranging from 10 W to 20 W. Power

consumption of most Ethernet links is independent of their link utilization [27]. A fully utilized Ethernet link consumes almost the same amount of power as an idle Ethernet link [28]. Power consumption measurements in the work [29] show that the average Ethernet link utilization is ranging from 1% to 5%.

Performance should never be the sole focus. As energy and power issue comes to the forefront recently, future network devices should integrate more features and deliver higher performance within the same or even less energy budget. Otherwise, without green techniques adopted into ICT network infrastructures, energy inefficiency in roughly three billion Ethernet connections worldwide [30] could potentially lead to massive energy waste and millions of unnecessary energy costs.

With the development of Next Generation Networks (NGN), it is widely believed that energy efficiency should be taken into account as a prominent design metric in ICT network infrastructures. To reduce the energy consumption of ICT and the environmental impact of ICT, many approaches have been proposed towards ICT energy efficient networking. All shared the same common aim to achieve the lowest possible power consumption while not compromising the performance.

Efforts in these approaches can be divided into two categories: device level and network level. In networks, device level techniques aim to improve the energy efficiency of individual network devices (e.g., switches, routers, etc.). This is mostly achieved by adopting energy-saving modes when the network device is underutilized or improving the energy efficiency of hardware components. On the other hand, network level techniques aim to improving the energy efficiency

of whole network systems. This is achieved by manipulating the energy profile of network components and consolidating network traffic.

1.3 Scope and Objectives

The energy consumption of ICT network device has been growing at a fast pace over the past two decades. The typical router capacity is increased from 100 Gb/s in 2000 to 10 Tb/s in 2008 with energy consumption increasing from 1.7 KW to 50 KW [31]. ICT network devices consume significant energy even in the idle state. Given that typical router utilization are ranging from 20% to 50% in Internet Service Providers (ISPs) backbone, from 8% to 25% in enterprise networks and less than 1% in Local Area Networks (LANs) [32], the energy wastage is significant. Observation in the work [3] reported that, in the life time of the ICT network device, operating the device contributed 80% of the lifetime energy consumption, while manufacturing the device contributed the remaining 20%. Therefore, reducing the operating energy consumption of the ICT network device is worth the effort. Furthermore, it is also reported in the work [3] that processing bits in network devices and data centers contributed 63% of energy consumption in ICT, while transporting bits in telecommunications contributed the remaining 37%. Therefore, reducing processing per bit energy consumption in ICT network device is even more crucial.

To reduce processing per bit energy consumption, the scope of this work is to design and develop an energy efficient network hardware router named the Frequency Adaptive Router (FAR), achieving energy proportionality between the router's energy consumption and the router's utilization. This work is validated on the NetFPGA 1G board [33] because it supports a community of open source hardware and software, and makes use of modular structure in the reference

pipeline. Therefore, it is capable of providing a fast way to develop and experiment custom green mechanisms with low cost reconfigurable feature. This work develops and implements a fine-grained energy proportional technique on the NetFPGA, and examines the impact of various factors, including the operating frequency, the number of active Ethernet ports, the traffic bit rate and the packet size, on both the power consumption and the performance. The main objectives of this work are to address the following three challenges.

Many approaches have been proposed to improve the energy efficiency of ICT networks at various components levels. However, most of these approaches have contributed to the energy-saving solutions in simulated environments [34] [35]. Due to the fact that different energy states adaptation is not supported by most of current hardware coupled with compatibility issues, practical application of the proposed energy saving approaches in existing networks is still challenging.

Energy aware network device may suffer from long response time and more energy consumption during transitions of different energy states. Transitions of different energy states may lead to performance degradation (e.g., increasing packet loss, increasing network delay, etc.). Therefore, minimizing the overall energy consumption of network device and minimizing the overhead of transitions of different energy states are crucial in a parallel manner. Building hardware components to efficiently deal with the overhead of energy states transitions is another challenge.

The last challenge is achieving proportionality between energy consumption and device utilization. Achieving energy proportionality not only can reduce the energy consumption of network devices, but also can eliminate the demand for other complicated energy saving approaches [36].

1.4 Contribution

The main contribution of this research is the design of a Frequency Adaptive Router, the exploration of frequency adaptation control policies, and the energy efficiency and performance evaluation of the Frequency Adaptive Router. This work mainly contributes in four aspects:

Energy savings benchmarking. Currently, switches and routers do not include comprehensive power consumption values. Most device specification sheets only report the maximum rated power [37] [38]. This information is insufficient to well understand the actual power consumption of a network device. As shown in the rest of this thesis, the actual power consumed by wired routers depends on various factors, such as: 1) operating frequency, 2) number of active Ethernet ports, 3) traffic bit rate and 4) packet size. Therefore, counting only the maximum rated power of the network device could grossly overestimate the actual energy consumption of the network device. This work introduces an accurate and fine-grained power consumption measurement method to measure the power consumption of peripheral component interconnect (PCI) based NetFPGA 1G router, which can be used to better quantify energy savings from the energy proportional technique on real network devices. Compared to the maximum rated power reported in the work [37] [38], this work measures real time power consumption of the NetFPGA router under different operating frequencies, number of active Ethernet ports, traffic bit rates and packet sizes.

Frequency Adaptive Router. Starting from the NetFPGA 1G Reference Router (RR), this work builds a dynamic frequency adaptive router (FAR) to develop and implement power scaling techniques on real network devices. When toggling the operating frequency of the NetFPGA core logic processor between 125MHz and

62.5MHz on the original RR, the frequency switching causes a board reset and all buffered packets are lost. Three additional operating frequency options, 31.3MHz, 15.6MHz and 7.8MHz, are added in the FAR for more finely tuned frequency switching without significant packet processing delay. The board reset problem is also eliminated. The FAR dynamically scales and tunes the operating frequency of the router in response to the traffic load it is currently handling, so that when the traffic load is low, the router can adaptively switch to an appropriate low routing capacity state to save power consumption. In ICT networks, peak traffic rate is much higher than average traffic rate. The Frequency Adaptive Router can operate at lower frequencies at off peak times to save power consumption. As shown in the rest of this thesis, compared to the RR in the work [33], the FAR consumes less quiescent power consumption by up to 52% and less total power consumption by up to 46%.

Frequency Adaptation Control Policy. To achieve energy efficiency and avoid compromising performance, a frequency adaptation control policy aims to offer the required performance with the lowest possible power consumption. Based on statistics monitoring and preset thresholds, three different dynamic frequency adaptation control policies are designed to determine when to initiate a frequency transition. The three dynamic frequency adaptation control policies are: Single Threshold Policy (STP), Double Threshold Policy (DTP) and Packet Loss Aware Policy (PLAP). Previous works [34] [35] have contributed to the energy-saving solutions in simulated environments. Compared to the simulation of the control policy in the work [34], this work implements the proposed frequency adaptation control policies directly in the FAR, providing more accurate and valid power consumption and performance evaluation results.

Hardware Acceleration. The dynamic frequency control policies of the FAR are directly implemented in hardware because software is slow compared to dedicated hardware. The dedicated hardware makes an enormous difference in the speed of time-sensitive operations, providing a significant statistics monitoring advantage over software. For example, frequency transition time on software implementation of a control policy consists of the delay in software reading buffer usage, operating frequency selection according to the control policy and setting the appropriate frequency control register, which involves communications between hardware and software through reading and writing registers. While, frequency transition time on hardware implementation of a control policy eliminates the delay in communications between hardware and software. Compared to the software implementation in the work [34] [39], this work reduces the frequency transition time by up to 85%.

1.5 Thesis Structure

The rest of this thesis is organized as follows. Chapter 2 reviews the existing techniques related to energy efficient networking. Chapter 3 describes the design architecture of the NetFPGA Reference Router and the Frequency Adaptive Router. Chapter 4 explains the frequency adaptation control policies for the Frequency Adaptive Router, and the system models for the Reference Router and the Frequency Adaptive Router. Chapter 5 addresses the implementation details, describing the lab setup, the tools for power consumption measurements and performance measurements. The energy efficiency and the performance of the Frequency Adaptive Router are evaluated and analyzed in comparison with the Reference Router. Chapter 6 concludes the thesis.

Chapter 2

Related Work

Internet traffic has been witnessing a sky-rocketing growth over the past decade. To keep pace with such growth, it is even more compelling to reduce power consumption while managing massive increases in Internet traffic and networking hardware [40]. Economic reasons and environmental concerns on sustainable growth have created a demand for green communication networks. Many approaches have been proposed to manage energy and power issues to improve the energy efficiency of the ICT infrastructures in a variety of areas (e.g. wired networks, wireless networks, optical networks, smart grids, etc) [41] [42].

To improve the energy efficiency of network devices, Guapta et al. [43] proposed the sleep mode in green networking to place network interfaces and components of network devices to sleep when they are idle. This work also discussed main implementation challenges and expected benefits from the sleep mode. Since this work [43], many green approaches have been proposed and several literature surveys in various areas and domains of energy efficiency have been published [44] [45].

These green approaches can be mainly divided into green approaches in device level and green approaches in network level. The green approaches in device level make efforts to design low power consumption network equipment and improve the energy efficiency of the hardware using sleep mode functionality or power scaling capabilities. Device level techniques reduce power consumption of individual network devices (e.g., switches, routers, etc.) by adapting the power saving mode of various components of the devices (e.g., processors, memories, network interfaces, etc.). This is achieved in a localized manner by locally collected information.

On the other hand, the green approaches in network level focus on routing policies to reduce the network energy consumption, such as maximizing the number of network interfaces and components that can be put into sleep while maintaining the required quality of service. Network level techniques minimize power consumption of a network using global information, including the topology, the status of links, the traffic demands, the performance requirements and the power saving modes of all the devices within the network. This is achieved either in a centralized manner by collecting all the information at a single governor or in a distributed manner through network nodes cooperation.

2.1 Device Level

Many approaches have been proposed to improve the energy efficiency of the network by re-engineering conventional network equipments and network protocols. Based on locally collected information, device level techniques for reducing the energy consumption can be divided into two categories: sleep mode and power scaling [46].

Sleep mode refers to a low power mode for electronic devices [47]. This mode can significantly reduce power consumption compared to leaving a device fully on all the time when the network presence of this device is not required [48]. In sleep mode, power is cut for unneeded subsystems and the RAMs are placed into a minimum power state, just sufficient to retain their data. Sleep mode is a promising technique in ICT green computing to save energy. In ICT green networking, only when there is no traffic to handle for an extended period of time can a router go to sleep.

Another energy saving technique is power scaling. It provides the ICT network devices the capability to dynamically scale into different processing states in response to instantaneous operational needs and traffic load. In power scaling category, Dynamic Voltage and Frequency Scaling (DVFS) has always been the king solution [49]. This fine-grained energy proportional technique allows the voltage supply and/or the clock speed of a processor to meet the instantaneous operational need of the operation being performed, while minimizing power draw and heat dissipation.

Once the power scaling capabilities are provided in a router, appropriate power scaling control policy must be introduced to balance the trade-off between performance and power consumption. An effective power scaling control policy should manage the router to always operate at the lowest appropriate routing capacity to handle the instantaneous traffic without affecting the performance as perceived by the user. The control policy must be effective and simple to be directly implemented inside a network router. Significant increases to packet loss and delay should be avoided when implementing a power scaling control policy, as this could degrade user experience.

2.1.1 Sleep Mode

Energy Star standard [50] [51] was launched in 1992 and was designed to improve energy efficiency in computers, servers, appliance, heating and cooling systems, etc. This promoted the widespread adoption of sleep mode in electronic devices [52]. Sleep mode refers to a low power mode for the electronic devices and this mode can save significant power consumption compared to leaving a device fully on [53]. When placed in sleep mode, power was cut for unneeded subsystems entering into a very low power state.

In green computing, power management first appeared in laptop computers to extend battery life for mobile users [54]. Figure 2.1 demonstrates the power management control flow. The firmware periodically sends request signals to the operating system to start power management. If the power management is enabled on the operating system and no activity is detected from the application, the operating system sends a responding signal back to the firmware to start a power management inactivity timer. After a specified time with no activity detected, the firmware initiates power management by sending signals to the hardware (e.g., the processor, the video card, the hard disk, etc). The corresponding hardware is then placed into a low power operating mode. An activity interrupt or a wake-up interrupt could lead the firmware to send signals to corresponding hardware to get back to an active and powered mode of operation. There are two modes of operation for the power management of the hardware, a slow clock mode and a stopped clock mode. The slow clock mode reduces speed of operation for the hardware with reduced power consumption. While, the stopped clock mode turns the hardware almost completely off and only an interrupt can cause a restart of the hardware.

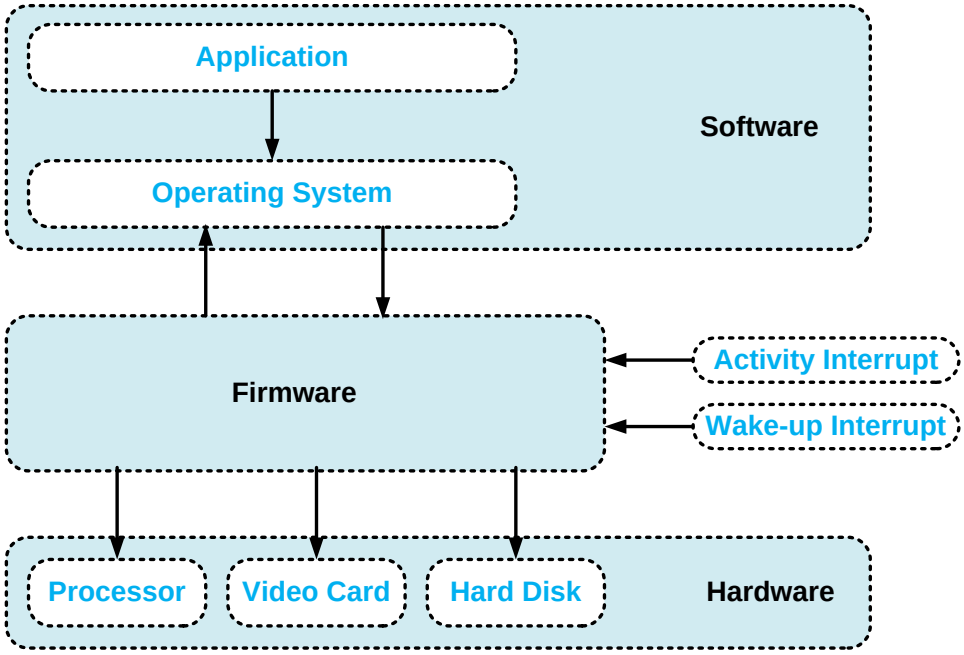

Figure 2.1: Power management control flow [54]

In green networking, sleep mode is based on similar power management primitives in green computing, which allows network devices or part of them turning themselves almost completely off, while all their functionalities are frozen. Compared to idle state, sleep mode refers to a deeper idle state characterized by more power savings but longer wake-up time. There are challenges to applying sleep mode in network devices because it takes more time and power to transition between the *on* and *off* state. Predicting the *off* period and adapting to the appropriate state is still difficult. For instance, when a network device or a part of it is placed into sleep mode, its applications and services stop working and its network connectivity is lost. As a result, the network device loses its network presence so that it can not maintain network connectivity and respond to its applications or services. In addition, when the network device wakes up, it takes longer time to re-initialize its applications and services by sending its signalling traffic.

To introduce sleep mode into green networking with less overheads, Christensen et al. [54] proposed to use a sleep proxy for a network host and to integrate power management functions into TCP/IP and network applications. TCP/IP connections follow a client-server (CS) architecture. For peer-to-peer (P2P) connections, one peer acts as the client, the other peer acts as the server. A server can listen to requests from multiple connected clients. When a client request is detected at the server, a three-way handshake is initiated to establish a connection. When the connection is established, data can be transferred from a client to a server or a server to a client. The sleep proxy is designed to handle TCP/IP connections on behalf of the sleeping hosts allowing a safe sleep of the network hosts maintaining network presence through the proxy while at sleep.

In their follow-on work, Christensen et al. [28] investigated network traffic on several heavy-loaded computers in a university campus and results show that even the heavy-loaded computers still have many hours of idle time per day during which they can be placed into sleep mode. This work developed a proxying Ethernet adapter which can handle routine protocol messages without waking-up a desktop computer. The proxying Ethernet adapter can also wake-up the sleeping computer if needed.

An energy-efficient network device should be energy proportional to its actual needs. An idle or lightly utilized PC, Ethernet link or switch should not consume the same power as heavily utilized ones. Gunaratne et al. [27] developed several methods to reduce energy consumption of PCs, Ethernet links, and switches. These methods are achieved by supporting centralized proxying and control for discovery protocols, and disabling unused network devices in the routing path.

Most Ethernet connected network devices are fully powered on all the time to maintaining their network connectivity. Nordman et al. [55] developed a power management proxy to enable PC power management in network host PCs. The power management proxy is a low-powered entity that maintains full network presence for a sleeping high-powered network device. Their proposed proxy is able to bring powered down network host PCs back into a fully powered on state with reliable wake-up operations.

Gupta et al. [56] proposed dynamic Ethernet link shutdown to reduce power consumption on Ethernet links. This work uses buffer occupancy, the behavior of previous packet arrival times and a configurable maximum bounded delay to make decisions for dynamic Ethernet link shutdown. Simulation results using a synthetic traffic generator show that power savings can be achieved with little noticeable impact on network delay and packet loss.

2.1.2 Power Scaling

In addition to sleep mode, power scaling [57] [58] can also be used to improve the energy efficiency of network hardware. Power scaling capabilities allow dynamically reducing the working speed of processing engines or link interfaces. The power scaling is usually accomplished by adopting two basic techniques: Low Power Idle (LPI) and adaptive performance scaling. The former forces links or processing engines to enter low-power states when not sending or processing packets and quickly switch to a high-power state when sending one or more packets. The latter allows dynamically modulating the capacity of a link or a processing engine in order to meet traffic load and service requirements. These techniques are not exclusive and can be jointly adopted in order to adapt system performance to current workload requirements. Scaling the working speed of

processing engines or link interfaces as a function of user demand can save less power compared to sleep mode, but it has less wake-up time and less performance impact to a user.

Nedevschi [59] firstly proposed to support such energy-aware capabilities with a special reference to the Low Power Idle (LPI) with I/O traffic handling mechanisms, able to shape traffic profiles in order to optimally exploit LPI and adaptive performance scaling. For example, an I/O traffic handling mechanism based on a simple polling policy well suits an optimal use of LPI. An optimization policy is generally needed to configure and control the usage of energy-aware capabilities and states with respect to the estimated workload and service requirements. Regarding the optimization policy, several methods have been proposed in order to estimate the current workload and to optimally control the trade-off between performance and energy consumption in the computing system field [60] [61]. These methods range among predictive techniques [62] and dynamic schemes [63] [64], which were studied for disk drives [65], processors [66] [67], and other components [68].

Christensen et al. [27] has specifically addressed how to reduce the direct energy use of Ethernet links, and has contributed to the development of the IEEE 802.3az energy-efficient Ethernet (EEE) standard. The work [27] first explored the adaptive link rate (ALR) for Ethernet so that an Ethernet link can be operated at a lower data rate during periods of low utilization and at high data rate only for high utilization periods. Most Ethernet links are highly over-provisioned, with ALR most Ethernet links could operate at a lower bit rate and thus reduce energy consumption compared to operation at a higher bit rate all the time [69].

The auto-negotiation scheme in the 802.3 standard [70] takes about 100 ms to change data rates at a 1 Gb/s link data rate. Gunaratne et al. [69] proposed the ALR that allows the speed of network links to be changed by adaptively switching to different processing states in response to the amount of data that is being transmitted. The transition time of changing data rates in the ALR can be significantly reduced to 1 ms through a newly-defined handshake mechanism. Later on, by combining the schemes of LPI and ALR, the IEEE 802.3az EEE standard [30] is proposed. The LPI specified in the IEEE 802.3az EEE standard currently allows a 10 Gb/s link to wake up in less than 3 us.

The IEEE 802.3az EEE standard [30] can be also used in smart grid. As described in the IEEE 802.3az EEE standard, the EEE uses Low Power Idle to reduce the power consumption of a link when the link is idle. During the low-power mode, refresh signals are sent periodically to maintain alignment between the transmitter and the receiver [71]. For a low link utilization of 25%, results in the work [42] indicated that the EEE can reduce power consumption by 25% .

Bolla et al. [72] analyzed and empirically modeled the energy modulation capabilities of processing engines in Linux-based software routers equipped with general-purpose and multi-core processors that already include LPI and ALR primitives. The results obtained by evaluating several hardware architectures indicated that both technologies permit the trade-off between power consumption and network performance to scale almost linearly.

Bolla et al. [73] extended their approach by introducing a control framework for optimally tuning LPI mechanisms and adaptive performance scaling to statistically meet current traffic loads and service requirements. The results obtained on real traffic traces show that up to 60% power savings can be achieved. Previous work

[59] especially focused on LPI primitives and performed a comprehensive study of the impact of transition times on LPI as a function of load. The results indicated that as the transition times shrink from the value of 10 ms to 1 ms and then further to 100 us, the time spent sleeping at 30 percent load goes from 0 at transition time of 10 ms to 40 percent when transition time is 1 ms, and to 70 percent when the transition happens in 100 us.

In the power scaling category, dynamic voltage scaling and dynamic frequency scaling are often used together as DVFS to conserve power [74] [75]. Dynamic voltage scaling [76] [77] is a power management technique in green ICT, where the voltage used in a unit is increased or decreased depending upon circumstances [78]. Decreasing voltage is to conserve power, particularly in laptops and other mobile devices, where energy comes from a limited battery. On the contrary, increasing voltage is to increase performance, or in rare cases, to increase reliability.

Dynamic frequency scaling [79] [80] is another power conservation technique that works on the same principles as dynamic voltage scaling. Dynamic frequency scaling technologies allow the clock speed of the processor to be dynamically changed to different processing states depending upon the traffic load. This allows the processor to meet the instantaneous performance needs of the operation being performed, while minimizing power draw and heat dissipation.

Meng et al. [34] proposed a multi-frequency scaling (MFS) scheme that examines buffer usage inside a network device so that the components of the device could dynamically scale its capacity according to its buffer occupancy. In

the MFS scheme, a clock adapter is the hardware module for frequency scaling, and a multi-dual-threshold policy is adopted as the MFS control policy.

Pham et al. [39] [81] proposed a power scaling mechanism on the NetFPGA OpenFlow switch that can adapt the operating clock frequencies of the FPGA processor and the link rates of the four Ethernet ports based on the actual traffic load. The power can be saved by changing the clock frequency from 125 MHz to 3.9 MHz in combination with four possible operation modes on each Ethernet port (idle, 10 Mb/s, 100 Mb/s and 1 Gb/s).

2.2 Network Level

In previous section, device level techniques are reviewed. However, device level techniques can not guarantee the minimization of Internet power consumption. The device level techniques are used based on locally collected information without any node coordination. Currently, real networks are over-provisioned to accommodate the maximum expected traffic demands and over-redundant to deal with link and node failures. As a result, many network links are under-utilized and many network devices are constantly in operation with maximum capacity. Therefore, consolidating network traffic to place specific network nodes and links to a power saving mode is promising to improve the energy efficiency of networks. Network level techniques requires cooperations between network nodes to collect information on global network state, including the topology, the status of links, the traffic demands, the performance requirements and the power saving modes of all the devices within the networks. The network level techniques can be implemented both at the design stage (network design) and the operating stage (network routing). In this section,

network level techniques in these two different categories network design and network routing are reviewed.

2.2.1 Network Design

Network design traditionally focuses on the minimization of network capital expenditure, including the device and installation costs of the network infrastructure [82]. However, as Internet traffic and energy costs are exponentially increasing, energy consumption is becoming a major issue for network operators for three main reasons, including significant increase in operational expenditure due to increasing traffic levels and energy costs, significant increase in CO_2 emissions and escalated heat dissipation problems [83]. As a result, improving the energy efficiency in network design has been paid much greater attention than before.

In digital communication networks, a core network is the backbone infrastructure of a network that usually interconnects metropolitan areas and may extend across different continents or nations. On the contrary, an access network is the last mile infrastructure of a network that connects the end users to their immediate core network. Internet can be divided into a core network and several access networks connected to the core network. Network devices in different network categories have different power consumption and performance. Understanding power consumption of different networking devices is the first step to identify where the most power savings can be achieved in ICT industry.

The majority of the energy used by the Internet today is consumed in the access networks, and this will continue to increase [84]. As access networks expand to deliver increasing amount of data traffic to increasing number of customers, efforts

have been made to improve the energy efficiency of access networks. The access networks can be classified according to the type of the transmission medium (e.g., copper cables, optical fiber or wireless). Optical networks use optical signals over optical fibers for data transmission. Since the optical transmission provides high bandwidth, low signal attenuation, immunity to electromagnetic interference, low energy consumption, etc, it has been widely spread over decades.

In optical access networks, central offices (COs) and remote nodes (RNs) provide connectivity between the end users and the core networks. The Point-to-Point (P2P) architecture can be used in optical access networks [85]. Each end user is connected to a CO through a dedicated optical fiber. However, the P2P requires massive optical fibers and corresponding transceivers deployed at the COs and the end users. Figure 2.2 demonstrates a simple example of P2P architecture in optical access networks. The P2P in Figure 2.2 requires three optical fibers and six transceivers including three transceivers at the CO and three transceivers at the end users.

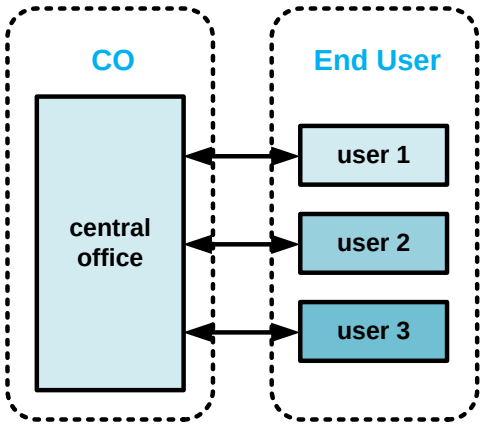

Figure 2.2: Point-to-Point architecture in optical access networks

The optical fiber is precious. To reduce the number of optical fibers, the Point-to-Multipoint (P2M) architecture, including the active optical network (AON) and

the passive optical network (PON), is proposed [86]. In the AON architecture, a RN is inserted between the CO and the end users [85]. The signals from the CO to the end users are distributed through the RN using an electronic powered router or switch. The AON requires only one optical fiber between the CO and the RN, while each end user is connected to the RN with a dedicated metallic cable. A power supply must be provided at the RN to feed the router or switch, especially for the optical-to-electrical-to-optical (O/E/O) conversions. The AON significantly reduces required amount of optical fibers, but introduces considerable amount of transceivers. Figure 2.3 demonstrates a simple example of active optical network architecture in optical access networks. The AON in Figure 2.3 requires only one optical fiber between the CO and the RN, and eight transceivers including one transceiver at the CO, four transceivers at the RN, three transceivers at the end users.

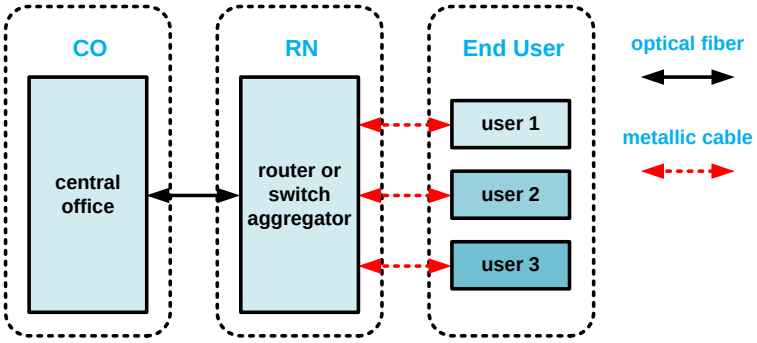

Figure 2.3: Active optical network architecture in optical access networks

Passive optical network (PON) is proposed to improve the energy efficiency of the AON by replacing the powered router or switch at the RN with an unpowered passive optical splitter [85] [87]. In the PON architecture, optical signals are directly processed in the optical domain [88]. The passive optical splitter at the RN is able to distribute the optical signals from the CO to the end users without electronic processing. Thus, the PON eliminates the power supply

for the router or switch at the RN and the power consumption for the optical-to-electrical-to-optical (O/E/O) conversions [89]. Figure 2.4 demonstrates a simple example of passive optical network architecture in optical access networks. The PON in Figure 2.4 requires only one optical fiber between the CO and the RN, and four transceivers including one transceiver at the CO and three transceivers at the end users.

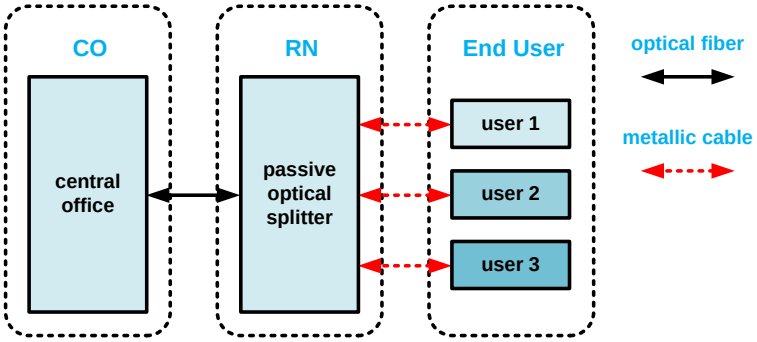

Figure 2.4: Passive optical network architecture in optical access networks

In optical core networks, the wavelength-division multiplexing (WDM) technology multiplexes different optical carrier signals over a single optical fiber using different wavelengths of optical signals [85]. Each wavelength carries an individual signal that does not interfere with the other wavelengths.

The WDM provides high system capacity and low energy consumption per bit [90]. The WDM creates virtual optical fibers, which multiply the system capacity. For example, the WDM systems can handle up to 160 signals and thus expand a basic 100 Gb/s system to a multiplied 16 Tb/s system over a single optical fiber. The WDM also minimizes the number of optical-to-electrical-to-optical (O/E/O) conversions per each provisioned connection and thus the overall power consumption is reduced. Figure 2.5 compares the difference between the

Peer-to-Peer (P2P) and the WDM. The P2P uses three optical fibers, while the WDM uses only one optical fiber with the same system capacity.

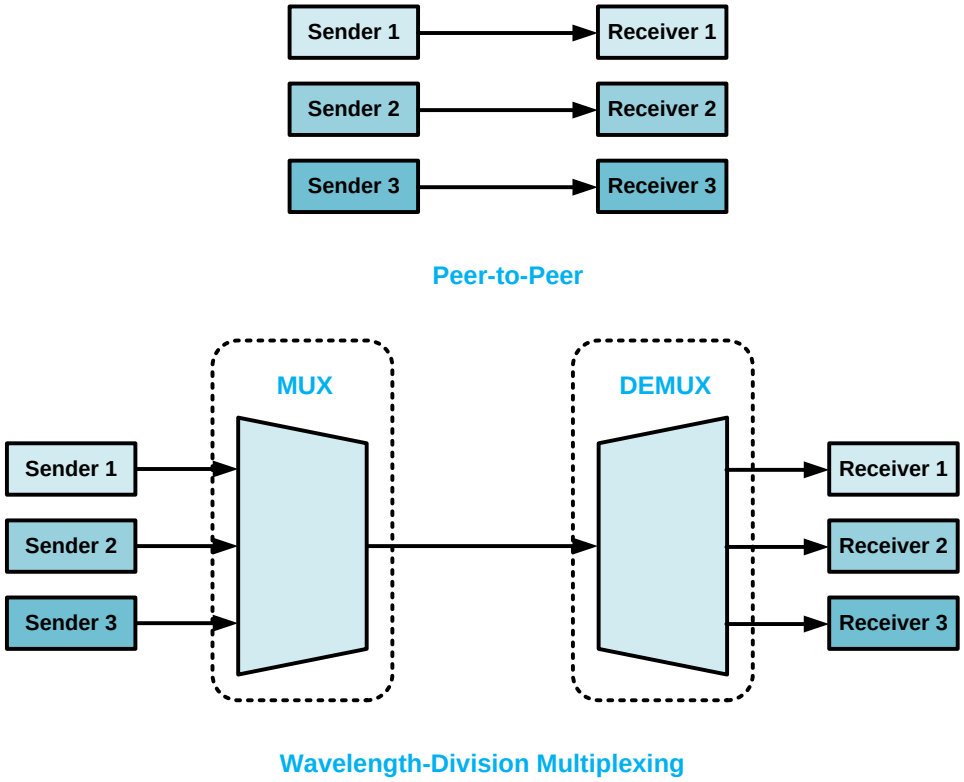

Figure 2.5: Difference between P2P and WDM

Based on the WDM-PON architecture, the energy efficiency of the optical networks can be further improved by putting unused optical devices into sleep mode in optical access and core networks. In optical access networks, the optical devices at the COs and the end users can be placed into sleep mode when there is no traffic to be processed. In addition to sleep mode, the power consumption in optical core networks can be further reduced by routing policies, such as maximizing low powered optical devices or minimizing used optical devices to place unused ones to sleep. The European Commission (EC) funded the Towards Real Energy-efficient Network Design (TREND) project to investigate the energy efficiency of the electrical and optical layers of Internet Protocol over

wavelength-division multiplexing (IP-over-WDM) networks [91]. The project also proposed energy-aware adaptive routing solutions (EA-ARSs) for the network operators.

Implementing the sleep mode in the optical networks requires careful design and operations to make sure that other metrics, including network delay [92], packet loss rate [93], blocking ratio, system reliability, device lifetime [94] [95], etc, are maintained at required level [96] [97]. Otherwise, the disadvantages of performance degradation may far outweigh the advantages of power savings from the sleep mode. Wiatr et al. [98] investigated the relationship between power savings and network performance in WDM networks.

Based on the wavelength-division multiplexing passive optical network (WDM-PON) architecture [85] [87], the energy efficiency of the optical networks can be improved by putting unused optical devices into sleep mode in optical access and core networks [88]. In optical access networks, the optical devices can be placed into sleep mode when there is no traffic to be processed [99] [100]. In addition to sleep mode, the power consumption in optical core networks can be further reduced by routing policies, such as maximizing the use of low powered optical devices or minimizing used optical devices and put unused ones to sleep [101].

2.2.2 Green Routing Algorithms

The green approaches in network level, such as the green routing algorithms, are also important to reduce the power consumption of network devices, since it manages switching network links, entire network devices, or parts of them to a sleep mode in a smart and effective way. Emerging green approaches to network control, routing, and traffic engineering [47] allows dynamically setting network

components to sleep during low utilization periods, to minimize the power consumption of the overall network while meeting the current traffic load and operational constraints.

Traffic engineering provides network traffic flows control to optimize resource utilization and network performance to meet specific performance requirements. Traditionally, traffic engineering focuses on load balancing among network links to avoid congestion from traffic bursts. On the contrary, the green traffic engineering aims to reduce the energy consumption of the network by consolidating traffic to fewer links and placing idle network components into power saving modes. Due to the over-provisioning and high-redundancy of the network, significant power savings can be achieved through green traffic engineering. However, exploiting the period of low traffic activity to periodically implementing green traffic engineering is still a challenging problem.

A green OSPF protocol in the work [102] aggregates network traffic at a neighbourhood of routers to place idle links into sleep. In the green OSPF, routers are divided into two categories, including exporter routers to compute their shortest path tree as usual and importer routers to run a modified version of shortest path tree rather than computing on their own. In this way, several routers can share the same shortest path tree so that several idle links can be placed into sleep.

Similarly, a general distributed routing protocol for power saving (GDRP-PS) switches network nodes off during off-peak hours [103]. In this protocol, the decision of a node sleeping is governed by a central node based on the aggregate link utilization of the node. Once in sleep, the sleep node periodically wakes up, re-connects to the network and examines the aggregate link utilization to decide

whether go back into sleep again. Evaluation results indicate that 20% power savings can be achieved through GDRP-PS.

The works in [56] [104] addressed the energy efficiency issue in ICT networks and developed several algorithms that used traffic prediction to set network links to low power idle modes. These algorithms use the upstream interface on the network link to maintain a window of inter packet arrival times. The window of inter packet arrival times is then used to determine the length of time that the network link can be put into sleep. In addition, these algorithms also reduce the probability of the buffer overflow for the network link. The constraint is the non-zero time it takes for the downstream interface to wake up. The results with real traces show that when the traffic load is up to 30% of link capacity, considerable power savings can be achieved.

The work [105] compared the power consumption of network devices under different network categories and results indicated that hubs and switches in local area networks (LAN) account for about 80% of the total power consumption of the entire networks in 2002. In 2005, the work [106] estimated power consumption of the Network Interface Cards (NICs) and other network devices, and results indicated that the power consumption of NICs take up to almost half of the total power consumption of entire networks. Power consumption in core networks is increasing exponentially. In 2009, the work [107] estimated that power consumption of the core networks will be equivalent to the access networks by 2017.

To reduce the power consumption of the core networks, Chiaraviglio et al. [108] proposed an approach to reduce the power consumption of backbone networks by turning off network nodes and links while guaranteeing full connectivity and link utilization constraints. It adopts an integer linear programming (ILP) method to formulate the problem and the energy efficiency of the backbone networks is

improved by minimizing used routers and links while meeting connectivity and quality of service constraints.

Bolla et al. introduced sleep mode into backbone networks [25]. This is achieved by periodically reconfiguring the network nodes and links in response to incoming traffic volumes and operational constraints. Results show that this approach can dynamically put hardware components of network nodes to sleep to reduce the power consumption of backbone networks while maintaining required quality of service.

2.3 Summary

This chapter introduces the existing techniques related to energy efficient networking. In Section 2.1, device level techniques including sleep mode and power scaling are discussed in detail. Since the energy consumption of an idle network device in normal operation mode is almost identical to a fully loaded one, sleep mode reduces power consumption of idle network devices by putting them into low power consumption sleep mode during idle periods of the network devices. Power scaling considers a set of approaches which are designed to dynamically adapt the processing capacity of network devices in response to the current service demand. While, in Section 2.2, energy efficient techniques in network design and green routing algorithms are reviewed. Energy efficient techniques in network design aim to find the design parameters that optimize an objective function to minimize energy consumption, while fulfilling provisioned traffic demand and design specifications (e.g., minimum end-to-end delay, maximum link utilization and reliability). Green routing algorithms are devoted to reducing energy consumption of network by consolidating traffic to a few active links so that some links can go to lower power consumption mode with lower transmission capacity to save power or even go to sleep mode for maximum power saving.

Chapter 3

Architecture of Reference Router and Frequency Adaptive Router

In this chapter, a Frequency Adaptive Router is designed and validated on the NetFPGA 1G board [33]. The NetFPGA 1G board supports a community of open source hardware and software, and makes use of modular structure in the pipeline design. Therefore, it is capable of providing a fast way to develop and experiment custom green mechanisms with low cost reconfigurable features. In the rest of this thesis, the NetFPGA refers to the NetFPGA 1G board for short.

The Frequency Adaptive Router is built based on the NetFPGA Reference Router [109]. The design and architecture of the Reference Router are introduced in Section 3.1 and Section 3.2. Understanding the design and architecture of the NetFPGA Reference Router is the first step to design and develop the Frequency Adaptive Router. Section 3.3 describes the design and architecture of the Frequency Adaptive Router. Section 3.4 summarizes the Chapter.

3.1 The NetFPGA platform

The NetFPGA is an open source and low-cost reconfigurable hardware platform optimized as a high-speed networking router. Before diving into this specific NetFPGA platform, it is crucial to understand what is inside a router. As shown in Figure 3.1, the generic router architecture consists of multiple input ports, multiple output ports, a switching fabric and a routing processor. In practice, multiple ports are often gathered together on a single line card within a router.

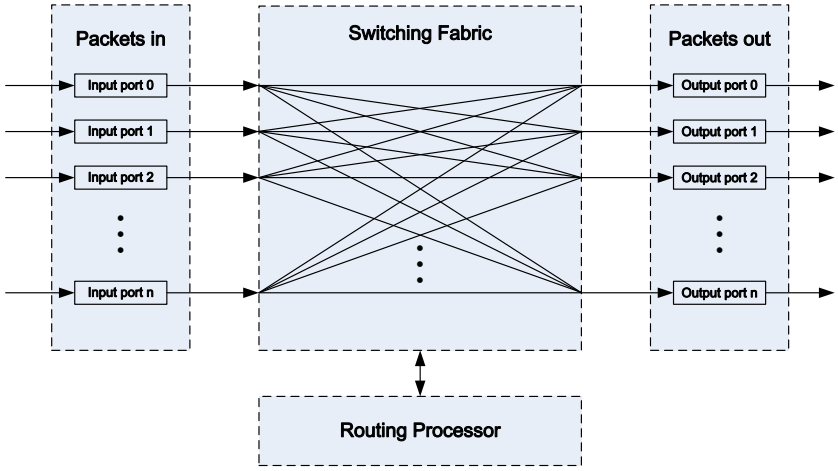

Figure 3.1: Generic router architecture

In the generic router architecture, the input port terminates an incoming physical link to a router and interoperates the data link layer with the other side of the incoming link. While, compared to the input port, the output port performs reverse physical link and data link layer on the outgoing link. The switching fabric connects input ports to corresponding output ports. The routing processor executes routing protocols, maintains routing tables and performs network managements. With lookup and forwarding in the input port, a data packet is pushed into the switching fabric and forwarded to the appropriate output port. While, a control packet is forwarded from the input port to the routing processor. The output port stores the data packet that has been forwarded to it through the switching fabric, and then transmits the data packet on the outgoing link.

The NetFPGA is a Peripheral Component Interconnect (PCI) based board that consists of a Broadcom quad-port Gigabit PHY interfaced with four Gigabit Ethernet Media Access Controllers (EMAC) for sending and receiving packets, two Cypress SRAMs for storing and buffering packets, a small Xilinx Spartan II FPGA for the control logic from the PCI interface to the NetFPGA host, and a larger Xilinx Virtex-II Pro FPGA for the user defined logic programming. The major hardware components of the NetFPGA 1G board are shown in Figure 3.2. The PHY operates at a fixed frequency of 125 MHz and the Spartan II works at a fixed frequency of 62.5 MHz. While, the Virtex II Pro clock which is the core logic clock can be toggled between 125 MHz and 62.5 MHz [110]. The two SRAMs run synchronously with the core logic clock at either 125 MHz or 62.5 MHz.

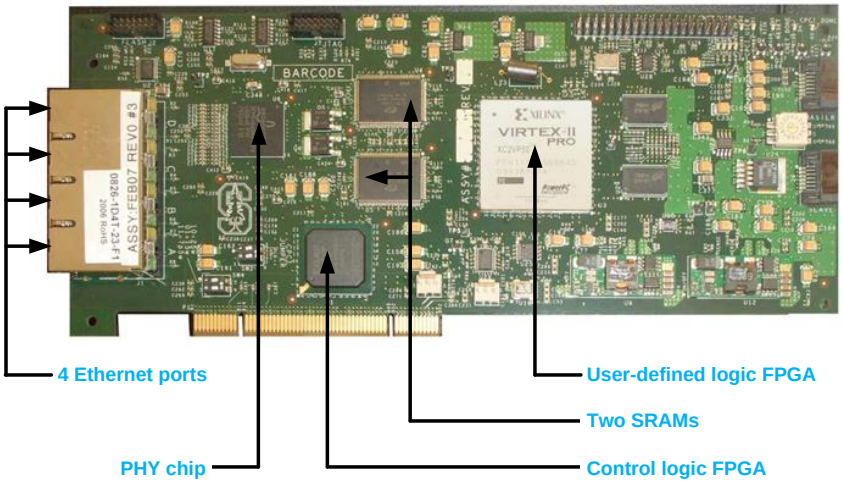

Figure 3.2: Major hardware components of the NetFPGA 1G board [110]

As shown in Figure 3.3, the NetFPGA adopts a modular and re-usable pipeline with a data bus (black solid line) and a register bus (red dotted line). Hardware modules implemented in Verilog can communicate with software running on a host Central Processing Unit (CPU) using Direct Memory Access (DMA) packet transfer and register access over the PCI bus. Both the DMA packet transfer and the register access involve interaction between the NetFPGA hardware and the host software. Figure 3.3 demonstrates the interaction. The registers in the register

system are used to indicate the status information and set the control signals for each separate hardware module. In the NetFPGA Reference Router, there is an inbuilt register in the register system which allows the switching of the operating frequency of the core logic FPGA between 125MHz and 62.5MHz [110]. There are also four other inbuilt registers which control the turning on and off of each of the four Ethernet ports. The control of the core logic FPGA frequency switching and the Ethernet ports turning on and off are implemented by writing to the relevant memory mapped I/O registers through the register bus on the NetFPGA.

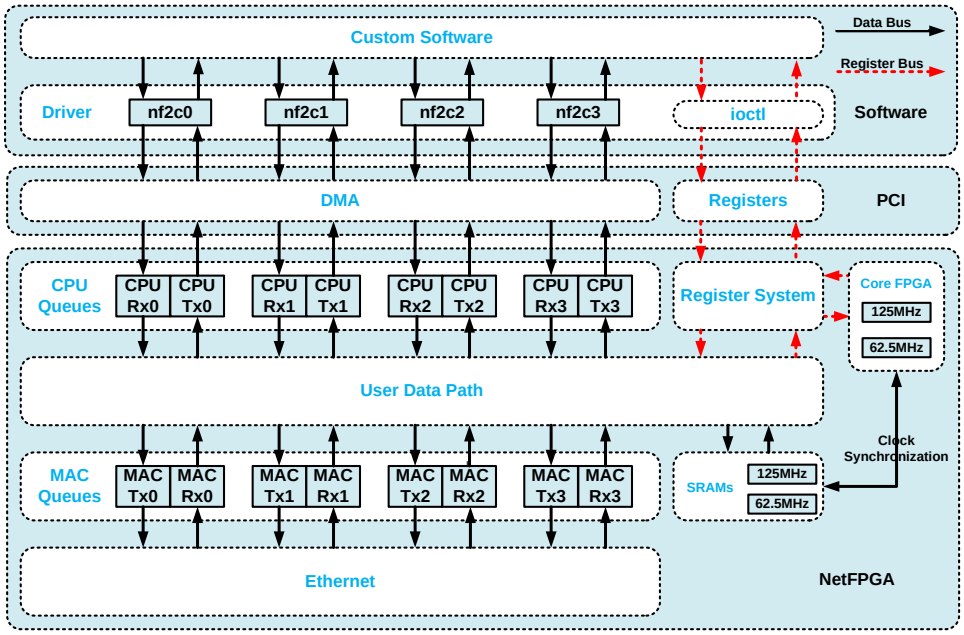

Figure 3.3: NetFPGA hardware and host software interaction in the Reference Router

3.2 The NetFPGA Reference Router

The NetFPGA Reference Router is designed in a modular style [111]. In the reference design, each stage is a separate module, which enables developers to make use of these modules to design and implement their own project without starting from scratch. New functions like energy efficient mechanisms can be integrated by adding custom modules or by modifications to the existing modules.

As shown in Figure 3.4, the NetFPGA Reference Router consists of multiple modules including eight receive queues, eight transmit queues and the user data path [111]. Both receive queues and transmit queues are divided into two groups: four MAC interfaces and four CPU via DMA interfaces. The receive queues receive packets from I/O ports such as the Ethernet ports and the PCI via DMA, while the transmit queues send packets out of the I/O ports instead of receiving.

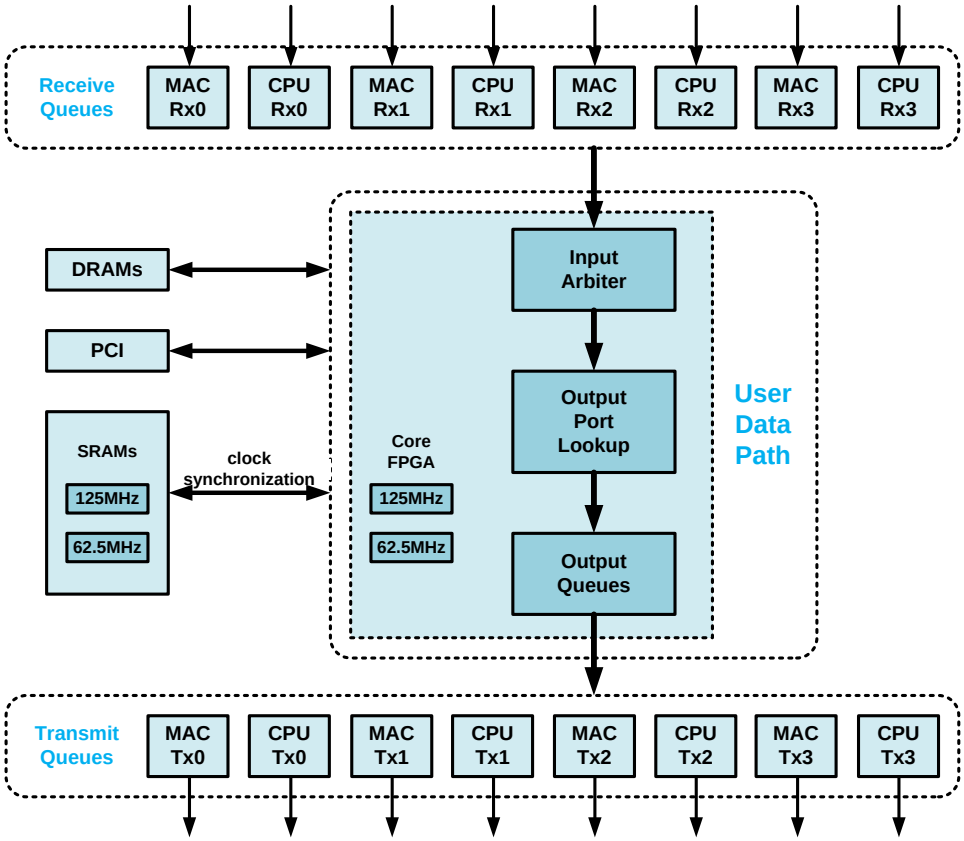

Figure 3.4: The architecture of the Reference Router

3.2.1 Receive Queues

Figure 3.5 presents a block diagram of the data bus and the register bus in the Reference Router. In the first stage, eight receive queues, including four MAC receive queues and four CPU via DMA receive queues, receive data packets from

each I/O port. Then, the Receive Queues module appends a module header for each packet, indicating the length and the ingress port of each packet. After the module header is appended to a packet, the packet is pushed into the Input Arbiter module in the user data path.

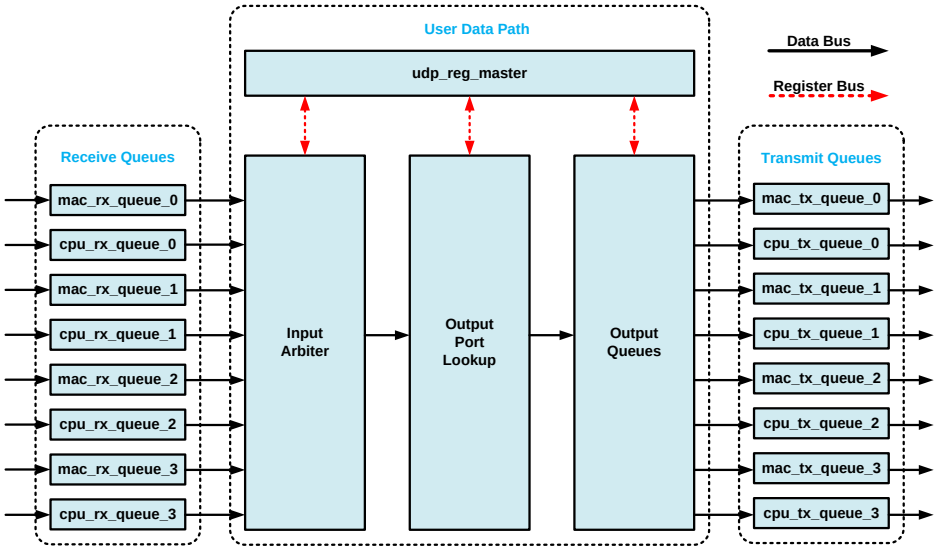

Figure 3.5: Block diagram of data bus and register bus in the Reference Router

3.2.2 User Data Path

The pipeline in the user data path is 64-bit wide and all the internal module interfaces use a standard request-grant First-In-First-Out (FIFO) protocol [111]. In the user data path, the Input Arbiter module decides which receive queue to service next, and pulls a packet from that receive queue and hands it to the Output Port Lookup module. The Output Port Lookup module decides which port the packet goes out of. After that decision is made, the packet is then handed to the Output Queues module which stores the packet in the corresponding output queue and sends the packet out of the output queue when the corresponding transmit queue is ready to accept the packet for transmission. Details about each of the three modules are given below.

Input Arbiter

Figure 3.6 presents a block diagram of the Input Arbiter module. In the user data path, the Input Arbiter module decides which receive queue to service next, and pulls a packet from that receive queue and hands it to the Output Port Lookup module. The decision is made through a packet-by-packet round-robin scheduler.

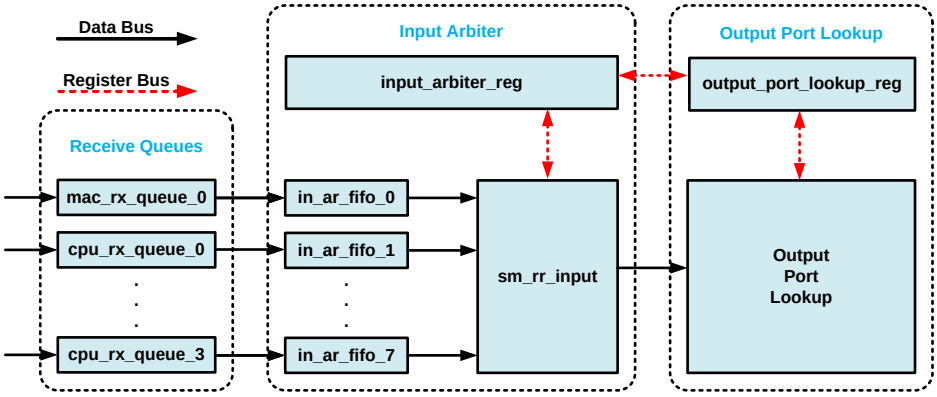

Figure 3.6: Block diagram of the Input Arbiter module

When congestion occurs in a network router, queuing mechanisms can be used to buffer data that the router can not handle at the moment, and send the data out until the bandwidth is available. In best effort communication, first-in first-out (FIFO) queuing is a basic queuing method. Packets are pushed into one single queue and pulled out of the queue on a first-come first-served (FCFS) basis. If multiple FIFO queues are involved, round robin scheduling can be adopted as an alternative to FCFS queuing. In round robin scheduling, priority is not considered and all the queues are equally served in a packet-by-packet round robin fashion.

Figure 3.7 demonstrates an example of the round robin scheduler. For simplicity, assuming there are two packets A_1 and A_2 in queue 1, and two packets B_1 and B_2 in queue 2 for transmission. One packet is taken from each of the FIFO queues

in turn in circular order for processing. Thus, the output from the round robin scheduler is A_1, B_1, A_2 and B_2.

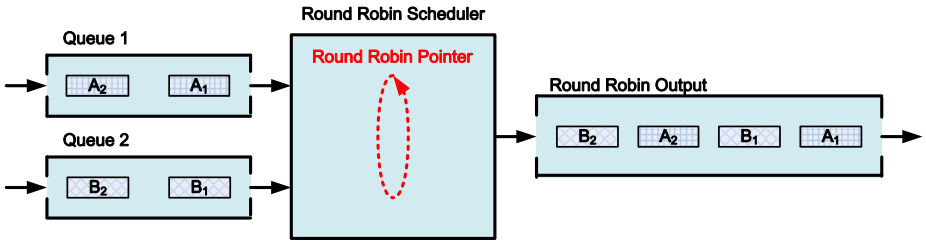

Figure 3.7: The round robin scheduler

Output Port Lookup

Figure 3.8 presents a block diagram of the Output Port Lookup module. Data packets received from the Input Arbiter module are pushed into a FIFO queue and preprocessed in a *preprocess_control* block. Five operations are performed in the *preprocess_control* block simultaneously. The submodule *ip_lpm* and the submodule *ip_arp* decide the next hop MAC address and the corresponding output port. The submodule *eth_parser* checks the destination MAC address and the packet type. The submodule *ip_cheksum_ttl* validates the IP checksum and creates the new checksum. The submodule *op_lut_hdr_parser* checks whether the packet is from the CPU via DMA. The submodule *dest_ip_filter* checks the destination IP address and decides whether the packet goes to the CPU via DMA.

After preprocessing, the packet is pulled out of the FIFO queue to the *op_lut_process_sm* block. The *op_lut_process_sm* block modifies the packet with updated information from the *preprocess_control* block (such as modifying the IP TTL, the next hop MAC address and source address), and sends the packet out to the corresponding output queue.

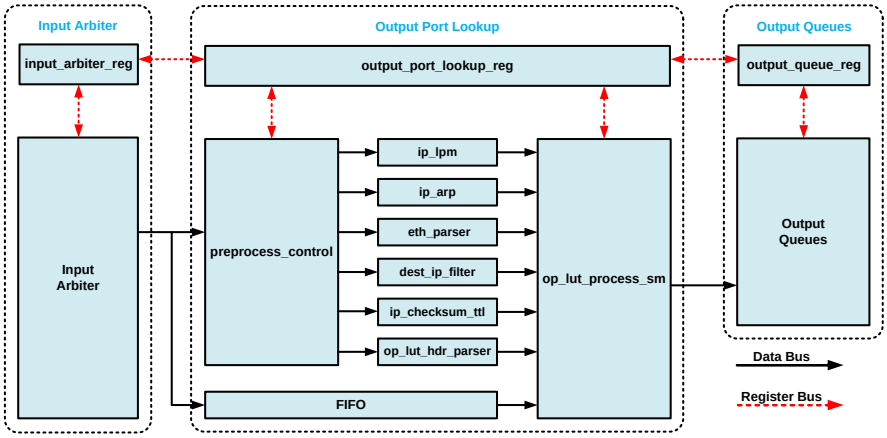

Figure 3.8: Block diagram of the Output Port Lookup module

Output Queues

Figure 3.9 presents a block diagram of the Output Queues module. The Output Queues module stores packets received from the Output Port Lookup module into the SRAMs and implements a round robin arbiter to service the output queues. The round robin arbiter operates in the same packet-by-packet round robin fashion as shown in Figure 3.7. The *oq_header_parser* block checks the destination port of a packet, and the packet is then pushed into a FIFO queue waiting to be processed. The *store_pkt* block pulls packets from the FIFO queue and stores them into the SRAMs. While, the *remove_pkt* block implements a round robin arbiter over the output queues, reads a packet from the SRAMs, and sends it to a corresponding transmit queue.

3.2.3 Transmit Queues

The Transmit Queues module strips out the module headers appended in the Receive Queues module and sends packets out through the corresponding output port to either one of the four MAC transmit queues or one of the four CPU via DMA queues.

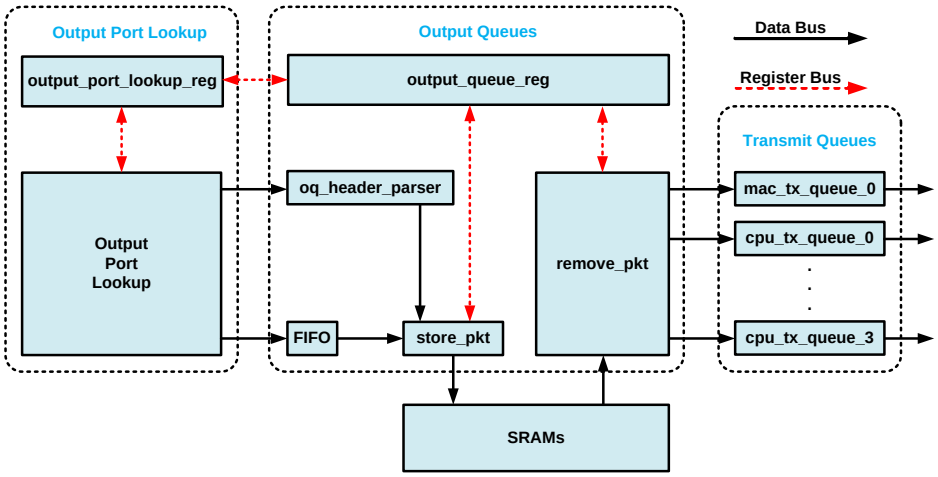

Figure 3.9: Block diagram of the Output Queues module

3.3 Frequency Adaptive Router

To improve the energy efficiency of the Reference Router without compromising its packet processing performance, it is modified to adapt its operating frequency (also known as its packet processing capacity) in response to the incoming traffic load. A frequency division module and an Asynchronous FIFO (AFIFO) module are integrated into the Reference Router, and the modified router is referred to a Frequency Adaptive Router. Figure 3.10 presents the architecture of the Frequency Adaptive Router. Details about the frequency division module and the AFIFO module are described below.

3.3.1 The Frequency Division Module

Understanding the clock resources of the Virtex-II Pro FPGA is the first step to add more frequency options into the Reference Router. The Virtex-II Pro FPGA clock resources consist of global clock input pads (GCLK), global input clock buffers (IBUFG), global clock buffers (BUFG), and digital clock managers (DCM) [112]. As shown in Figure 3.11, the global clock input pad (GCLK) can be fed directly into a low-skew global clock network through an IBUFG and a BUFG

[113]. The IBUFG is an input clock buffer with one clock input and one clock output. While, the BUFG is a global clock buffer with one clock input and one clock output, driving a low skew clock distribution network. As shown in Figure 3.12, a DCM can be inserted between the IBUFG and the BUFG to provide advanced clocking capabilities [113].

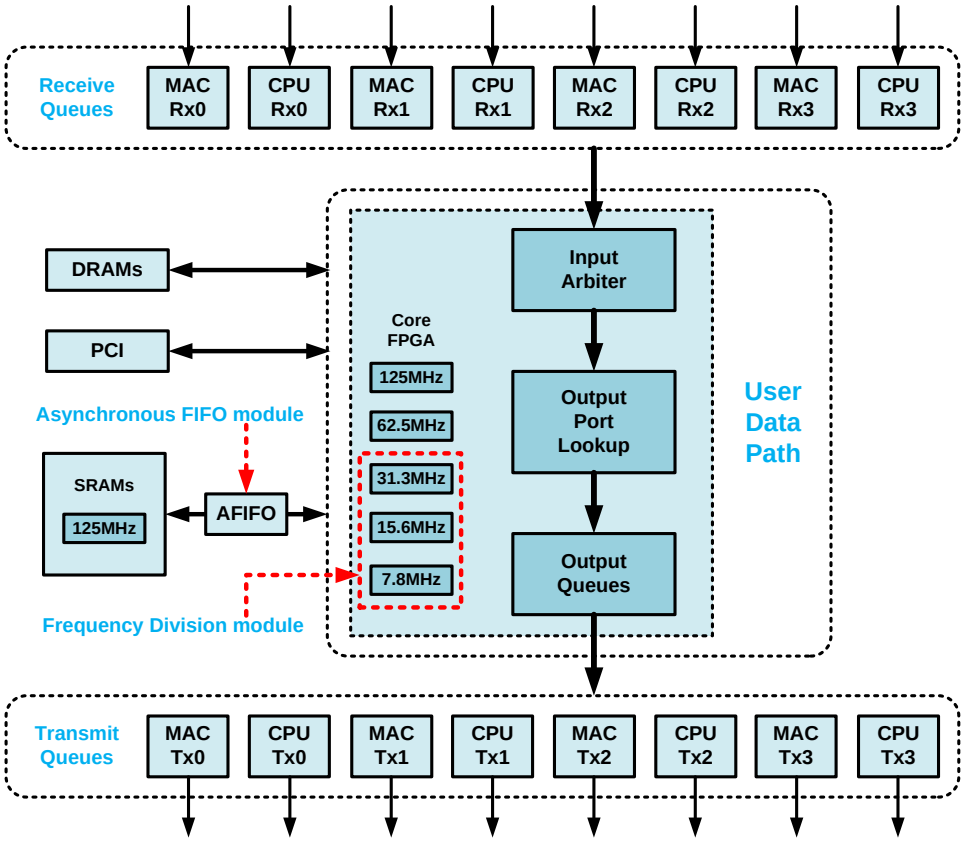

Figure 3.10: The architecture of the Frequency Adaptive Router

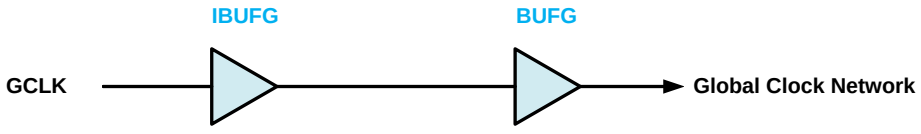

Figure 3.11: A GCLK fed into a low-skew global clock network through an IBUFG and a BUFG [113]

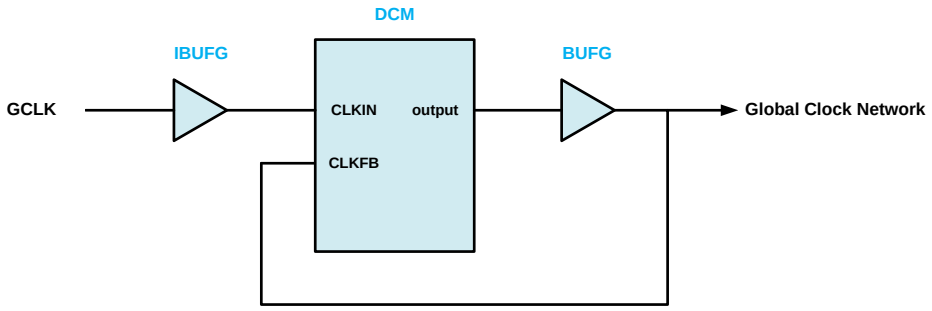

Figure 3.12: A GCLK fed into a low-skew global clock network through an IBUFG, a DCM and a BUFG [113]

Figure 3.13 demonstrates the global clock design of the Reference Router. The clock for the control logic FPGA is fed by 62.5 MHz through an IBUFG and a BUFG. The clock for the core logic FPGA, the transmit clock and the receive clock for the four Ethernet ports are all fed by 125 MHz through an IBUFG, a DCM and a BUFG.

The DCM can generate new clock frequencies by dividing source clock frequency with allowed divisors. Figure 3.14 presents an example of generating a new frequency using the DCM. 125 MHz is fed into clock input CLKIN in the DCM through a IBUFG. In the DCM, the clock output CLK0 is the same as the input clock CLKIN at 125 MHz, while the clock output CLKDV is the input clock CLKIN divided by a factor of 16, providing a much lower clock output at 7.8 MHz. A global clock buffer multiplexer (BUFGMUX) is used to select between the two clock outputs from the DCM module. When the select input S is Low, the 125 MHz clock at input I_0 of the BUFGMUX is selected for the output. When the select input S is High, the 7.8 MHz clock at input I_1 of the BUFGMUX is selected for the output.

With CLKDV in the DCM, the Frequency Adaptive Router can offer 23 grades of operating frequencies. This is accomplished by dividing source clock frequency with 23 different divisors allowed on the Virtex II Pro FPGA. A wide range of 23

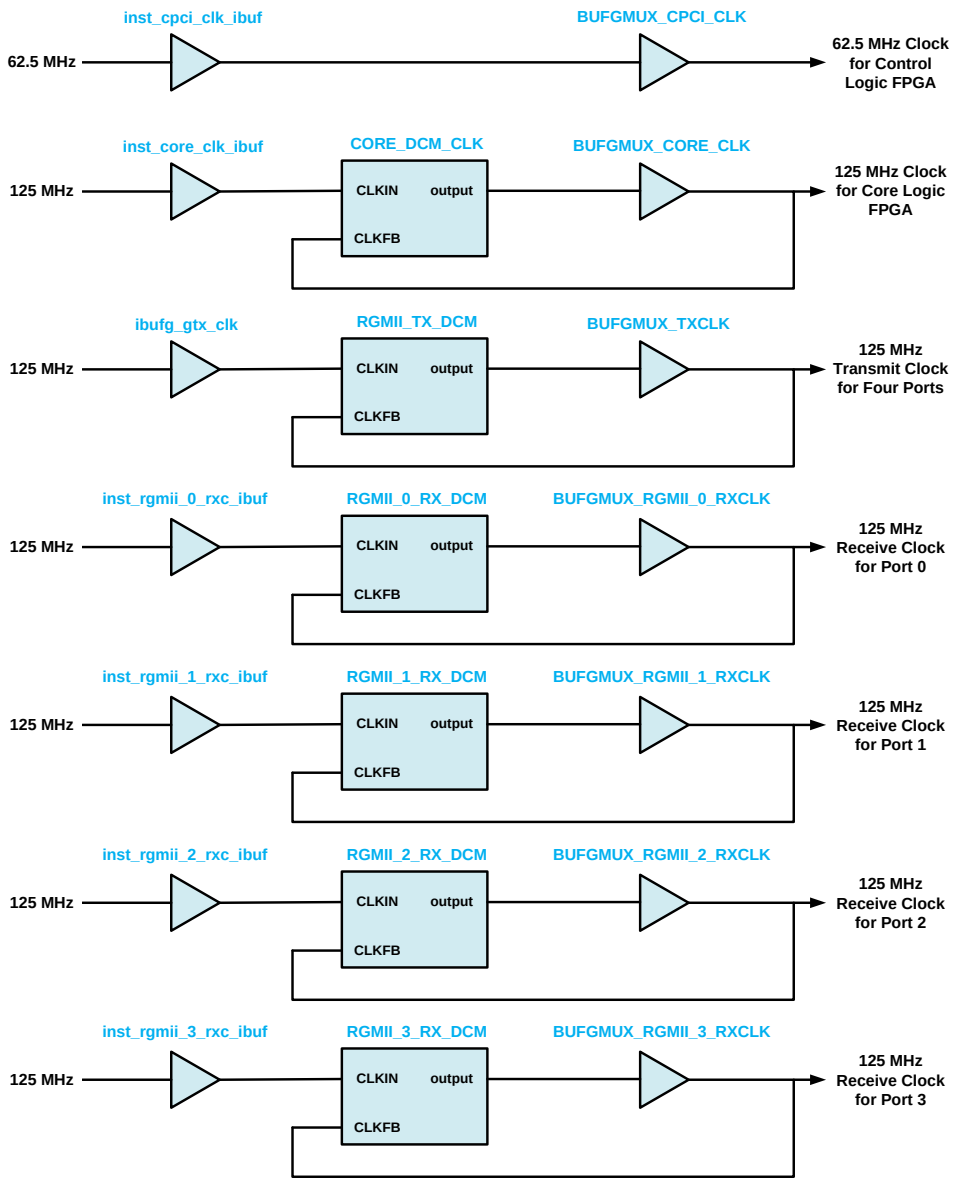

Figure 3.13: Global clock design of the Reference Router

operating frequency options (125MHz, 83.3MHz, 62.5MHz, 50MHz, 41.7MHz, 35.7MHz, 31.3MHz, 27.8MHz, 25MHz, 22.7MHz, 20.8MHz, 19.2MHz, 17.9MHz, 16.7MHz, 15.6MHz, 13.9MHz, 12.5MHz, 11.4MHz, 10.4MHz, 9.6MHz, 8.9MHz, 8.3MHz and 7.8MHz) can be derived from the source clock 125MHz by simultaneous frequency division with a set of divisors (1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 9, 10, 11, 12, 13, 14, 15 and 16). Table 3.1 presents the DCM

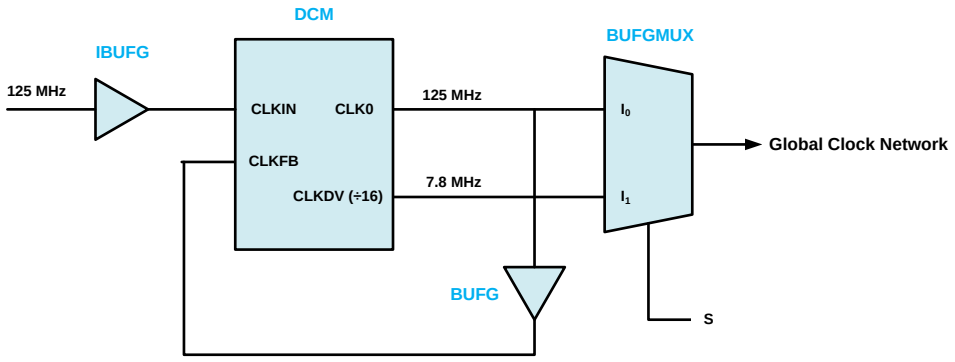

Figure 3.14: Dividing source clock using the DCM

output frequencies corresponding to the different frequency divisors. The theoretical throughputs of the Reference Router corresponding to the different DCM output frequencies are also listed. The theoretical throughput is linearly proportional to the operating frequency. Figure 3.15 visually presents the linear pattern and relationship between the operating frequency and the theoretical throughput.

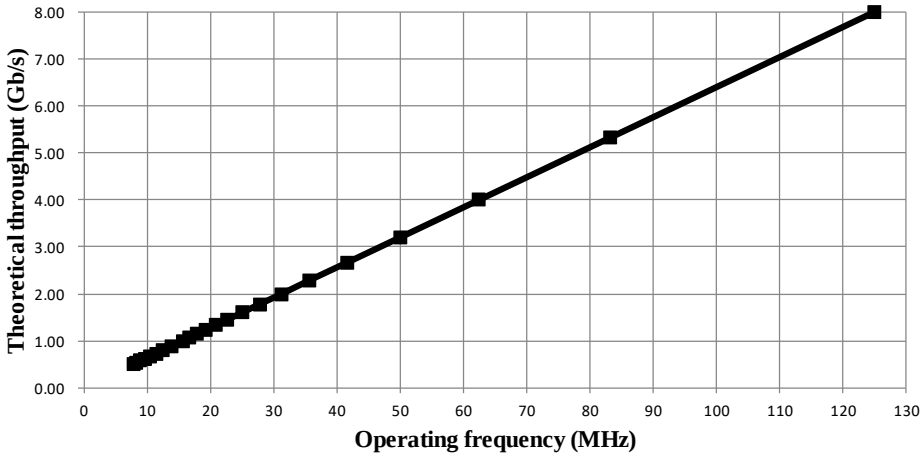

Figure 3.15: The linear relationship between the operating frequency and the theoretical throughput

The Virtex-II Pro FPGA provides 8 DCMs and the Reference Router uses 6 out of 8 DCMs as shown in Figure 3.13. The limited clock resources in the Virtex II

Table 3.1: DCM frequency divisors, corresponding DCM output frequencies and theoretical throughputs of RR

DCM frequency divisors (Value)	DCM output frequencies (MHz)	theoretical throughputs of RR (Gb/s)
1	125	8
1.5	83.3	5.33
2	62.5	4
2.5	50	3.20
3	41.7	2.67
3.5	35.7	2.29
4	31.3	2
4.5	27.8	1.78
5	25	1.60
5.5	22.7	1.45
6	20.8	1.33
6.5	19.2	1.23
7	17.9	1.14
7.5	16.7	1.07
8	15.6	1
9	13.9	0.89
10	12.5	0.80
11	11.4	0.73
12	10.4	0.67
13	9.6	0.62
14	8.9	0.57
15	8.3	0.53
16	7.8	0.50

Pro FPGA cannot allow to implement these 23 frequencies at the same time. In the design of the Frequency Adaptive Router, three additional frequencies 31.3 MHz, 15.6 MHz and 7.8 MHz with frequency divisor 4, 8 and 16 are added into the Reference Router to balance the trade-off between power consumption and performance. Figure 3.16 demonstrates the design details for generating the three new frequencies in the Frequency Adaptive Router. The 7.8 MHz is generated by the CLKDV in the core clock DCM (CORE_DCM_CLK) with frequency divisor 16. The 15.6 MHz is generated by the CLKDV in the receive clock port 0 DCM (RGMII_0_RX_DCM) with frequency divisor 8. The 31.3 MHz is generated by the CLKDV in the transmit clock DCM (RGMII_TX_DCM) with frequency divisor 4.

There are also other options to generate these three new frequencies from other DCMs. For example, other than generating the three new frequencies the way shown in Figure 3.16, the 7.8 MHz can be generated by the CLKDV in the receive clock port 0 DCM (RGMII_0_RX_DCM) with frequency divisor 16. The 15.6 MHz can be generated by the CLKDV in the receive clock port 1 DCM (RGMII_1_RX_DCM) with frequency divisor 8. The 31.3 MHz can be generated by the CLKDV in the receive clock port 2 DCM (RGMII_2_RX_DCM) with frequency divisor 4. However, experiments demonstrate that, using the other options to generate the three new frequencies, the router may not work at all even though the Verilog code is synthesized successfully. No further design effort was spent on the other options to generate the three additional frequencies, after the design in Figure 3.16 is tested to guarantee the router can properly work and provide five scalable frequencies to the core logic FPGA.

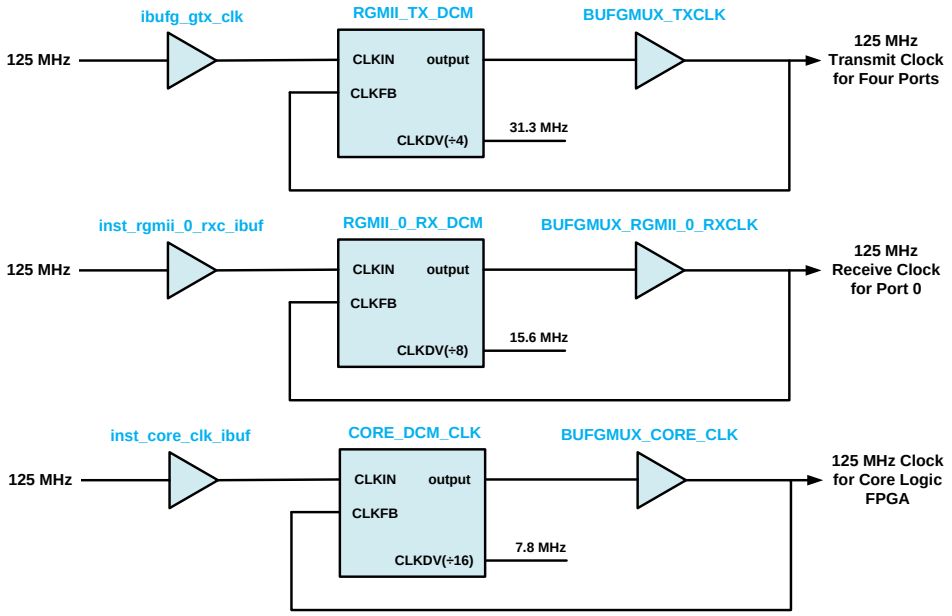

Figure 3.16: Design details for generating three new frequencies in the Frequency Adaptive Router

The design in Figure 3.16 is customizable. Other frequencies in Table 3.1 can be also used instead of 31.3 MHz, 15.6 MHz and 7.8 MHz. For example, an earlier version of a Frequency Adaptive Router implemented 125 MHz, 83.3 MHz, 62.5 MHz, 50 MHz and 41.7 MHz. The 83.3 MHz, 50 Mhz and 41.7 MHz are generated using frequency divisor 1.5, 2.5 and 3. The design in Figure 3.16 provides flexibility for other users to customize the generated frequencies to meet their specific needs.

After the three additional frequency options are provided, four BUFGMUX resources are then cascaded to create a 5-to-1 clock multiplexer as shown in Figure 3.17. 125 MHz, 62.5 MHz, 31.3 MHz, 15.6 MHz and 7.8 MHz are the source clock inputs for the cascaded BUFGMUXs, while S_0, S_1, S_2 and S_3 are the control inputs. With four cascaded BUFGMUXs, the four control inputs decide which one clock is selected from the five source clock inputs for output. For example, if S_0 is High, the selected clock for the core logic FPGA clock is 125 MHz. 7.8 MHz is the selected clock if S_0, S_1, S_2 and S_3 are all Low. Table 3.2 presents the mapping between the control inputs and the selected clock output. Value '1' for a control input stands for the control input is High, value '0' represents Low, while '-' means doesn't care.

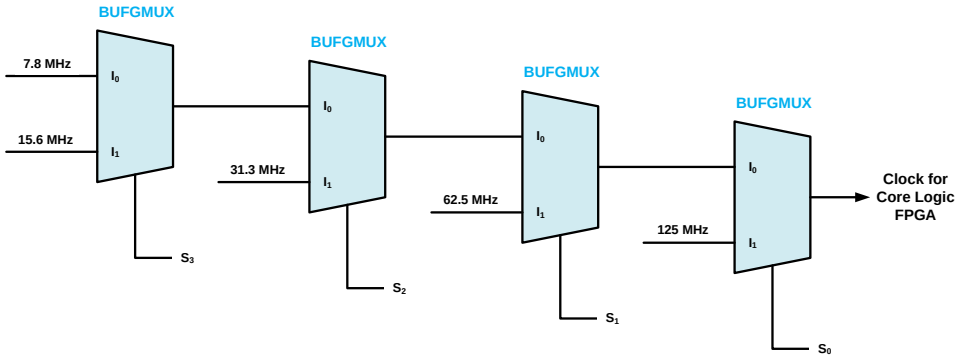

Figure 3.17: 5-to-1 cascaded clock multiplexer

Table 3.2: Mapping between control inputs and selected clock output

S_0 (value)	S_1 (value)	S_2 (value)	S_3 (value)	clock output (MHz)
1	-	-	-	125
0	1	-	-	62.5
0	0	1	-	31.3
0	0	0	1	15.6
0	0	0	0	7.8

A custom frequency control register 0x200218 is further integrated into the register system to allow software running on the NetFPGA host PC to communicate with the NetFPGA hardware, and switch the core logic FPGA frequency among these five different frequencies. The Table 3.3 indicates the mapping between the custom register values and the operating frequencies of the core logic FPGA.

Table 3.3: Frequency control register values and corresponding operating frequencies of core logic FPGA

register (value)	operating frequency (MHz)
0	125
1	62.5
2	31.3
3	15.6
4	7.8

3.3.2 The AFIFO Module

In the Reference Router, the two SRAMs use the same clock as used by the core logic FPGA processor for writing and reading data, to ensure the transmit queues could transmit data with little or no delay between packets [109]. An inbuilt register controls the core logic FPGA to operate at either 125 MHz or 62.5 MHz. However, due to the clock synchronization between the SRAMs and the core logic FPGA, the Reference Router cannot switch the operating frequency of the core

logic FPGA on the fly. When toggling the operating frequency of the Reference Router between 125 MHz and 62.5 MHz, the frequency switching causes a board reset to restart the SRAMs and the core logic FPGA hardware with updated synchronous frequency. The board reset involves remirroring and reloading MAC addresses, IP addresses, routing table and ARP table into the core logic FPGA hardware, which takes approximately 2 ms. All the buffered packets are lost during the board reset.

Most applications are able to gracefully handle this board reset. When an user application realizes that packets are lost, the lost packets will be re-transmitted if the application is a file download, an email, or other none real-time applications. The user probably won't ever notice the board reset. However, if the application is a real-time conversation, a video conference or other real-time applications that has a low tolerance for packet loss, the effect of board reset could be very noticeable to the user. The audio could be distorted and the video could show artifacts. The user may even suffer from connection lost if the packet loss is severe.

To eliminate the board reset problem, a custom module of asynchronous FIFO (AFIFO) is inserted between the SRAMs and the core logic FPGA. The AFIFO allows safe data exchange between the SRAMs clock domain and the core FPGA clock domain, where the two clock domains are asynchronous to each other. The AFIFO involves a FIFO design where packets are written into the FIFO using a clock from the SRAMs clock domain for buffering, and the packets are read from the same FIFO using a clock from the core logic FPGA clock domain for transmitting. The AFIFO module can isolate the SRAMs alone and keep them running at 125MHz constantly, while the operating frequency of the core FPGA can be tuned among allowed frequencies in response to actual traffic processing

needs. Figure 3.18 demonstrates the NetFPGA hardware and host software interaction in the Frequency Adaptive Router.

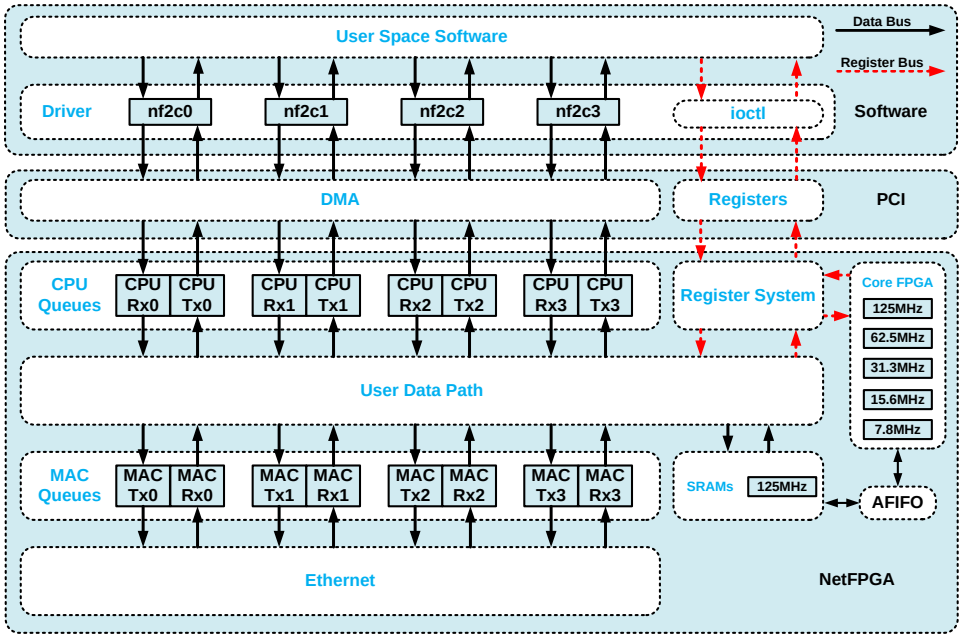

Figure 3.18: NetFPGA hardware and host software interaction in the Frequency Adaptive Router

3.4 Summary

This chapter presents the design of the Frequency Adaptive Router and compares the design differences between the Reference Router and the Frequency Adaptive Router. Since there are only two operating frequencies 125 MHz and 62.5 MHz provided in the Reference Router, it is insufficient to examine and quantify the power savings from dynamic frequency scaling. To better quantify the power savings from energy proportional techniques, the Frequency Adaptive Router is developed to provide the core logic FPGA with five operating frequency options (125 MHz, 62.5 MHz, 31.3 MHz, 15.6 MHz and 7.8 MHz).

Chapter 4

Frequency Control Policies and System Models

Once the frequency scaling capability is provided, appropriate frequency control policy must be implemented to choose the most desirable operating frequency to meet certain performance criteria. Previous work [35] for Adaptive Link Rate (ALR), the control policy adopts output buffer thresholds and utilization monitoring to switch link data rate. Similar work [34] for Multi-Frequency Scaling (MFS) on the NetFPGA, the control policy adopts buffer occupancy and multi-dual-threshold to adapt the operating frequency. However, the power consumption and performance evaluation reported in these works are based on simulated results. For example, multi-dual-threshold policy in the work [34] is evaluated using a Markov model. To provide more accurate and valid power consumption and performance evaluation results, the work develops and implements the Frequency Adaptive Router and its frequency control policies directly in NetFPGA hardware.

As explained in chapter 3, the Frequency Adaptive Router can operate at five different frequencies with five different packet routing capacities. The frequency

control policy is crucial in the Frequency Adaptive Router because it manages the operating frequency of the router, which in turn affects the power consumption and the performance of the router. The frequency control policies must be as simple as possible for direct implementation inside the router. An effective policy should manage the router to operate at the lowest frequency with enough routing capacity to handle the instantaneous traffic without affecting the performance as perceived by the user. Significant increases to network delay and packet loss could degrade user experience, which would be unacceptable.

To meet different demands under different application scenarios, six different frequency control policies are designed to balance the trade-off between power consumption and performance. Each control policy has an unique behavior, purpose and suitability in response to the instantaneous traffic load. The six implemented frequency control policies are: Single Threshold Policy (STP), Double Threshold Policy (DTP), Packet Loss Aware Policy (PLAP), Performance First Policy (PFP), Power Saving First Policy (PSFP) and User Defined Policy (UDP).

These six frequency control policies can be divided into three self-adaptive local frequency control policies and three policies for global frequency control. The first three frequency control policies implemented in the NetFPGA hardware can locally and dynamically adapt the operating frequency in response to the instantaneous traffic load. While the three latter ones implemented in software can work with network wide power management technique, such as green traffic engineering (GTE), which can globally control the operating frequency of the Frequency Adaptive Router through reading and writing the custom frequency control register 0x200218 in Table 3.3.

4.1 Local Frequency Control Policies

For dynamic frequency adaptation in the Frequency Adaptive Router, three frequency control policies STP, DTP and PLAP are designed and implemented based on statistics monitoring and preset thresholds as shown in Figure 4.1. The statistics monitoring is designed to indicate the current core logic frequency, the total current traffic rate from all receive queues (*current_traffic_rate*) and the total number of bytes dropped from all output queues (*byte_counter_dropped*) in 10 ms sampling period. The preset thresholds are adopted to divide the routing capacity into five grades in response to the incoming traffic load in the previous 10 ms sampling period. Different frequency control policy may have a different set of rules for setting the thresholds.

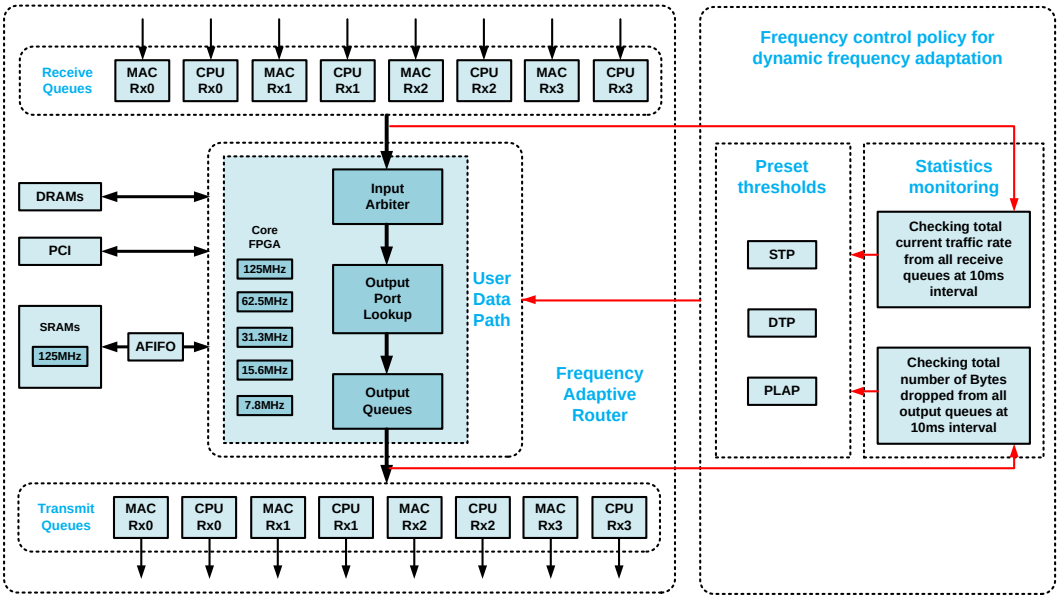

Figure 4.1: Implementation of dynamic frequency adaptation on FAR

With the statistics monitoring and the preset thresholds, the frequency adaptation transition time in the local frequency control policies is the sum of two components: the delay T_1 between the need for a new frequency and the issue of

the frequency transition command, and the delay T_2 between the issue of the frequency transition command and the completion of the frequency transition. The delay T_1 is determined by the sampling period and the delay T_2 is determined by the speed of digital electronic circuits hardware.

Figure 4.2 explains the frequency adaptation transition time in the local frequency control policies. Figure 4.2 (a) presents a simple example of traffic burst from 0 Mb/s (idle) to aggregate input traffic 6,400 Mb/s. Figure 4.2 (b) presents the operating frequency corresponding to the traffic burst in Figure 4.2 (a). The corresponding T_1 and T_2 are shown in Figure 4.2. At the sampling time points *1* and *2*, traffic load increases gradually from the 0 Mb/s within the routing capacity of the operating frequency at 7.8 MHz, so no frequency switching is necessary. At the time point *A*, traffic load increases significantly to the aggregate input traffic 6,400 Mb/s, so a new operating frequency 125 MHz is needed to handle the current 6,400 Mb/s traffic load. However, the control policy only checks the traffic load at the next sampling time point *3*, finds out there is a need for a new operating frequency 125 MHz and issues the frequency transition command.

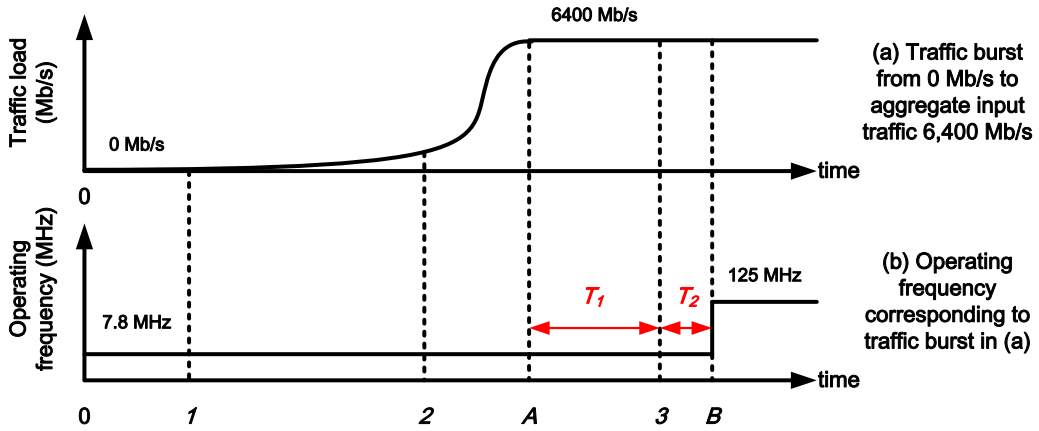

Figure 4.2: Analysis of frequency adaptation transition time in local frequency control policies

The sampling period determines the maximum delay between the time point *3* and the time point *A*. If the traffic burst occurs immediately after the sampling action at time point *2*, then the maximum delay between the sampling point *3* and the time point *A* is the sampling period. If the traffic burst occurs immediately before the sampling action at time point *3*, then the minimum delay between the sampling point *3* and the the time point *A* is 0. Thus, the sampling period directly determines the delay T_1 between the need for a new frequency at the time point *A* and the time that the control policy finds out the need for a new frequency at the time point *3*. While, the delay T_2 between a request for a new frequency at the time point *3* and the actual completion of transition to the new frequency at the time point *B*, is not dependent on the sampling period. It is dependent on the speed of digital electronic circuits that perform the frequency transition.

Table 4.1 presents the packet loss rate of FAR under different sampling period (1 ms, 10 ms, 100 ms and 1 s) with the same experimental scenario corresponding to the traffic burst in Figure 4.2. In this experimental scenario, there are 40,000,000 packets sent in total. The packet loss only occurs during the time of frequency switching from 7.8 MHz to 125 MHz, and the total number of packet loss is recorded. Experimental results in Table 4.1 indicate that higher sampling period could lead to higher packet loss rate during the frequency switching. The sampling period is extremely sensitive regarding the performance of the FAR. The results also indicate that 10 ms is a reasonable interval time for the sampling period and is consistent with that used in implementing adaptive link rate (ALR) [69].

Table 4.1: Packet loss rate of FAR under different sampling period at traffic rate 6400 Mb/s

sampling period (ms)	packets dropped (number)	packet loss rate (%)
1	2,488	0.006
10	21,675	0.054
100	61,689	0.154
1,000	498,930	1.247

4.1.1 Single Threshold Policy (STP)

Due to the finite queue size, the Single Threshold Policy is to assess an associated traffic throughput threshold beyond which the router will begin to loss packets for each operating frequency. Table 4.2 presents the mapping table of operating frequencies, capacity states, measured capacities and preset thresholds for STP. As shown in Figure 4.3, the capacity states S_1, S_2, S_3, S_4 and S_5 represent the routing capacities of the FAR working at core operating frequency of 7.8 MHz, 15.6 MHz, 31.3 MHz, 62.5 MHz, and 125 MHz respectively. C_i represents the peak measured throughput for each capacity state S_i. At 125 MHz, the peak measured throughput C_5 is below the theoretical full operation throughput 8 Gb/s which is the maximum throughput for the NetFPGA Reference Router. For each capacity state S_i, the threshold T_i should not be preset too aggressively and too close to C_i. Because the statistics monitoring is implemented by checking current traffic rate at a fixed sampling period and the statistics may vary slightly from time to time. The current traffic rate in Gb/s is calculated from the total number of bytes received from all receive queues in 10 ms. Thus, T_i must be set lower than C_i and experiments indicate a reasonable value for T_i is 95% of C_i. A level of 95% was chosen for consistency with prior work on ALR [69], as was the 10 ms sampling period. The choice of these parameters involves a trade-off between power saving and packet loss.

Table 4.2: Frequencies, capacities and thresholds mapping table for STP

Index (No.)	Frequency f_i (MHz)	State S_i	Capacity C_i (Gb/s)	Threshold T_i (95% of C_i) (Gb/s)
1	7.8	S_1	0.443	0.421
2	15.6	S_2	0.878	0.834
3	31.3	S_3	1.714	1.628
4	62.5	S_4	3.457	3.284
5	125	S_5	6.896	6.551

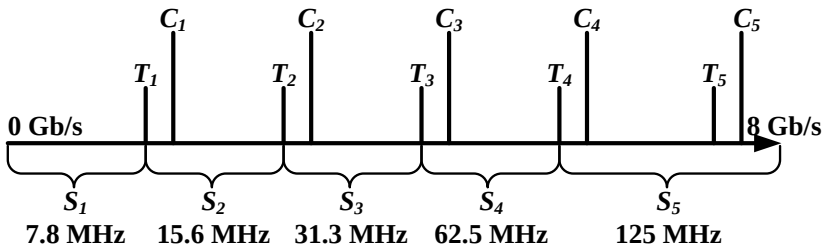

Figure 4.3: Thresholds of single threshold policy

Suppose the FAR is working in state S_i (i = *1 to N*), once the total current traffic rate from all receive queues (which is checked at 10 ms intervals) is below or equal to a threshold T_i ($1 \leqslant$ i $\leqslant$ N-1), the router can switch to the lower capacity state S_i consuming less power while maintaining QoS. On the contrary, if the total current traffic rate from all receive queues is beyond the threshold T_i ($1 \leqslant$ i $\leqslant$ N-1), the router must switch to the higher capacity state S_{i+1} with more routing capacity to avoid packet loss resulted from queue overflow. Otherwise, the router stays in the current capacity state S_i. Algorithm 1 presents the pseudo-code of the self-adaptive local frequency control policy STP.

Algorithm 1: Single Threshold Policy (STP)

Input: $current\_traffic\_rate$
Output: op_freq // operating frequency

N = 5 // number of working frequencies, default to 5
i = 1 // current working frequency index, initialize to 1
sampling_period = 10 ms // sampling period, default to 10 ms
op_freq = 125 // current working frequency, initialize to 125 MHz
$f_n = \{7.8, 15.6, 31.3, 62.5, 125\}$ // set of working frequencies in MHz
$T_n = \{0.421, 0.834, 1.628, 3.284, 6.551\}$ // thresholds for each f_i in Gb/s
while $(STP \&\& i < N)$ **do**
    if $current\_traffic\_rate \leq T_i$ **then**
        $op\_freq \leftarrow f_i$
    else
        i++
    end if
end while
if $(i == N)$ **then**
    $op\_freq \leftarrow f_N$// set to maximum working frequency
end if
$wait for sampling\_period$
// update current_traffic_rate every sampling period

4.1.2 Double Threshold Policy (DTP)

The STP is an elegant scheme if the incoming traffic does not change frequently and abruptly so that the Frequency Adaptive Router could switch its capacity state in response to real-time traffic load without impact on QoS. However, when the incoming traffic fluctuates drastically, especially in the case when input traffic fluctuates near a preset threshold, frequently switching the capacity state could introduce extra power consumption and even extra packet loss. An effective frequency adaptation control policy should manage the Frequency Adaptive Router to operate at the lowest appropriate frequency for as long as possible. Otherwise, switching energy overhead and potential QoS degradation during transition time may outweigh the power savings from the DFS scheme.

Table 4.3: Frequencies, capacities and thresholds mapping table for DTP

Index (No.)	Frequency f_i (MHz)	State S_i	Capacity C_i (Gb/s)	Threshold	
				T_{il} (85% of C_i) (Gb/s)	T_{ih} (95% of C_i) (Gb/s)
1	7.8	S_1	0.443	0.368	0.421
2	15.6	S_2	0.878	0.746	0.834
3	31.3	S_3	1.714	1.457	1.628
4	62.5	S_4	3.457	2.938	3.284
5	125	S_5	6.896	5.862	6.551

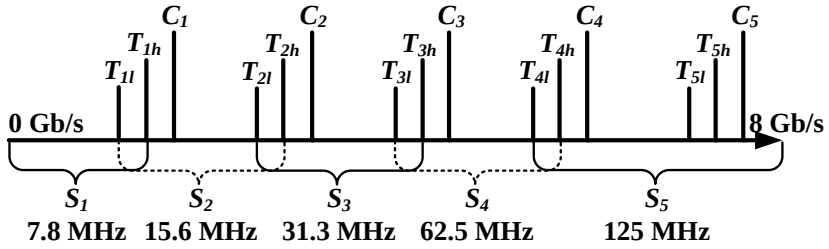

Figure 4.4: Thresholds of double threshold policy

To eliminate the problem of possible oscillation of capacity changes around a threshold associated with the STP, double thresholds are introduced as shown in Figure 4.4. The Double Threshold Policy uses a high threshold T_{ih} ($1 \leqslant i \leqslant 5$), and a low threshold T_{il} ($1 \leqslant i \leqslant 5$) for each capacity state S_i ($1 \leqslant i \leqslant 5$). In the DTP, the buffer between T_{il} and T_{ih} could effectively reduce the possibility of ping pong effect so as to avoid energy waste and overheads on too frequent capacity switching. In addition, the frequency scaling in the DTP is following a level-by-level mechanism, which means that switching only appears between adjacent frequency levels. Table 4.3 presents the mapping table of operating frequencies, capacity states, measured capacities and preset thresholds for DTP. For a certain capacity state S_i, if the total current traffic rate from all receive queues is beyond or equal to $T_{(i-1)l}$ and below or equal to T_{ih}, the router will stay in the current capacity state S_i. The FAR will change to a lower routing capacity only when the total current traffic rate from all receive queues is less than $T_{(i-1)l}$ or, change to a higher routing capacity when the total current traffic rate from all

receive queues is more than T_{ih}. Algorithm 2 presents the pseudo-code of the self-adaptive local frequency control policy DTP.

Algorithm 2: Double Threshold Policy (DTP)

Input: $current\_traffic\_rate, current\_frequency\_index$
Output: op_freq // operating frequency

N = 5 // number of working frequencies, default to 5
sampling_period = 10 ms // sampling period, default to 10 ms
op_freq = 125 // current working frequency, initialize to 125 MHz
$f_n = \{7.8, 15.6, 31.3, 62.5, 125\}$ // set of working frequencies in MHz
$T_{nl} = \{0.368, 0.746, 1.457, 2.938, 5.862\}$
// lower thresholds for each f_i in Gb/s
$T_{nh} = \{0.421, 0.834, 1.628, 3.284, 6.551\}$
// higher thresholds for each f_i in Gb/s
$i \leftarrow current\_frequency\_index$
if $(2 \leq i \leq N-1)$ **then**
    if $T_{(i-1)l} \leq current\_traffic\_rate \leq T_{ih}$ **then**
        $op\_freq \leftarrow f_i$ // maintain current frequency
    else if $current\_traffic\_rate < T_{(i-1)l}$ **then**
        $op\_freq \leftarrow f_{i-1}$ // go to lower frequency
    else if $current\_traffic\_rate > T_{ih}$ **then**
        $op\_freq \leftarrow f_{i+1}$ // go to higher frequency
    end if
else if $(i == 1)$ **then**
    if $current\_traffic\_rate > T_{ih}$ **then**
        $op\_freq \leftarrow f_{i+1}$ // go to higher frequency
    end if
else if $(i == N)$ **then**
    if $current\_traffic\_rate < T_{(i-1)l}$ **then**
        $op\_freq \leftarrow f_{i-1}$ // go to lower frequency
    end if
end if
$wait for sampling\_period$
// update current_traffic_rate every sampling period

4.1.3 Packet Loss Aware Policy (PLAP)

For both the STP and the DTP, the trade-off between power consumption and performance is mainly affected by the fixed preset thresholds and the period of sampling the current traffic rate. The DTP can outperform the STP for the extremely frequent and abrupt traffic fluctuation scenarios at the cost of potentially less power savings, but it cannot properly cope with traffic burst scenarios. When a traffic burst occurs, the level-by-level frequency switching in the DTP may not be able to switch to the most appropriate frequency in time. During the switching, the FAR may suffer severe network delay and packet loss.

To eliminate these deficiencies associated with thresholds based policies, the Packet Loss Aware Policy (PLAP) is introduced. The PLAP is built on the DTP and the operating frequency directly jumps to the highest operating frequency 125 MHz when packet loss is detected. Algorithm 3 presents the pseudo-code of the self-adaptive local frequency control policy PLAP.

4.2 Global Frequency Control Policies

Apart from implementing frequency control policy in the NetFPGA hardware, the frequency control policy can be also implemented in software. In the Frequency Adaptive Router, a register 0x2000218 is added into the register system to control the operating frequency of the core logic FPGA of the Frequency Adaptive Router. This register provides a standard interface between hardware and software, which allows network wide global green traffic engineering to control the operating frequency of the Frequency Adaptive Router.

Algorithm 3: Packet Loss Aware Policy (PLAP)

Input: $current\_traffic\_rate, current\_frequency\_index, byte\_counter\_dropped$
Output: op_freq // operating frequency

N = 5 // number of working frequencies, default to 5
sampling_period = 10 ms // sampling period, default to 10 ms
$op\_freq = 125$ // current working frequency, initialize to 125 MHz
$f_n = \{7.8, 15.6, 31.3, 62.5, 125\}$ // set of working frequencies in MHz
$T_{nl} = \{0.368, 0.746, 1.457, 2.938, 5.862\}$
// lower thresholds for each f_i in Gb/s
$T_{nh} = \{0.421, 0.834, 1.628, 3.284, 6.551\}$
// higher thresholds for each f_i in Gb/s
$i \leftarrow current\_frequency\_index$
if $(byte\_counter\_dropped > 0)$ **then**
    $op\_freq \leftarrow f_N$// if packet drop detected, go to maximum frequency
else
    if $(2 \leq i \leq N-1)$ **then**
        if $T_{(i-1)l} \leq current\_traffic\_rate \leq T_{ih}$ **then**
            $op\_freq \leftarrow f_i$ // maintain current frequency
        else if $current\_traffic\_rate < T_{(i-1)l}$ **then**
            $op\_freq \leftarrow f_{i-1}$ // go to lower frequency
        else if $current\_traffic\_rate > T_{ih}$ **then**
            $op\_freq \leftarrow f_{i+1}$ // go to higher frequency
        end if
    else if $(i == 1)$ **then**
        if $current\_traffic\_rate > T_{ih}$ **then**
            $op\_freq \leftarrow f_{i+1}$ // go to higher frequency
        end if
    else if $(i == N)$ **then**
        if $current\_traffic\_rate < T_{(i-1)l}$ **then**
            $op\_freq \leftarrow f_{i-1}$ // go to lower frequency
        end if
    end if
end if
$wait for sampling\_period$
// update current_traffic_rate every sampling period

4.2.1 Performance First Policy (PFP)

The Performance First Policy statically sets the operating frequency of the Frequency Adaptive Router at the highest frequency 125 MHz all the time. This policy is to achieve the maximum routing capacity out of the Frequency Adaptive Router by setting the operating frequency to the maximum level and staying at this level invariably. The PFP does not attempt to provide any power saving features by default and it is suitable for constant heavy traffic load during rush hours. This policy is also the default frequency control policy in the original NetFPGA Reference Router. This static frequency control policy can be used in conjunction with a network wide global green traffic engineering. It is not self-adaptive and cannot switch to a lower operating frequency in response to the traffic load by itself without receiving frequency switching instruction from a control unit. Algorithm 4 presents the pseudo-code of the PFP.

Algorithm 4: Performance First Policy (PFP)

Input: PFP
Output: $operating\_frequency$

```
1 operating_frequency ← 125MHz
  // setting the operating frequency at 125 MHz constantly
```

4.2.2 Power Saving First Policy (PSFP)

On the contrary, the Power Saving First Policy statically sets the operating frequency of the Frequency Adaptive Router at the lowest frequency 7.8 MHz all the time. This policy is to keep the Frequency Adaptive Router running at the lowest operating frequency to obtain maximum power savings at the cost of the minimum routing capacity. The PSFP should be used when the router constantly works at expected low traffic load. Otherwise, any unexpected network traffic burst can degrade the performance because this policy will never scale up the

frequency in response to increased traffic load by itself without receiving frequency switching instruction, which may lead to significant network delay and packet loss. Thus, this policy should be used in conjunction with a network wide global green traffic engineering. Algorithm 5 presents the pseudo-code of the PSFP.

Algorithm 5: Power Saving First Policy (PSFP)

Input: $PSFP$
Output: $operating\_frequency$

```
1 operating_frequency ← 7.8MHz
  // setting the operating frequency at 7.8 MHz constantly
```

4.2.3 User Defined Policy (UDP)

The User Defined Policy can adaptively control the operating frequency by automatically reading and writing the new memory-mapped I/O register introduced into the Frequency Adaptive Router, in response to the instantaneous traffic load. It is the most customizable frequency adaptation control mechanism and provides the flexibility for the users to design and experiment their own policies. If a frequency control policy is well customized by an user, the frequency control policy could be the best solution to balance the trade-off between power consumption and performance under certain circumstances. For example, in a SDN network, the routing capacity of the Frequency Adaptive Router can be controlled by OpenFlow switch in conjunction with the network wide global green traffic engineering and the mechanism of Ethernet port shutdown. It is possible to achieve more power savings than the self-adaptive local frequency control policies. Table 4.4 compares the six different frequency control policies and summarizes the main advantages and disadvantages of each frequency control policy.

Table 4.4: Advantages and disadvantages for each frequency control policy

Policy	Advantages	Disadvantages
STP	self-adaptive and dynamic hardware switching	poor QoS when traffic fluctuates around thresholds
DTP	self-adaptive and dynamic hardware switching	poor QoS when traffic burst
PLAP	self-adaptive and dynamic hardware switching	packet loss when jumping to 125 MHz
PFP	maximum routing capacity	slow software switching
PSFP	maximum power saving	slow software switching
UDP	flexibility for global control	slow software switching

4.3 System Models

4.3.1 System Model of Reference Router

The NetFPGA Reference Router can be modelled as an M/M/1/N queuing system to analyze the system performance of the Reference Router. As shown in Figure 4.5, the system model of the Reference Router consists of a finite FIFO queue of buffer size N with packets arriving randomly, and a server that retrieves packets from the FIFO queue at a specified service rate [114]. The packet arrival process is a Poisson process with exponentially distributed random inter-arrival times with an average arrival rate λ. The packet service time is an exponentially distributed random variable with an average service rate μ. The packet arrival process and the packet service process are independent of each other.

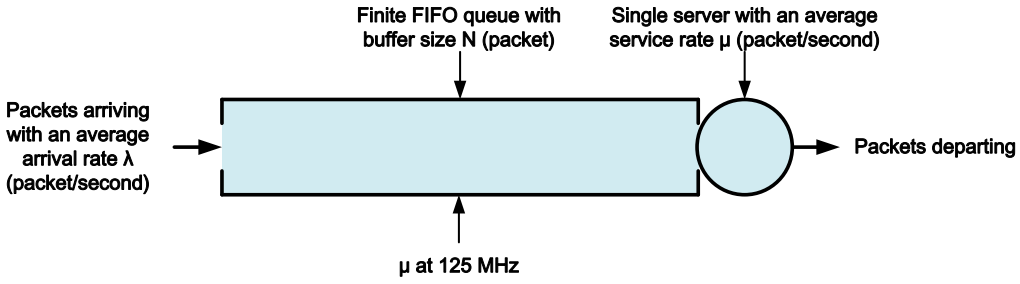

Figure 4.5: System model of the Reference Router

In the M/M/1/N queuing model [115], the state of queuing system is represented by the number of packets currently in the system as shown in Figure 4.6. The packets arrive and depart only one packet at a time. Thus, the system state can change only by one unit at a time. If currently there are n customers in the system and a packet arrives to the system, the system state increases from n to n+1 with an average arrival rate λ. If currently there are n customers in the system and a packet departs out the system, the system state decreases from n to n-1 with an average service rate μ. The average packet service rate must exceeds the average packet arrival rate, otherwise the unprocessed packets could continue to infinitely grow in the finite FIFO queue and the finite queue will eventually overflow. The finite FIFO buffer with buffer size N can only accommodate N-1 packets and the packet N will be dropped due to queue overflow.

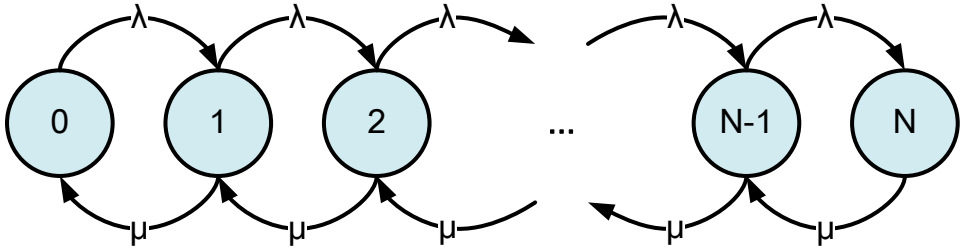

Figure 4.6: M/M/1/N queuing model

The system state moves from n-1 to n at the rate of $\lambda \times P_{n-1}$ and the the system state moves from n to n-1 at the rate of $\mu \times P_n$. The probability of having one packet in the queuing system is shown in equation (4.1). ρ is the average utilisation factor for the queueing system. Similarly, the probability of having N packets in the queueing system is shown in equation (4.2).

$$P_1 = \left(\frac{\lambda}{\mu}\right) \times P_0 = \rho \times P_0 \tag{4.1}$$

$$P_n = \rho^n \times P_0 \tag{4.2}$$

When the M/M/1/N queue is in equilibrium, the steady state equations are shown in equation (4.3) [116]. Table 4.5 summarizes the notations used for the M/M/1/N queuing model for analyzing the performance of the system model of the Reference Router.

$$\begin{cases} \lambda \times P_0 = \mu \times P_1 \\ \lambda \times P_{n-1} + \mu \times P_{n+1} = \lambda \times P_n + \mu \times P_n \quad n = 1, 2, ...N-1 \end{cases} \tag{4.3}$$

Table 4.5: Notations used for M/M/1/N queuing model

notation	description
n	number of packets in the queuing system
λ	average packet arrival rate
λ_e	average effective packet arrival rate
μ	average packet service rate
ρ	average utilisation factor
P_0	probability of having no packet in the queuing system
P_N	probability of queue overflow
L_s	average number of packets in the queuing system
L_q	average number of packets in the FIFO queue
W_s	average waiting time in the queuing system
W_q	average waiting time in the FIFO queue

As the sum of the probabilities of the system state is 1 as shown in equation (4.4), the probability of having no packet in the queuing system is shown in equation (4.6). If the number of packets in the queuing system equals N, a new arriving packet will be dropped due to queue overflow. Thus, the probability of queue overflow is shown in equation (4.7).

$$\sum_{n=0}^{N} P_n = P_0 + P_1 + P_2 + ... + P_{N-1} + P_N = 1 \tag{4.4}$$

$$\sum_{n=0}^{N} P_n = P_0 + P_0 \times \rho + P_0 \times \rho^2 + ... + P_0 \times \rho^{n-1} + P_0 \times \rho^n = \sum_{n=0}^{N} P_0 \times \rho^n = 1 \tag{4.5}$$

$$P_0 = \frac{1-\rho}{1-\rho^{N+1}} \tag{4.6}$$

$$P_N = \frac{(1-\rho)\rho^N}{1-\rho^{N+1}} \tag{4.7}$$

The average number of packets in the queuing system L_s can be determined using equation (4.8) and (4.9) [115]. The average number of packets in the FIFO queue L_q can be determined using equation (4.10) and (4.11) [115]. Figure 4.7 compares the difference between the average number of packets in the queuing system and that in the FIFO queue. For example, in Figure 4.7, the number of packets in the FIFO queue is 4, while the number of packets in the queuing system is 5.

$$L_s = \sum_{n=0}^{N} n \times P_n \tag{4.8}$$

$$L_s = \frac{\rho - (N+1)\rho^{N+1} + N\rho^{N+2}}{(1-\rho)(1-\rho^{N+1})} \tag{4.9}$$

$$L_q = \sum_{n=1}^{N} (n-1) \times P_n \tag{4.10}$$

$$L_q = \frac{\rho^2 - N\rho^{N+1} + (N-1)\rho^{N+2}}{(1-\rho)(1-\rho^{N+1})} \tag{4.11}$$

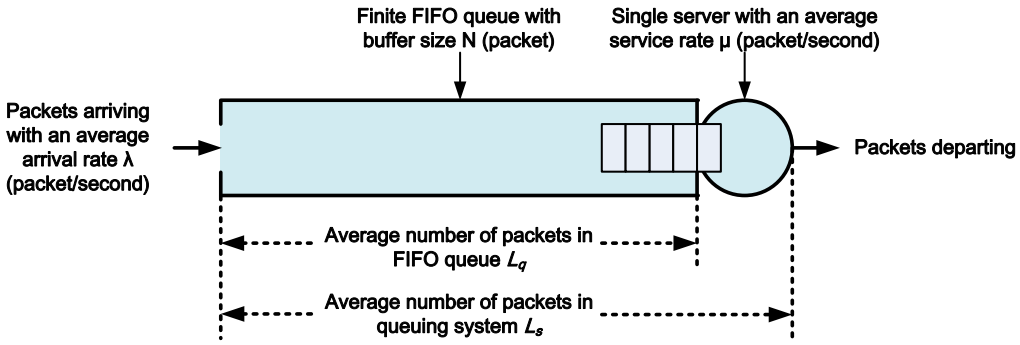

Figure 4.7: Difference between the queuing system and the FIFO queue

Since the probability of queue overflow P_N is acquired in equation (4.7), the λP_N represents the average packets dropped due to queue overflow. Thus, the effective packets arrival rate defined as λ_e can be calculated with equation (4.12). The average waiting time in the queuing system W_s can be calculated with equation (4.13) and the average waiting time in the FIFO queue W_q can be calculated with equation (4.14). Table 4.6 summarizes the performance characteristics in system model of Reference Router.

$$\lambda_e = \lambda - \lambda P_N \tag{4.12}$$

$$W_s = \frac{L_s}{\lambda_e} \tag{4.13}$$

$$W_q = \frac{L_q}{\lambda_e} \tag{4.14}$$

Table 4.6: Performance characteristics in system model of Reference Router

characteristic	expression
ρ	$\frac{\lambda}{\mu}$
P_0	$\frac{1-\rho}{1-\rho^{N+1}}$
P_N	$\frac{(1-\rho)\rho^N}{1-\rho^{N+1}}$
L_s	$\frac{\rho-(N+1)\rho^{N+1}+N\rho^{N+2}}{(1-\rho)(1-\rho^{N+1})}$
L_q	$\frac{\rho^2-N\rho^{N+1}+(N-1)\rho^{N+2}}{(1-\rho)(1-\rho^{N+1})}$
W_s	$\frac{L_s}{\lambda_e}$
W_q	$\frac{L_q}{\lambda_e}$

4.3.2 System Model of Frequency Adaptive Router

The System Model of the Frequency Adaptive Router is built based on the system model of the Reference Router. It is basically the same as the system

model of the Reference Router except for the average packet service rate as shown in Figure 4.8.

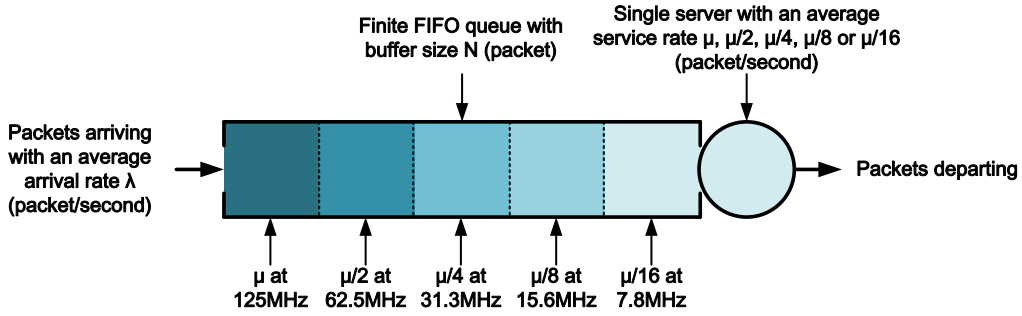

Figure 4.8: System model of the Frequency Adaptive Router

Since there are five different operating frequencies provided in the Frequency Adaptive Router, the average packet service rate under different operating frequencies is different corresponding to the actual routing capacity of the FAR in Table 4.2. The average packet service rate under five different operating frequencies in the Frequency Adaptive Router is $\frac{\mu}{2^K}$ (K = 0, 1, 2, 3 and 4 corresponding to 125 MHz, 62.5 MHz, 31.3 MHz, 15.6 MHz and 7.8 MHz respectively). Table 4.7 summarizes the performance characteristics in the system model of the Frequency Adaptive Router.

Table 4.7: Performance characteristics in system model of Frequency Adaptive Router

characteristic	expression
ρ	$\frac{2^K\lambda}{\mu}$
P_0	$\frac{1-\rho}{1-\rho^{N+1}}$
P_N	$\frac{(1-\rho)\rho^N}{1-\rho^{N+1}}$
L_s	$\frac{\rho-(N+1)\rho^{N+1}+N\rho^{N+2}}{(1-\rho)(1-\rho^{N+1})}$
L_q	$\frac{\rho^2-N\rho^{N+1}+(N-1)\rho^{N+2}}{(1-\rho)(1-\rho^{N+1})}$
W_s	$\frac{L_s}{\lambda_e}$
W_q	$\frac{L_q}{\lambda_e}$

4.4 Summary

This chapter describes six different frequency control policies in detail. These frequency control policies are designed to cover a variety of application scenarios for the Frequency Adaptive Router. Experiments in Chapter 5 will evaluate these frequency control policies. The two system models of the Reference Router and the Frequency Adaptive Router are established using the M/M/1/N queuing system, which can be used to analyze the system performance of the Reference Router and the Frequency Adaptive Router.

Chapter 5

Energy Efficiency and Performance Evaluation of Frequency Adaptive Router

Experiments are conducted to evaluate the energy efficiency and the performance of the Frequency Adaptive Router in comparison with the Reference Router. Each experiment has a unique behavior, purpose and suitability to investigate and explore the latent relationship among operating frequency, performance and power consumption. The interaction and interdependency among these three individual factors then can be interpreted and explained.

To investigate the energy efficiency of the Frequency Adaptive Router, quiescent power consumption and total power consumption of the Reference Router and the Frequency Adaptive Router are measured for comparisons. To examine the performance of the Frequency Adaptive Router, four network performance measures are considered including peak measured throughput, round trip time (RTT), packet loss rate and frequency adaptation transition time. Performance comparisons between the Reference Router and the Frequency

Adaptive Router are made to analyze the trade-off between the energy efficiency and the performance.

Section 5.1 describes each evaluation metric in detail. Section 5.2 is experimental setup for the energy efficiency and the performance evaluation. While, Section 5.3, 5.4 and 5.5 evaluate the energy efficiency and the performance of the Reference Router and the Frequency Adaptive Router under different scenarios, including fixed traffic experiments in Section 5.3, experiments with synthetic traces in Section 5.4 and integration with other green techniques in Section 5.5. Section 5.6 summarizes the Chapter.

5.1 Evaluation Metrics

To evaluate the energy efficiency and the performance of the FAR in comparison with the RR, six evaluation metrics are investigated and examined. In the energy efficiency category, 1) **quiescent power consumption** and 2) **total power consumption** are measured. While, in the performance category, the four evaluated metrics are: 3) **peak measured throughput**, 4) **round trip time (RTT)**, 5) **packet loss rate** and 6) **frequency adaptation transition time**. Details about each evaluation metric are explained below.

5.1.1 Energy Efficiency Metrics

Energy efficiency metrics are measured in terms of power consumption. Power consumption in a network device is the sum of two components: quiescent power and dynamic power [117] as given in equation (5.1):

$$P_t = P_q + P_d \tag{5.1}$$

where P_t is the total power, P_q is the quiescent power, P_d is the dynamic power. Quiescent power is the static power drawn by the device when it is powered up, configured with user logic but without processing traffic load. Dynamic power is the power consumed during processing traffic load in the core or I/O of the device and it is, therefore, operating frequency dependent.

In CMOS technology, the quiescent power is caused by the leakage power in transistors when the transistors are all placed in *off* state. The leakage power is the power consumed by unintended leakage, constantly consuming power. The state of a transistor (*on* or *off*) is controlled by the gate voltage on the gate terminal. If this gate voltage is above the threshold voltage (V_T), the transistor is placed in the *on* state allowing on-current to flow from the source (V_{dd}) to the drain (ground (GND)). On the contrary, if this gate voltage is below the threshold voltage (V_T), the transistor is placed in the *off* state and theoretically no off-current should flow from the source to the drain. But in reality, the transistors leak off-current between the source and the drain even if the gate voltage is below the threshold voltage. The quiescent power can be calculated with equation (5.2)[118]:

$$P_q = V \times I_{leak} \tag{5.2}$$

where P_q is the quiescent power, V is the operating voltage and I_{leak} is the leakage current. The quiescent power can be also affected by the technology node, the doping level and the gate oxide thickness in integrated circuits. The technology node refers to the feature size of an integrated circuit [119, 120]. Static CMOS gates are used to be very energy efficient and nearly dissipate no power consumption when placed in *off* state. However, when the CMOS technology entered deep-submicron territory in feature size (<180 nm), the leakage power has increased significantly and accounts for approximately from 20% to 40% in

total power consumption [121]. Increasing the doping level or decreasing the gate oxide thickness allow higher on-current for faster transitions but cause higher leakage off-current.

Compared to the the quiescent power, the dynamic power contributes to integrated circuits functions and user applications. The dynamic power of a network device can be modeled as a linear function of the traffic characteristics as given in equation (5.3):

$$P_d = ACfV^2 \tag{5.3}$$

where A is the activity factor represents the fraction of the circuit that is switching, C is the equivalent load capacitance, f is the core operating frequency, and V is the operating voltage. For a given NetFPGA board, C and V are fixed values and A is affected by the number of active ports, the instantaneous traffic bit rate and the forwarding packet size.

The power consumption equation (5.1) is the basis for theoretical analysis. In operation, the NetFPGA 1G board is a PCI based board and all the power supplies are derived from the 3.3 V and the 5 V power rails on the PCI bus. The embedded FPGA chips draw power from the 3.3 V rail of the PCI bus, and the PHY components are supplied by the 5 V rail. The 3.3 V and 5 V pins of the PCI bus extender are used to measure the overall current drawn by the 3.3 V and 5 V powered components on the NetFPGA board mounted on the PCI bus extender. Hence, the total power consumption can be calculated with equation (5.4):

$$P_t = I_{3.3} * 3.3 + I_5 * 5 \tag{5.4}$$

where P_t is the total power, $I_{3.3}$ (measured in amperes) is the total current of board components drawn power from the 3.3 V rail of the PCI bus and I_5 (measured in amperes) is the total current of 5 V powered components.

5.1.2 Performance Metrics

The peak measured throughput is the peak measured rate of successful data delivery over a real implemented system. In the performance evaluation, it is the most important metric to be measured because it directly determines the preset thresholds in the frequency adaptation control policies. Each operating frequency has an associated traffic throughput threshold beyond which the router will begin to lose a significant number of packets. The peak measured throughput must be measured as accurate as possible. Otherwise, the difference between the input traffic load and the inaccurately preset thresholds could degrade the performance of the FAR.

The round trip time (RTT) is also an important network performance metric in the performance evaluation. It is the propagation time for a packet to travel from a specific source to a specific destination and return back to the source. Measuring the RTT assists network operators and end users in understanding the network performance and taking measures to improve the QoS if needed.

To ensure the required QoS, a router should speedily and accurately forward data packets with no or little packet loss. Packet loss rate is another important metric adopted in the performance evaluation in the experiments. In the NetFPGA user data path pipeline, packet loss occurs in four MAC receive queues and four MAC output queues, but not in any MAC transmit queues. This is because the transmit queues simply take packets from the corresponding output queues and

send the packets out through the Ethernet ports. Thus, the total packet loss can be calculated with following equation:

$$L_t = L_r + L_o \tag{5.5}$$

where L_t is the total packet loss, L_r is the total packet loss in the four receive queues, and L_o is the total packet loss in the four output queues.

Frequency adaptation transition time is closely associated with packet loss. It is the time required for the FAR to switch from one operating frequency to another. Experimental results show that, under constant heavy traffic load, longer transition time could result in greater packet loss, especially during switching from a lower operating frequency to a higher one. Thus, frequency adaptation transition time is another important metric in the performance evaluation.

5.2 Experimental Setup

The lab setup is shown in Figure 5.1. The implemented lab setup consists of: 1) two NetFPGA 1G boards, 2) three personal computers (PCs), 3) one Ultraview PCI bus extender PCIEXT-64U [122], and 4) one National Instruments (NI) data acquisition (DAQ) USB-6251 [123] for power consumption measurements.

In the experimental setup, one NetFPGA board is configured as a packet generator (PG) [124], while the other one is configured as the RR or the FAR in turn for power consumption and performance measurements. The two experimental NetFPGA boards are both in version 2.1 revision 3. Each of the NetFPGA board is hosted by a separate PC installed with CentOS version 5.5. The host PC hardware is a Dell Optiplex 780 equipped with an Intel Core 2 Duo

processor model E7500 running at 2.93 GHz and a Samsung 4GB DDR3 1333 MHz memory. Another host PC installed with Windows 7 is used for the LabVIEW software, which collects the power consumption data from the NI USB DAQ. In order to accurately collect power consumption data, there were no other applications running on either of these two NetFPGA host PCs.

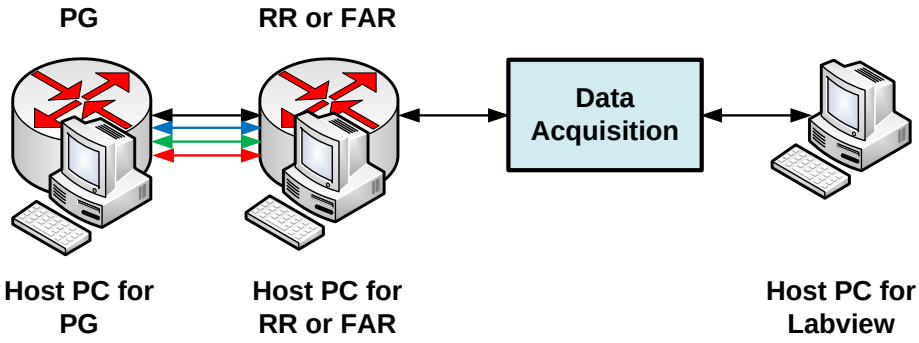

Figure 5.1: Experimental setup

5.2.1 Power Consumption Measurements

Power consumption measurements of the PCI-based NetFPGA 1G board can be divided into coarse-grained power consumption measurements and fine-grained power consumption measurements. The coarse-grained power consumption measurements take the power measurements of the global system, including the entire host PC and the NetFPGA board, through the main power supply cable depicted by point 1 in Figure 5.2 with a power consumption measurement device such as the OWL+USB wireless electricity monitor [125]. And then the power consumption of the NetFPGA board can be estimated by measurement comparison of the system without the NetFPGA board running. On the other hand, the fine-grained power consumption measurements are made through a dedicated PCI bus extender [122], as shown by point 2 in Figure 5.2.

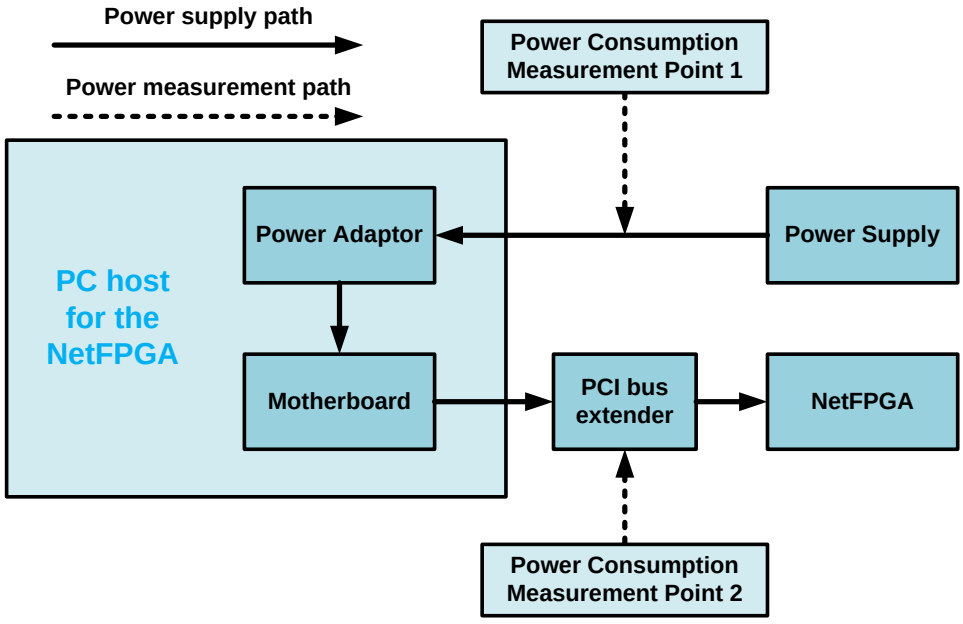

Figure 5.2: Coarse-grained and fine-grained power consumption measurement methods

Coarse-grained power consumption measurements

Some initial power consumption measurements were taken with this OWL energy meter to estimate the power consumption of the NetFPGA. These measurements show that the instantaneous power consumption of the global system fluctuates among four fixed values (32 W, 48 W, 64 W and 128 W). These big power consumption fluctuations are most likely due to different working status of the power-hungry components in the host PC, such as digital versatile disk (DVD) drives and fans. Since different hardware components in the host PC can affect the instantaneous power consumption of the host PC, the accuracy of the power consumption measurements of the NetFPGA board cannot be guaranteed. Thus, the results of the coarse-grained power consumption measurements of the NetFPGA board are not reported in the thesis.

Fine-grained power consumption measurements

To isolate the power consumption of the NetFPGA board, fine-grained power consumption measurement tools are used as shown in Figure 5.3. The fine-grained power consumption measurement tools consist of an Ultraview PCI bus extender PCIEXT-64U, a National Instruments (NI) data acquisition (DAQ) device USB-6251, and a LabVIEW software. The high speed PCI bus extender card is plugged into the motherboard slot of the host PC. This smart bus extender has a 5 V and 3.3 V signalling environment which provides a way to precisely measure the real-time power consumption of the PCI-based board while the board is working [126]. The power consumption of the NetFPGA board can be measured by connecting the voltage supply pins 5 V and 3.3 V on the bus extender to the corresponding pins on the NI USB DAQ device.

The NI USB DAQ is a USB based high-speed data acquisition device optimized for superior accuracy at fast sampling rates, which is compatible with LabVIEW 2011 (version 11.0, 32-bit) for measuring and capturing power consumption data. It provides a link between the input/output signals of the DAQ and the LabVIEW software running on a host PC. Power consumption data outputs are collected by the DAQ coupled with its software script on the host desktop computer. The outputs of the 3.3 V and 5 V voltage supply pins have a one-to-one voltage-to-current correspondence [122]. Thus, the power consumption of the NetFPGA can be calculated. This is an accurate and fine-grained measurement method used in the thesis for power consumption measurements. It provides much more accurate power consumption measurement data of the NetFPGA board than the coarse-grained power consumption measurements with OWL meter.

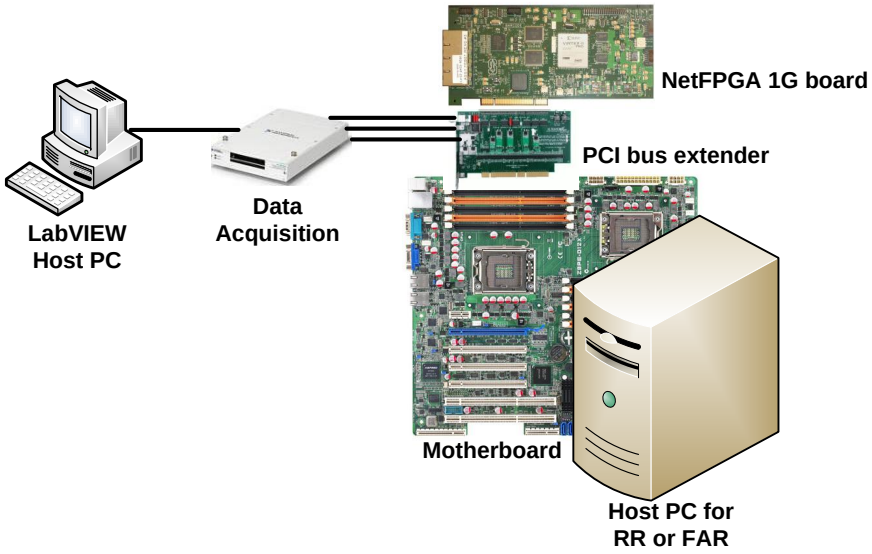

Figure 5.3: Fine-grained power consumption measurement tools

5.2.2 Performance Measurements

The NetFPGA board provides four external network interfaces to the host machine operating system: nf2c0, nf2c1, nf2c2 and nf2c3. The NetFPGA configured as the packet generator is directly connected to the host machine motherboard through a PCI slot. While, the NetFPGA configured as the Reference Router or the Frequency Adaptive Router are connected to the PCI slot of the PCI bus extender, which is mounted on the PCI slot of the host machine motherboard. To set up the packet routing environment, a user-level software called SCONE is used to perform IPv4 forwarding, handle ARPs and various ICMP messages [127]. SCONE has telnet (port 23) and HTTP (port 8080) services to handle router control and implements a subset of Pee Wee OSPF (PW OSPF). It can also configure the NetFPGA with the MAC and IP addresses of the four interfaces, and the routing and ARP tables onto the NetFPGA, which hardware accelerates the forwarding path.

The software SCONE can communicate with the NetFPGA hardware using its corresponding kernel driver. Performance evaluation metrics such as packet loss can be obtained from SCONE by communicating with the register system that is built in the NetFPGA hardware. From these memory-mapped I/O registers in the register system, status information such as the total number of packets received, dropped and transmitted for each queue can be acquired.

Peak measured throughput and packet loss rate evaluation

To measure the peak measured throughput and the packet loss rate, the experimental setup is shown in Figure 5.4. All four ports on the packet generator are connected to the corresponding ports on the Reference Router or the Frequency Adaptive Router under evaluation. The packet generator is able to send user specified traffic at any required bit rate up to 4 Gb/s. Network traffic with different bit rates and packet sizes are generated by the packet generator and sent from its four Ethernet ports to the corresponding ports on the Reference Router or the Frequency Adaptive Router. Then, the traffic is routed through the four Ethernet ports on the Reference Router or the Frequency Adaptive Router with different operating frequencies, and sent back to the corresponding ports on the packet generator.

To measure the peak measured throughput, the traffic bit rate on the packet generator is configured to increase linearly using typical sized packets (64 bytes, 576 bytes and 1500 bytes). When packet loss is detected in the SCONE, the peak measured throughput is recorded. While, packet loss is measured through the packet generator output. The output can indicate the total number of packets transmitted (*no_packets_transmitted*) and received (*no_packets_received*) on the

packet generator. The packet loss is calculated as the difference between *no_packets_transmitted* and *no_packets_received*.

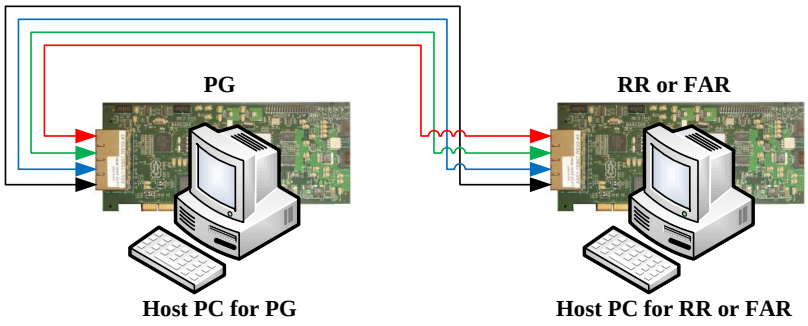

Figure 5.4: Experimental setup for peak measured throughput and packet loss rate evaluation

Round trip time evaluation

To evaluate the round trip time (RTT), the experimental setup is slightly different from the setup in Figure 5.4. As shown in Figure 5.5, all four ports on the packet generator are used to measure the round trip time: two ports of the packet generator are connected directly to one another and the RTT is measured to provide a baseline RTT reference as RTT_0, the other two ports of the packet generator are connected to the Reference Router or the Frequency Adaptive Router under test and the RTT is measured as RTT_1. The RTT is calculated as the difference between RTT_1 and RTT_0.

Frequency adaptation transition time evaluation

The NetFPGA board draws power from both the 3.3 V and the 5 V power rails of the PCI bus on the host PC through the PCI bus extender. Since the core logic FPGA draws power from the 3.3 V rail, scaling the operating frequency of the core

logic FPGA can affect the 3.3 V current. Thus, the frequency adaptation transition time can be calculated from captured current waveform of the 3.3 V pin of the PCI bus extender.

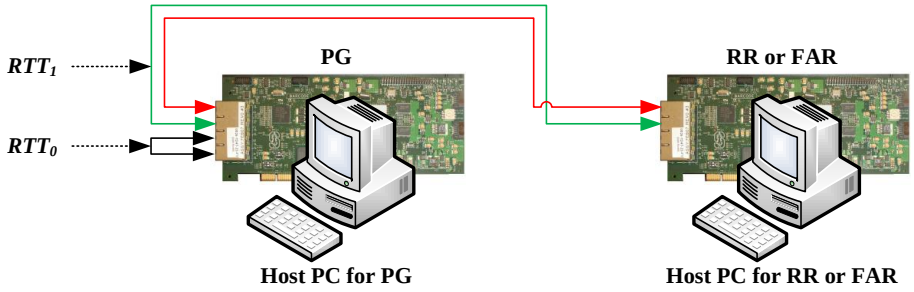

Figure 5.5: Experimental setup for round trip time evaluation

5.3 Fixed traffic experiments

The total power consumption of a router is a function of the operating frequency, the number of active ports, the traffic bit rate and the packet size. To evaluate the impact of each of these factors on the performance and the power consumption of the Reference Router and the Frequency Adaptive Router, a series of carefully designed experiments are conducted.

5.3.1 Quiescent power consumption

The experiments start with measuring the quiescent power consumption of the Reference Router (RR) and the Frequency Adaptive Router (FAR). The quiescent power consumption is the power consumption of the RR and the FAR in idle state when the NetFPGA is configured as the RR or the FAR but without routing any traffic. To evaluate the impact of the operating frequency and the number of active ports on the power consumption of the NetFPGA, the RR bitfile and the FAR bitfile

are downloaded into the core logic FPGA to configure the NetFPGA as a RR and a FAR in turn for comparison. The four Ethernet ports are activated one by one, and the power consumption of the RR and the FAR are measured at their supported operating frequencies. The measurement results in Table 5.1 and 5.2 show that the power consumption of the RR and the FAR is proportional to the number of active ports, with approximately 1 W additional power consumption for each activated port.

Table 5.1: Impact of operating frequency and number of active ports on power consumption of RR

Active ports No.	RR Power Consumption (W)	
	125MHz	62.5MHz
0	6.994	5.333
1	7.841	6.325
2	8.767	7.302
3	9.760	8.373
4	10.725	9.441

Table 5.2: Impact of operating frequency and number of active ports on power consumption of FAR

Active ports No.	FAR Power Consumption (W)				
	125MHz	62.5MHz	31.3MHz	15.6MHz	7.8MHz
0	7.394	6.245	5.592	5.253	5.118
1	8.313	7.092	6.531	6.246	6.109
2	9.149	8.055	7.508	7.250	7.124
3	10.145	9.067	8.564	8.330	8.188
4	11.121	10.117	9.638	9.373	9.237

Figure 5.6 compares the quiescent power consumption of the RR and the FAR under different operating frequencies (125 MHz and 62.5 MHz) and different number of active ports. Experimental results indicate that under the same operating frequency of 125 MHz or 62.5 MHz, the FAR consumes more quiescent power than the RR. This is due to the fact that an additional frequency division module module and an additional AFIFO module are integrated into the FAR. Each additional module incurs additional power consumption. Moreover, for the

RR, the two SRAMs work synchronously with the core logic FPGA for reading and writing data at 125 MHz or 62.5 MHz. For the FAR, the core logic FPGA can be scaled among five different frequencies 125 MHz, 62.5 MHz, 31.3 MHz, 15.6 MHz and 7.8 MHz, but the two SRAMs are working asynchronously with the core logic FPGA and constantly running at 125 MHz. This means the power consumption difference between the RR and the FAR at 62.5 MHz is more than that at 125 MHz, because at 62.5MHz, the two SRAMs run at 62.5 MHz in the RR and run at 125 MHz in the FAR, although the core logic FPGA processor is running at the same 62.5 MHz. For example, with 4 active ports at 125 MHz, the quiescent power consumption of the RR is 10.725 W, while the quiescent power consumption of the FAR is 11.121 W, and the FAR consumes approximately 3.5% more quiescent power than the RR. However, with 4 active ports at 62.5 MHz, the quiescent power consumption of the RR is 9.441 W, while the quiescent power consumption of the FAR is 10.117 W, and the FAR consumes approximately 6.7% more quiescent power than the RR.

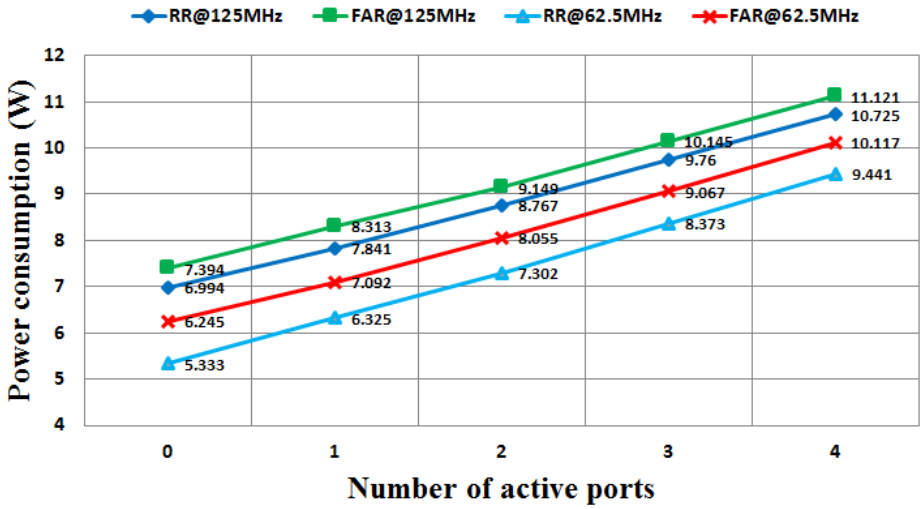

Figure 5.6: Quiescent power consumption comparison between RR and FAR under different operating frequencies (125 MHz and 62.5 MHz) and different number of active ports (from 0 to 4)

5.3.2 Power consumption under fixed traffic

The traffic characteristics (traffic bit rate and packet size) could also affect the power consumption of the Reference Router and the Frequency Adaptive router. As indicated in [128], packets of 64 bytes, 576 bytes and 1500 bytes are typical in real network links and the packet size profile peaks at 64 bytes and 1500 bytes. The minimum size of a standard Ethernet packet is 64 bytes. 1500 bytes is the maximum transmission unit (MTU) for Ethernet. MTU is an important factor for network throughput and should be as large as possible, because larger MTU introduces less overhead for payload transmission. In multi-network environments, if a maximum sized packet travels from a network with a larger MTU to a smaller MTU network, the packet will have to be fragmented to smaller sized packets. 576 bytes is the default IP maximum datagram size. It is also the default conservative packet size that all IP routers should support.

To evaluate the impact of traffic characteristics on router power consumption, a series of experiments were performed using different operating frequencies (125 MHz, 62.5 MHz, 31.3 MHz, 15.6 MHz and 7.8 MHz), different traffic bit rates (from 100 Mb/s to 1 Gb/s for each link) and different packet sizes (64 bytes, 576 bytes and 1500 bytes). Packet streams are sent from the four ports on the packet generator, routed through the four ports on the Reference Router or the Frequency Adaptive router, and back to the corresponding ports on the packet generator. Figure 5.7, 5.8 and 5.9 present the total power consumption of the RR and the FAR under the same 64, 576 and 1500 bytes packet stream with different operating frequencies and different aggregated input traffic bit rates. Figure 5.10 presents the power consumption of FAR under different packet sizes and different traffic bit rates at 125 MHz. Results indicate that power consumption is proportional to the traffic bit rate and the operating frequency for both the RR and the FAR. Refer to equation

(5.3), higher traffic bit rate means higher activity factor *A* which represents the average number of switching events of the transistors in the chip. Increasing the operating frequency of a router also increases its power consumption.

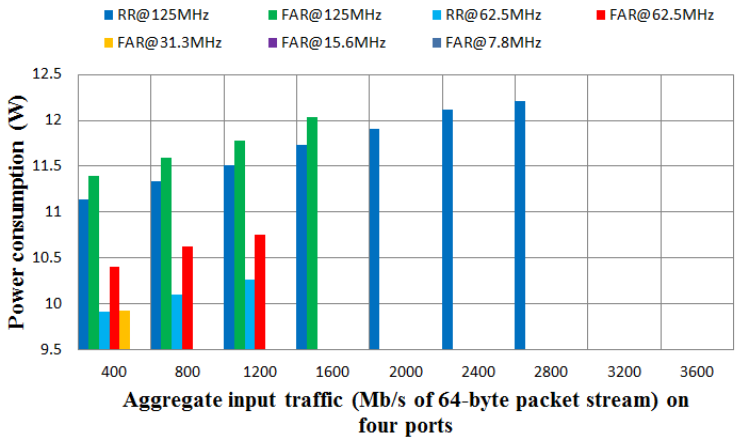

Figure 5.7: Power consumption of RR and FAR under different frequencies and traffic bit rates with 64-byte packet stream

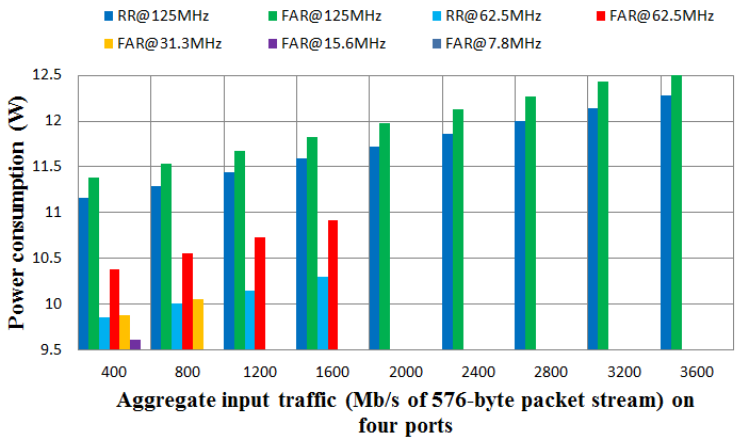

Figure 5.8: Power consumption of RR and FAR under different frequencies and traffic bit rates with 576-byte packet stream

One key point to be noted is that the power differences between the RR running at 125 MHz and 62.5 MHz come from the power consumption difference in both the core logic FPGA processor and the two SRAMs, whereas the power consumption differences in the FAR only come from the core logic FPGA

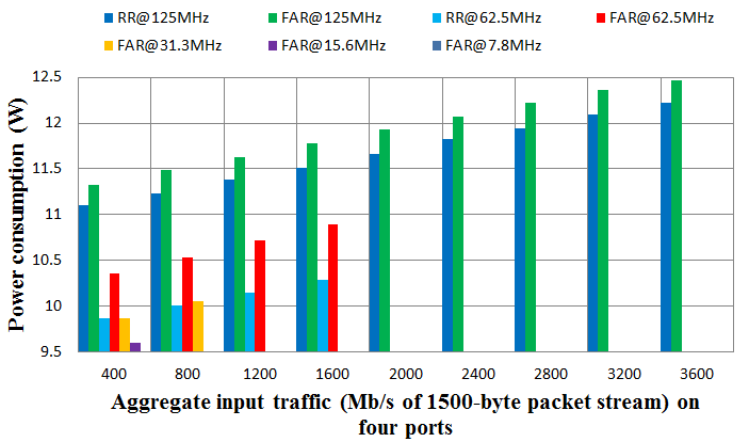

Figure 5.9: Power consumption of RR and FAR under different frequencies and traffic bit rates with 1500-byte packet stream

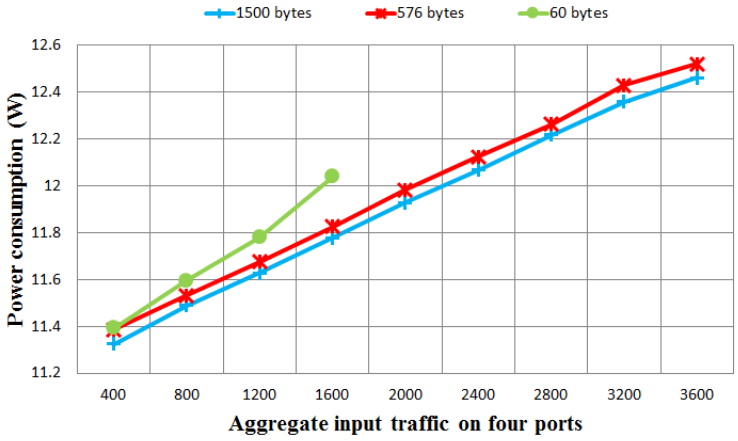

Figure 5.10: Power consumption of FAR under different packet sizes and different traffic bit rates at 125 MHz

processor because the two SRAMs are always running at 125 MHz. Figure 5.7, 5.8 and 5.9 also show that at the same working frequency of 125 MHz or 62.5 MHz, the FAR consumes more power than the RR, due to the additional AFIFO module and frequency division modules. Although at the same operating frequency, the FAR consumes around 6% more power than the RR, experiments with synthetic traces imitating real traces in the next subsection indicate that the FAR can work at lower frequencies when the traffic is low, so that it can result in overall power

savings of up to 46% in certain typical traffic pattern. For light traffic load, the FAR can save significant amount of power rather than leaving the routers always on the maximum operating frequency all the time.

5.3.3 Peak measured throughput

To evaluate the impact of the traffic characteristics and the operating frequencies on the performance of the Reference Router and the Frequency Adaptive Router, the link capacity is one of the most important metrics to be firstly measured. Link capacity is also known as the peak measured throughput. Figure 5.11 presents the link capacity of the RR and the FAR under different operating frequencies and different typical packet sizes. Experimental results show that higher operating frequency and larger packet size could lead to higher link capacity. A higher frequency produces more cycles per second to fit more data in per second and it thus provides higher routing capacity. Due to less overhead incurred by packet head processing, larger packet size typically means higher routing capacity for routers, or lower power consumption, or both. So it is advisable to use larger packet sizes whenever possible.

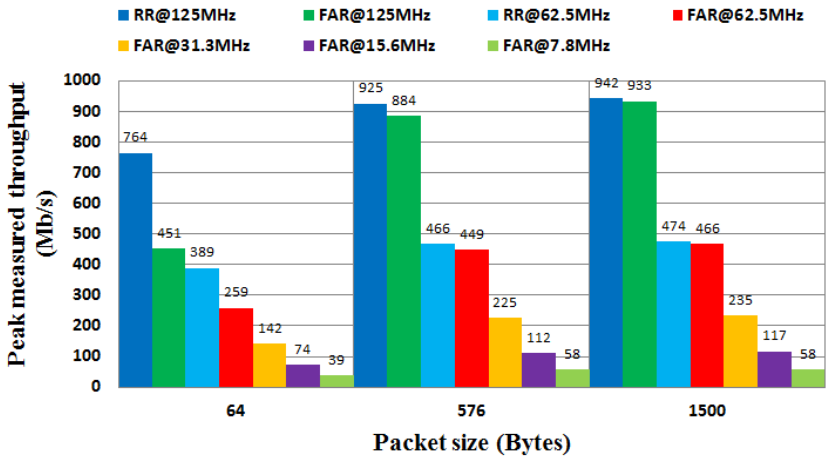

Figure 5.11: Link capacity of RR and FAR under different frequencies and packet sizes

5.3.4 Round trip time

The traffic characteristics and the operating frequencies could also affect the round trip time (RTT). The RTT is measured using the NetFPGA packet generator [124] to achieve more accurate RTT results due to the fact that timestamping in packet generator is performed in hardware. Compared with using hardware, the software RTT measurements, such as *PING*, involve the process of notifying the kernel when a packet arrives. Transferring the packet into the kernel introduces a variable length delay thus limiting the accuracy of the results. In addition, the RTT measurements are performed by passively measuring the RTT using widely deployed TCP timestamp options carried in TCP headers. Thus, the hardware RTT measurements do not need to launch out-of-band Internet Control Message Protocol (ICMP) echo requests (*PINGs*), nor to embed timing information in application traffic.

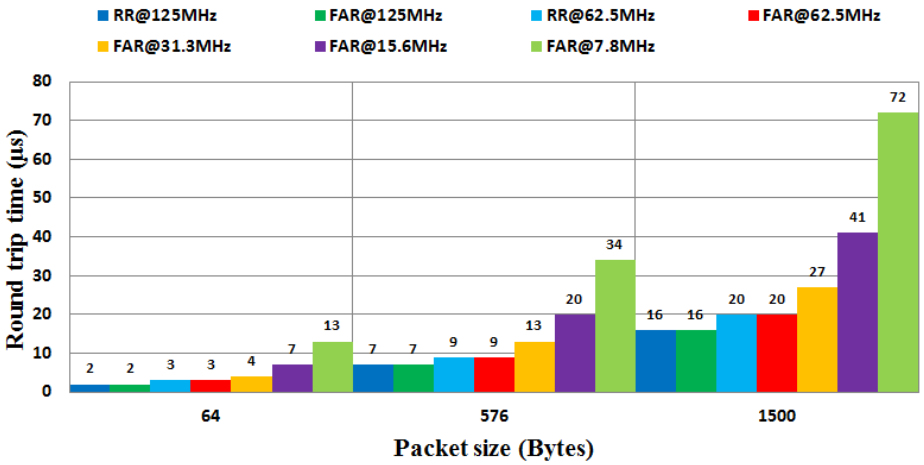

Figure 5.12: Round trip time of RR and FAR under different frequencies and packet sizes

Experimental results indicate that the FAR doesn't affect the RTT. At either 125 MHz or 62.5 MHz, the FAR has the same RTT as the RR. For example, as shown in Figure 5.12, a packet 64 bytes long has a 2 µs RTT at the same operating frequency

of 125 MHz for both the RR and the FAR. Experimental results also show that higher operating frequency and smaller packet size could give rise to a shorter round trip time. Compared with smaller packet, larger packet has longer RTT because it takes longer time to buffer and process the packet payload. For example, as shown in Figure 5.12, a packet 64 bytes long has a 2 µs RTT, while a packet 1500 bytes long has a 16 µs RTT at the same operating frequency of 125 MHz for the RR.

5.3.5 Packet loss rate

The traffic characteristics and the operating frequencies also affect the packet loss rate. Figure 5.13, 5.14 and 5.15 presents the packet loss rate of the RR and the FAR under the 64 bytes, 576 bytes and 1500 bytes packet streams with different operating frequencies and different aggregated input traffic bit rates. Experimental results indicate that, for both the RR and the FAR, the packet loss rate is proportional to the traffic bit rate. Besides, higher operating frequency could reduce packet loss rate.

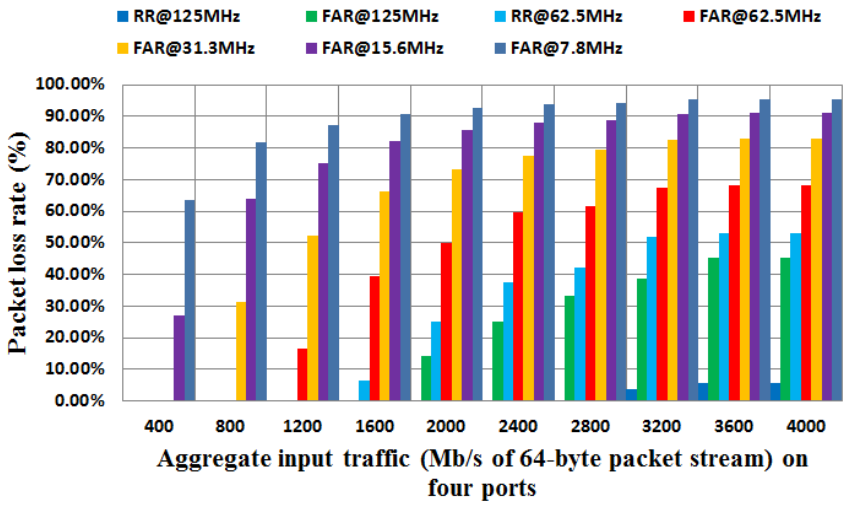

Figure 5.13: Packet loss rate of RR and FAR under different frequencies and different traffic bit rates with 64-byte packet stream

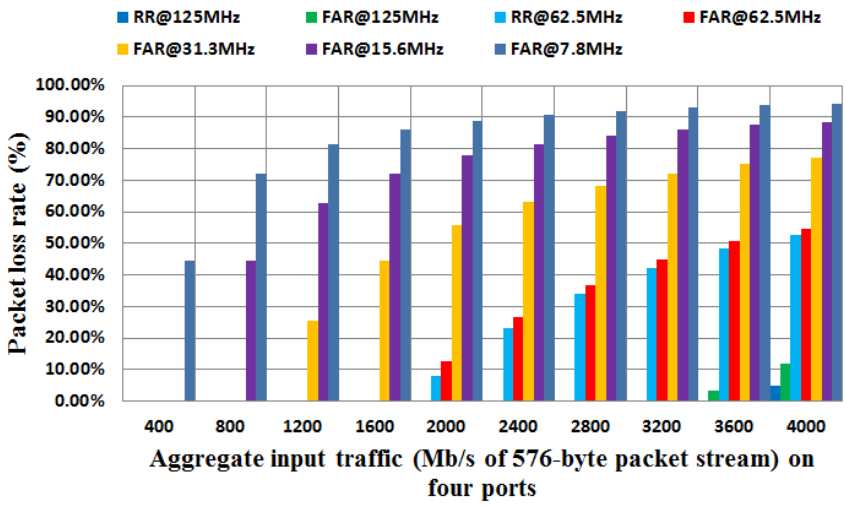

Figure 5.14: Packet loss rate of RR and FAR under different frequencies and different traffic bit rates with 576-byte packet stream

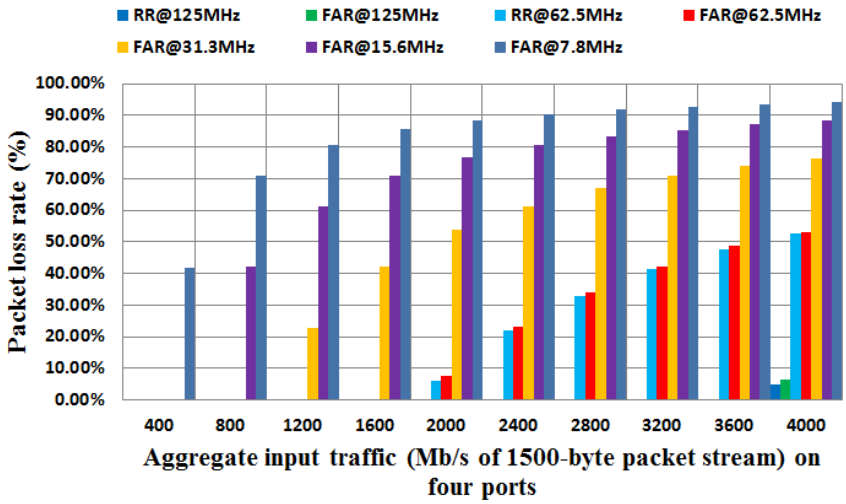

Figure 5.15: Packet loss rate of RR and FAR under different frequencies and different traffic bit rates with 1500-byte packet stream

5.3.6 Frequency adaptation transition time

When toggling the operating frequency of the NetFPGA between 125 MHz and 62.5 MHz on the Reference Router, the frequency switching causes a board reset (which takes about 2 ms) [34, 129] and all the buffered packets are lost. In the

Frequency Adaptive Router, three additional frequencies, 31.3 MHz, 15.6 MHz and 7.8 MHz are introduced for more finely tuned frequency scaling. The board reset problem is also eliminated without significant extra packet processing delay and loss. Since the RR does not provide dynamic frequency scaling capability and all the buffered packets are lost during the board reset, the Frequency Adaptive Router outperforms the Reference Router from the frequency adaptation transition time point of view.

As mentioned in the experimental setup, the NetFPGA board draws power from both the 3.3 V and 5 V power rails of the PCI bus on the host PC through a PCI bus extender. The real time 3.3 V current and 5 V current can be measured from the PCI bus extender. To calculate the frequency adaptation transition time, the samples of the 3.3 V current are the focus. This is because scaling the operating frequency mainly affects the 3.3 V current, since the core logic FPGA draws power from the 3.3 V rail. The PHY components draw power from the 5 V rail which is not affected by the frequency scaling of the core logic FPGA.

Figure 5.16 is an example of adapting the operating frequency of the Frequency Adaptive Router from 7.8 MHz to 15.6 MHz. As shown in the Figure 5.16, when the Frequency Adaptive Router is operating at 7.8 MHz with a very low aggregated traffic load of 108 Mb/s, the 3.3 V current reading is around 0.78 A. Once the traffic load is increased from 108 Mb/s to 228 Mb/s at time 5.62 ms, the 3.3 V current is increased to around 0.82 A immediately corresponding to the current drawn at 15.6 MHz. The maximum throughput that the FAR can handle when operating at 7.8 MHz is 156 Mb/s and the maximum throughput that the FAR can handle when operating at 15.6 MHz is 296 Mb/s. Thus, the FAR switches to 15.6 MHz when the traffic load is increased from 108 Mb/s to 228 Mb/s. The frequency transition

time, calculated from the captured current waveform of the 3.3 V pin on the PCI bus extender, is from around 0.3 ms to around 0.5 ms.

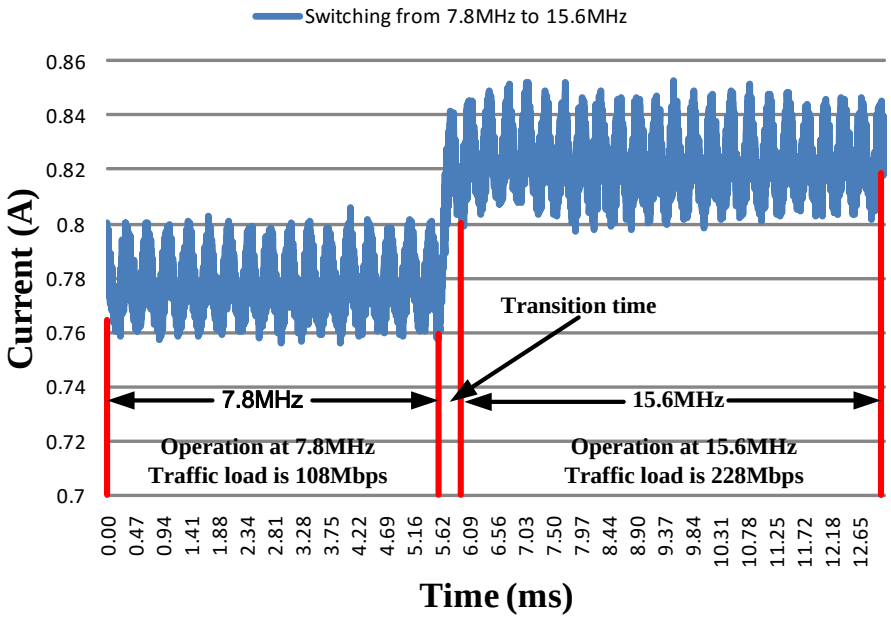

Figure 5.16: An example of frequency adaptation transition time calculation

Time stamp is recorded for each sample of the 3.3 V current data collected. After scaling up and down between two different operating frequencies, the collected 3.3 V current data are exported to a file. Frequency switching will result in significant current change. By locating the starting and ending time stamps of significant current change, the duration of frequency switching can be determined. The transition time of frequency switching can be calculated with following equation:

$$T_s = \frac{T_{c2} - T_{c1}}{764000} \tag{5.6}$$

where T_s is the transition time, T_{c1} is the starting time stamp for significant increase or significant decrease of current, while T_{c2} is the ending time stamp of significant

current change, and 764000 denotes the sampling rate. In this equation, T_{c1} and T_{c2} can be tracked and obtained from the captured 3.3 V current data. Thus, T_s can be calculated and the results are shown in Table 5.3. The frequencies in the column of the table represent the operating frequencies the FAR switches from. The frequencies in the row represent the frequencies the FAR switches to. The frequency adaptation transition times in the two directions are different due to the readings from the 3.3 V current samples may vary slightly resulted from the effect of noise. However, all the frequency adaptation transition times are in the same magnitude, ranging from around 0.3 ms to around 0.5 ms as expected. With frequency adaptation control policies implemented in hardware, the FAR reduces the frequency adaptation transition time by up to 85% compared to the frequency adaptation transition time of the RR (2 ms).

Table 5.3: Frequency adaptation transition time of FAR (from column to row)

Frequency	Transition time (ms)				
	125MHz	62.5MHz	31.3MHz	15.6MHz	7.8MHz
125MHz	0	0.348	0.329	0.297	0.352
62.5MHz	0.306	0	0.273	0.316	0.405
31.3MHz	0.302	0.344	0	0.332	0.362
15.6MHz	0.358	0.263	0.323	0	0.299
7.8MHz	0.291	0.337	0.488	0.293	0

Implementation of frequency adaptation control policy in software is slow compared to dedicated hardware implementation. The dedicated hardware makes an enormous difference in the speed of time-sensitive operations, providing a significant utilisation monitoring advantage over software. Frequency transition time on software implementation of a frequency adaptation control policy consists of the delay in software reading the buffer usage, frequency selection according to a frequency adaptation control policy and setting the appropriate frequency control register, which involves communications

between hardware and software through reading and writing registers. For example, as reported in [69], in theory, the transition time of the adaptive link rate (ALR) is significantly reduced to 1 ms through a newly defined handshake mechanism. However, the rate adaptation control policy of the ALR is implemented using software based utilisation monitoring and buffer thresholds. The actual transition time ranges from 10 ms to 100 ms due to software constraints. The transition time in the work [34] built on the NetFPGA Reference Router, is slightly longer (3.4 ms) than that in the Reference Router (2 ms) [34] [129]. In the work [39], the transition time of adapting the link rates of Ethernet ports is reported to be approximately 2 s, however, the transition time of frequency scaling is not measured.

The power savings from green approaches are mostly achieved at the cost of transition time overhead. The transition time is the time difference between the time when the request for a new state is issued and the time when the request is entered. Transition from one state to another can lead to potential performance degradation, especially in the case of transition from a lower capacity state to higher ones. Longer transition time can result in higher network delay and even severe packet loss during the transition. The transition time in a hardware implementation eliminates the delay in communications between hardware and software. The frequency adaptation control policies of the FAR are directly implemented in hardware. This involves building hardware modules and adding corresponding software instructions to incorporate with the hardware modules. Experimental results indicate that the transition times of the FAR range from 0.3 ms to 0.5 ms.

5.4 Experiments with synthetic traces

To better evaluate the energy efficiency and performance of the proposed Frequency Adaptive Router, synthetic traces with application mix of TCP and UDP traffic (including the Internet's most popular applications HTTP, FTP, Email, DNS, SNMP, streaming media, etc.) were constructed. The synthetic traces imitate the real traffic trace patterns from a border router in the Princeton University campus network [130]. Figure 5.17 shows a clear link traffic volume diurnal pattern. In this diurnal pattern, the maximum traffic bit rate is approaching 649 Mb/s between 15:00 and 17:00, and the average traffic bit rate is about 280 Mb/s.

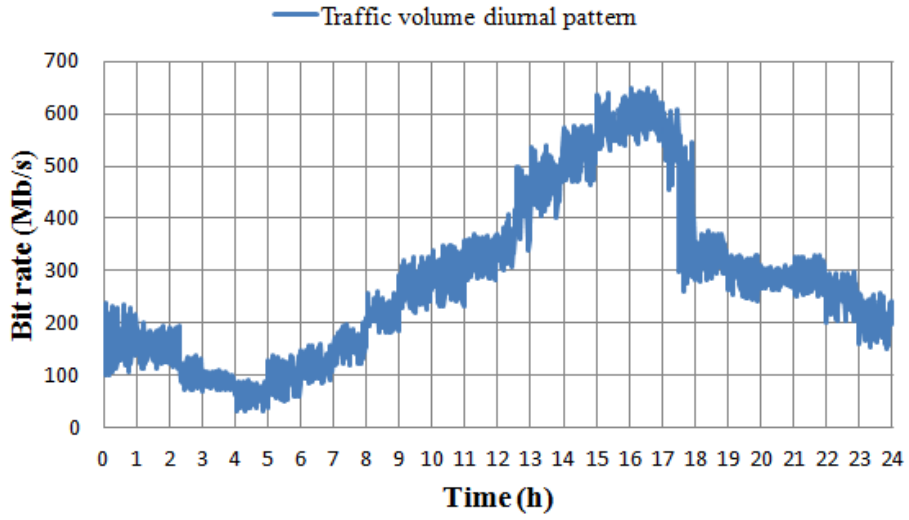

Figure 5.17: Link traffic bit rate diurnal pattern

To quantify the benefits brought from the dynamic frequency scaling technique, a synthetic traffic replicating the traffic pattern in Figure 5.17 is routed through the Reference Router and the Frequency Adaptive Router in turn for power consumption and performance comparison. With the same traffic pattern, Figure 5.18 presents the corresponding operating frequencies of the RR and the FAR (Single Threshold Policy) for the day. As shown in Figure 5.18, the upper red

line at 125 MHz represents the RR is constantly working at 125 MHz. The lower blue curve demonstrates frequency adaptation from time to time in the FAR, indicating that the FAR is capable of adapting to appropriate frequencies in response to the instantaneous traffic load.

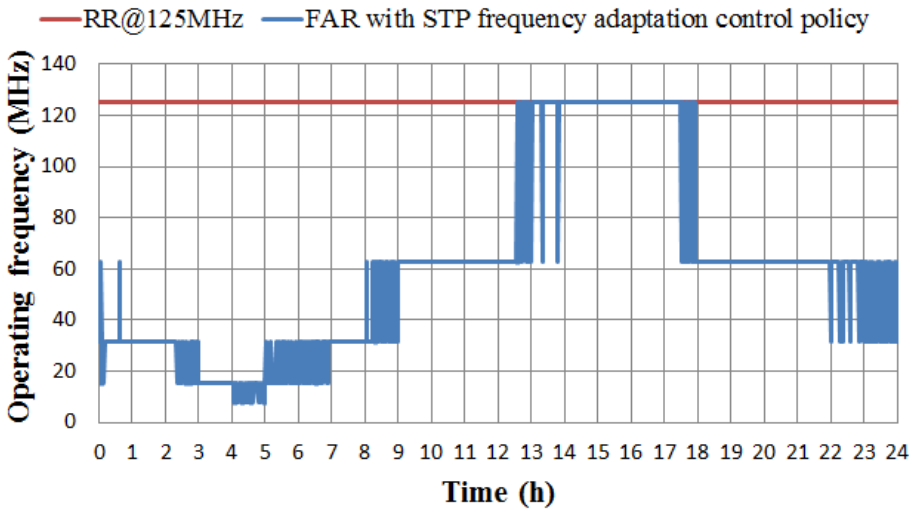

Figure 5.18: Operating frequency of RR and FAR corresponding to traffic trace in Figure 5.17

Figure 5.19 depicts the corresponding power consumption of the RR and the FAR for the day. Since the RR is running at the fixed frequency of 125 MHz, the power consumption of the RR (the upper red curve) fluctuates only in response to the traffic characteristics. However, the power consumption of the FAR (the lower blue curve) fluctuates in response to both traffic characteristics and operating frequencies. The difference between these two curves reveals the power savings from the DFS technique, which indicates that the FAR dynamically adapts the operating frequency of the core FPGA processor among five different frequencies in response to the instantaneous traffic load, rather than leaving the routers running at the highest frequency all the time.

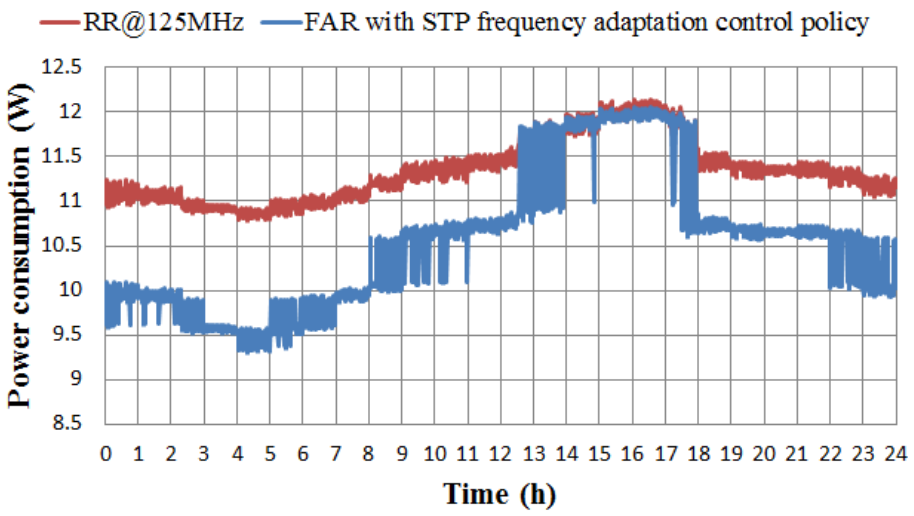

Figure 5.19: Power consumption of RR and FAR corresponding to traffic trace in Figure 5.17

If a fixed operating frequency is used throughout the day, a high frequency will ensure negligible packet loss, even during the peak period from 15:00 to 17:00 hours in Figure 5.17, at the expense of high energy consumption. Adapting the operating frequency in accordance with one of the control policies described in Chapter 4 results in considerable energy savings. The resulting graph of operating frequency versus time is shown in Figure 5.18 for the STP frequency adaptation control policy. Broadly similar results were obtained for the other two dynamic frequency adaptation control policies. The aggregate duration in each frequency mode for different dynamic frequency adaptation control policies are shown in Table 5.4. The second highest available operating frequency (62.5 MHz) was selected for an aggregate of 9 hours and 58 minutes during the day, and the highest frequency (125 MHz) for about 4 hours and 29 minutes. Lower frequencies were needed for the remaining approximately nine and a half hours of the day. This results in a significant power saving. The resulting power savings of the Frequency Adaptive Router with different frequency adaptation control policies are shown in Table 5.5.

Table 5.4: Aggregate duration in each frequency mode

Frequency	STP		DTP		PLAP	
	Time	Ratio	Time	Ratio	Time	Ratio
125MHz	4h29m	18.7%	4h43m	19.7%	4h49m	20.1%
62.5MHz	9h58m	41.5%	9h56m	41.4%	9h47m	40.8%
31.3MHz	5h46m	24.0%	5h47m	24.1%	5h50m	24.3%
15.6MHz	3h17m	13.7%	3h09m	13.1%	3h12m	13.3%
7.8MHz	0h30m	2.1%	0h25m	1.7%	0h22m	1.5%

5.5 Network wide global green techniques

Dynamic frequency scaling provides active performance scaling feature to achieve energy proportional routing, which is an appropriate energy saving approach for network routers given that their connections have to be always on to deal with routing protocols and unpredictable traffic. However, the energy efficiency improvement from dynamic frequency scaling is usually less than that can be achieved from the Ethernet port shutdown. There are challenges to applying Ethernet port shutdown in core routers because, compared to active performance scaling, it takes more time and power to switch between the *on* and *off* state of an Ethernet port. Predicting the idle period of a link in a core router and adapting to the appropriate state is still difficult. The end routers such as home routers, on the other hand, usually follow human behavior and Ethernet ports can be shutdown when people are at sleep to achieve more power savings.

Through green traffic engineering [131] [132], more power consumption can be saved by rerouting traffic to other ports when traffic on a port is low, and turning off the port. Disabling each port can save around 1 W and turning off the ports at both ends of the connection can save around 2 W. For the NetFPGA router, turning off all four Ethernet ports can save around 4 W, while scaling down the frequency from 125 MHz to 7.8 MHz while keeping all four Ethernet ports active can only save a maximum of around 2.2 W. Thus, traffic rerouting through network wide

energy-aware traffic engineering coupled with disabling Ethernet ports is a more effective way to reduce power consumption.

An Ethernet port shutdown module is integrated into the Frequency Adaptive Router to disable an Ethernet port when there is no traffic going through the port for a period of time. With this module, the router can save significant power compared to leaving Ethernet ports fully on all the time. In the Ethernet port shutdown module, several byte counters are introduced to count the number of bytes passing through each Ethernet FIFO queue. The Ethernet ports can be controlled by a network level power management technique. Energy aware traffic engineering is a promising network level technique to manage the routing path and disable the Ethernet ports as many as possible without significant packets delay and loss. If a network level technique is involved, network wide coordination such as the green abstract layer (GAL) [133] is compulsory. The GAL can provide advanced power management capabilities to decouple distributed high level algorithms from heterogeneous hardware.

For Ethernet port shutdown, the control policy combines a network level power management technique with a local level approach. Ethernet port shutdown should be controlled at a higher level by a network wide global power management policy with energy aware traffic engineering capability, so that traffic destined for a port can be diverted to other active ports to enable the safe shut down of a port. Otherwise, a sudden surge of traffic to a disabled port can result in significant packet loss, as the disabled port will be unable to be turned on quick enough when new packets arrive. With energy aware traffic engineering, the disabled port is able to be woken up in advance allowing traffic to traverse the port again. The energy aware traffic engineering is performed at a scheduler to decide when to safely enable or disable every Ethernet port. Figure 5.20 presents

a simple example of energy aware traffic engineering. Suppose router A needs to send 300 Mb/s aggregate traffic to router E. To prevent network congestion from traffic burst, traditional routing with load balancing may split the 300 Mb/s traffic into 100 Mp/s for each of the three links as shown in Figure 5.20 (a). Whereas, energy aware traffic engineering reroutes and aggregates the 300 Mb/s traffic from three separate links (each link 100 Mb/s) to one single link (300 Mb/s), so that the Ethernet ports on the two links without traffic can be disabled for a period of time to achieve more power savings as shown in Figure 5.20 (b).

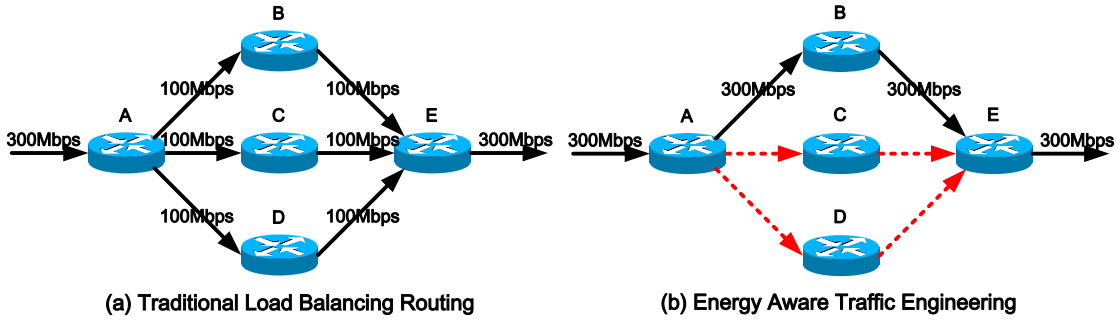

Figure 5.20: Difference between traditional load balancing routing and energy aware traffic engineering

If the Frequency Adaptive Router is in idle state and no traffic is being processed, all four Ethernet ports on the FAR can be disabled and its operating frequency is switched to the lowest frequency 7.8 MHz. In this state, the power consumption of the FAR is 5.118 W. If this 5.118 W power consumption state of the FAR is compared to the 10.725 W power consumption state of the RR running at 125 MHz with all four Ethernet ports on, there is a power saving of up to 52%. Table 5.5 summarizes the average total power consumption for the experiments with 24 hours synthetic traffic traces in Figure 5.17. The RR has the highest power consumption of 11.750 W. The FAR with PLAP consumes the least power consumption of 10.565 W and saves 10.09% power consumption compared with

that of the RR. The differences of power consumption among the three different dynamic frequency adaptation control policies in the FAR are small, because the differences only come from different number of transitions. However, if dynamic frequency scaling (DFS), traffic rerouting through network wide green traffic engineering (GTE) and Ethernet port shutdown (EPS) are all implemented, up to 46% power consumption can be saved in this specific experiment. Table 5.6 summarizes the control policy for the specific DFS, GTE and EPS implementation on the FAR in detail.

Table 5.5: Average total power consumption for the day

Type	Average total power (W)	Power Saving (%)
RR	11.750	0%
FAR with STP	10.565	10.09%
FAR with DTP	10.568	10.05%
FAR with PLAP	10.569	10.04%
FAR with DFS,GTE&EPS	6.321	46.04%

Table 5.6: Control policy for DFS, GTE and EPS implementation on FAR

aggregated traffic (Mbp/s)	control policy for aggregating 4 ports traffic (number of active ports and operating frequency)
0 to 232	from 4 ports to 1 port at 7.8 MHz
233 to 468	from 4 ports to 1 port at 15.6 MHz
469 to 952	from 4 ports to 1 port at 31.3 MHz
953 to 1876	from 4 ports to 2 ports at 62.5 MHz
1877 to 2000	from 4 ports to 2 ports at 125 MHz
2001 to 3000	from 4 ports to 3 ports at 125 MHz
3001 to 4000	DFS, GTE and EPS are not implemented

5.6 Summary

This chapter describes the experimental setup and the metrics used for the energy efficiency and performance evaluation of the RR and the FAR. The power consumption of the RR and the FAR are measured with fine-grained power consumption measurements through a dedicated PCI bus extender. The

performance of the RR and the FAR under different traffic bit rates, different packet sizes and different operating frequencies are measured and compared in detail. Frequency adaptation control policies governing the dynamic frequency adaptation of the FAR are also evaluated and discussed.

It is concluded that, for quiescent power consumption and total power consumption, the FAR consumes less power than the RR by up to 52% and 46% respectively. Compared to the RR, the power savings from the FAR are achieved at the expense of less peak measured throughput about 4% and more packet loss rate about 4%. However, the FAR shares the same round trip time as the RR. With frequency adaptation control policy implemented in hardware, the FAR reduces frequency adaptation transition time by up to 85% compared to the RR. Table 5.7 summarizes the results.

Table 5.7: Summary of power consumption and performance comparisons between the Reference Router and Frequency Adaptive Router

Metric Name	Comparison Result
Quiescent Power Consumption	FAR less than RR by up to 52%
Total Power Consumption	FAR less than RR by up to 46%
Peak Measured Throughput	FAR less than RR about 4%
Round Trip Time	FAR the same as RR
Packet Loss Rate	FAR more than RR about 4%
Transition Time	FAR less than RR by up to 85%

As indicated in Table 5.5, through dynamic frequency scaling, up to 10% power consumption can be saved for a typical daily traffic pattern. However, if local dynamic frequency adaptation and network wide global green techniques, such as energy aware traffic engineering and Ethernet port shutdown, are used together, up to 46% power consumption can be saved for the typical daily traffic pattern.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

The NetFPGA 1G board is a research platform with open source hardware and software. Developers can start with the provided NetFPGA packages and modify them to design and implement their own mechanisms or algorithms with new custom modules. A green Frequency Adaptive Router with energy proportional routing capability is prototyped on the NetFPGA Reference Router. This is accomplished by integrating a custom frequency division module and an AFIFO module into the Reference Router. The custom frequency division module in the DCM provides advanced clocking capability which can generate new clock frequencies by dividing source clock frequency with different allowed divisors. Three additional operating frequencies (31.3 MHz, 15.6 MHz and 7.8 MHz) in the Frequency Adaptive Router are derived from the source clock 125 MHz with three frequency divisors (4, 8 and 16). The AFIFO module is added to eliminate the board reset problem, allowing the core logic FPGA in the Frequency Adaptive Router to adapt its operating frequencies in response to the instantaneous traffic load on the fly.

Based on the proposed prototype, this work further investigates and examines the fine-grained energy proportional DFS technique at the five different operating frequencies with different number of active ports, traffic bit rates and packet sizes, which can be used to better quantify the energy savings from DFS scheme. The research carried out in this work was part of the European FP7 ECONET project, which has resulted in an European Telecommunications Standards Institute (ETSI) standard 203-237.

Experimental results indicate that dynamic frequency scaling can effectively reduce the power consumption of hardware components inside a network router by up to 10%. Control policies governing frequency adaptation are based on current traffic levels while avoiding performance degradation such as increased network delay and packet loss. When traffic load is low, power consumption of the router can be reduced by switching to a lower operating frequency with lower routing capacity without performance degradation. In the case of no traffic to be handled, the router can be switched to an idle state and shut down the Ethernet ports, which can result up to 52% of power savings. Experiments with synthetic traces that reflect traffic pattern in real network traffic traces indicate that up to 46% of power savings can be achieved by integrating dynamic frequency scaling with Ethernet ports shutdown and energy aware traffic engineering. The frequency adaptation control policies of the FAR are implemented in the NetFPGA hardware, which reduces the frequency adaptation transition time by up to 85% compared to software implementation. Although these numbers relate to a specific NetFPGA implementation, the design principles are of general application and can be deployed in commercial hardware to significantly lower its power consumption.

6.2 Future Works

The same frequency adaptation mechanism implemented on the Frequency Adaptive Router can be also implemented on the NetFPGA OpenFlow switch. In the ECONET project, a green abstract layer (GAL) is designed to provide a standard interface between data and control planes for exchanging information regarding the power status of a device. GAL can centrally control the Frequency Adaptive Router. In the follow-on H2020 project INPUT, the Frequency Adaptive OpenFlow switch will be combined with a distributed software router (DROP) and the Green Abstract Layer (GAL) to build an advanced green OpenFlow switch system in future works.

A level of 95% threshold in the STP frequency adaptation control policy was chosen for consistency with prior work on ALR, as was the 10 ms sampling period. These choice of parameters involves tradeoff between energy saving and packet loss rate. A sensitivity analysis of the selection of threshold values, related packet loss rate and the amount of hysteresis required will be undertaken in future works.

There are also other ways to reduce the power consumption of the Reference Router, such as clock gating. A BUFGCE is a submodule in the BUFGMUX available in virtex II Pro FPGA. The BUFGCE is a global clock buffer incorporating a smart enable function that avoids output glitches or runt pulses. If the BUFGCE input is inactive (Low) prior to the incoming rising clock edge, the clock pulse does not pass through the clock buffer. With BUFGCE, clock gating will be investigated and examined in future works.

Appendix A

List of Publications

Publications included in the thesis in chronological order:

Journal

- **Feng Guo**, Olga Ormond, Leonardo Fialho, Martin Collier and Xiaojun Wang. "Power consumption analysis of a NetFPGA based router." The Journal of China Universities of Posts and Telecommunications, Elsevier. 2012, 19: 94-99.
- **Feng Guo**, Xiaojun Wang, Mei Song, Yifei Wei, Olga Ormond and Martin Collier. "Greening the NetFPGA reference router." Energies, Multidisciplinary Digital Publishing Institute Publishing. 2016, 9(7), 500.

Conference

- **Feng Guo**, Olga Ormond, Martin Collier and Xiaojun Wang. "Power measurement of NetFPGA based router." 2012 IEEE Online Conference on Green Communications (GreenCom). IEEE, 2012: 116-119.

- Jie Jin, Lingling Sun, **Feng Guo** and Xiaojun Wang. "Low power design for on-chip networking processing system." 2015 28th IEEE International System-on-Chip Conference (SOCC). IEEE, 2015: 154-159.

- **Feng Guo**, Mei Song, Yifei Wei, Luigi Sambolino, Pengcheng Liu, Xiaojun Wang and Martin Collier. "Green Precision Time Protocol Router Using Dynamic Frequency Scaling." International Conference on Human Centered Computing, Springer International Publishing. 2016: 104-115.

Publications not included in the thesis in chronological order:

Journal

- Yousheng Zhou, Junfeng Zhou, Feng Wang and **Feng Guo**. "An Efficient Chaotic Map-Based Authentication Scheme with Mutual Anonymity." Applied Computational Intelligence and Soft Computing, Hindawi. 2016.

Conference

- Ming Zhao, Tao Luo, Guangxin Yue, Xiaojun Wang and **Feng Guo**. "Multiuser power control with competitive market equilibrium." Wireless Communications & Signal Processing (WCSP), 2012 International Conference on. IEEE, 2012: 1-5.

- Tom Molloy, Xing Zheng, Olga Ormond, **Feng Guo** and Xiaojun Wang. "Power consumption in a zfilter publish/subscribe based forwarding node." Information and Communications Technologies (IETICT 2013), IET International Conference on. IET, 2013: 14-20 (Best paper award).

- Yifei Wei, Xiaojun Wang, **Feng Guo**, Gabriel Hogan and Martin Collier. "Energy saving local control policy for green reconfigurable routers." 2015 IEEE International Conference on Communications (ICC). IEEE, 2015: 221-225.

References

[1] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti. Energy efficiency in the future Internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures. *Communications Surveys & Tutorials, IEEE*, 13(2):223–244, 2011.

[2] The EU climate and energy package 2020. European Commission. http://ec.europa.eu/clima/policies/strategies/2020/index_en.htm. 2007. (accessed on 1 August 2016).

[3] M. Webb et al. Smart 2020: Enabling the low carbon economy in the information age. *The Climate Group. London*, 1(1):1–1, 2008.

[4] British Telecom. British Telecom Sustainability Report 2010. 2010.

[5] Telecom Italia. Telecom Italia Sustainability Report 2011. 2011.

[6] Telecom Italia. Telecom Italia Sustainability Report 2015. 2015.

[7] Deutsche Telecom. Deutsche Telecom Corporate Responsibility Report 2011. 2011.

[8] Deutsche Telecom. Deutsche Telecom Corporate Responsibility Report 2015. 2015.

[9] Telefonica. Telefonica Sustainability Report 2015. 2015.

[10] France Telecom. France Telecom Corporate Social Responsibility Report 2015. 2015.

[11] Verizon. Verizon Corporate Responsibility Report 2012. 2012.

[12] AT&T. AT&T Sustainability Report 2012. 2012.

[13] Nippon Telegraph and Telephone Group. Nippon Telegraph and Telephone Group Corporate Social Responsibility Report 2015. 2015.

[14] China Mobile. China Mobile Sustainability Report 2011. 2011.

[15] China Mobile. China Mobile Sustainability Report 2015. 2015.

[16] BT Press. Bt announces major wind power plans, 2007.

[17] IT Wales. Green evangelist to call for big changes in computer use to aid environment. In *ITWales Conf., November*, 2007.

[18] K. Kawamoto, J. Koomey, B. Nordman, R. Brown, M. Piette, M. Ting, and A. Meier. Electricity used by office equipment and network equipment in the US: Detailed report and appendices. 2001.

[19] Ministry of Internal Affairs and Communications. Report by investigation society on ICT system and network that contributes to a decrease in environmental impact. 2007.

[20] Trade Ministry of Economy and Industry. Energy in Japan 2006. 2006.

[21] United States Energy Information Administration. State Electricity Profiles 2015. 2015.

[22] J. Moyer and B. Hughes. ICTs: Do they contribute to increased carbon emissions? *Technological Forecasting and Social Change*, 79(5):919–931, 2012.

[23] Gartner Inc. http://www.gartner.com/newsroom/id/503867. 2007. (accessed on 1 August 2016).

[24] G. Zhang, Guo. Zhang, Q. Yang, S. Cheng, and T. Zhou. Evolution of the Internet and its cores. *New Journal of Physics*, 10(12):123027, 2008.

[25] Raffaele Bolla, Roberto Bruschi, Antonio Cianfrani, and Marco Listanti. Introducing standby capabilities into next-generation network devices. In *Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow*, page 10. ACM, 2010.

[26] Telegeography backone status and trends. http://bpastudio.csudh.edu/fac/lpress/471/hout/telegeographygig_execsumm.pdf. 2007. (accessed on 1 August 2016).

[27] Chamara Gunaratne, Ken Christensen, and Bruce Nordman. Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed. *International Journal of Network Management*, 15(5):297–310, 2005.

[28] Kenneth J Christensen, Chamara Gunaratne, Bruce Nordman, and Alan D George. The next frontier for communications networks: power management. *Computer Communications*, 27(18):1758–1770, 2004.

[29] Andrew Odlyzko. Data networks are lightly utilized, and will stay that way. *Review of Network Economics*, 2(3), 2003.

[30] Ken Christensen, Pedro Reviriego, Bruce Nordman, Michael Bennett, Mehrgan Mostowfi, and Juan Antonio Maestro. IEEE 802.3 az: the road to energy efficient Ethernet. *IEEE Communications Magazine*, 48(11):50–56, 2010.

[31] Masaki Yamada, Takeki Yazaki, Nobuhito Matsuyama, and Takehisa Hayashi. Power efficient approach and performance control for routers. In *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, pages 1–5. IEEE, 2009.

[32] Ishfaq Ahmad and Sanjay Ranka. *Handbook of Energy-Aware and Green Computing-Two Volume Set*. Chapman and Hall/CRC, 2012.

[33] G. Watson, N. McKeown, and M. Casado. NetFPGA: A tool for network research and education. In *2nd workshop on Architectural Research using FPGA Platforms (WARFP)*, volume 3, 2006.

[34] Wei Meng, Yi Wang, Chengchen Hu, Keqiang He, Jun Li, and Bin Liu. Greening the Internet using multi-frequency scaling scheme. In *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, pages 928–935. IEEE, 2012.

[35] Chamara Gunaratne and Ken Christensen. Ethernet adaptive link rate: System design and performance evaluation. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 28–35. IEEE, 2006.

[36] Luiz André Barroso and Urs Hölzle. The case for energy-proportional computing. *Computer*, 40(12), 2007.

[37] Philip Walters. Small network equipment energy consumption in us homes. 2013.

[38] ISPreview. http://www.ispreview.co.uk/index.php/2017/01/energy-usage-uk-home-broadband-routers-big-isps-compared.html/3. 2017. (accessed on 1 April 2017).

[39] Nam Pham Ngoc, Thanh Nguyen Huu, Trong Vu Quang, Vu Tran Hoang, Huong Truong Thu, Phuoc Tran-Gia, and Christian Schwartz. A new power profiling method and power scaling mechanism for energy-aware NetFPGA Gigabit router. *Computer Networks*, 78:4–25, 2015.

[40] Aruna Prem Bianzino, Claude Chaudet, Dario Rossi, and Jean-Louis Rougier. A survey of green networking research. *IEEE Communications Surveys & Tutorials*, 14(1):3–20, 2012.

[41] M Nishan Dharmaweera, Rajendran Parthiban, and Y Ahmet Şekercioğlu. Toward a power-efficient backbone network: the state of research. *IEEE Communications Surveys & Tutorials*, 17(1):198–227, 2015.

[42] Melike Erol-Kantarci and Hussein T Mouftah. Energy-efficient information and communication infrastructures in the smart grid: A survey on interactions and open issues. *IEEE Communications Surveys & Tutorials*, 17(1):179–197, 2015.

[43] Maruti Gupta and Suresh Singh. Greening of the Internet. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 19–26. ACM, 2003.

[44] Christoph Lange, Dirk Kosiankowski, Rainer Weidmann, and Andreas Gladisch. Energy consumption of telecommunication networks and related improvement options. *IEEE Journal of Selected Topics in Quantum Electronics*, 17(2):285–295, 2011.

[45] Willem Vereecken, Ward Van Heddeghem, Margot Deruyck, Bart Puype, Bart Lannoo, Wout Joseph, Didier Colle, Luc Martens, and Piet Demeester. Power consumption in telecommunication networks: overview and reduction strategies. *IEEE Communications Magazine*, 49(6):62–69, 2011.

[46] R. Bolla, F. Davoli, R. Bruschi, K. Christensen, F. Cucchietti, and S. Singh. The potential impact of green technologies in next-generation wireline networks: Is there room for energy saving optimization? *Communications Magazine, IEEE*, 49(8):80–86, 2011.

[47] J. Restrepo, C. Gruber, and C. Machuca. Energy profile aware routing. In *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, pages 1–5. IEEE, 2009.

[48] M. Jimeno, K. Christensen, and B. Nordman. A network connection proxy to enable hosts to sleep and save energy. In *Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International*, pages 101–110. IEEE, 2008.

[49] E. Le Sueur and G. Heiser. Dynamic voltage and frequency scaling: The laws of diminishing returns. In *Proceedings of the 2010 international conference on Power aware computing and systems*, pages 1–8. USENIX Association, 2010.

[50] ENERGY STAR. Energy Star. *History: ENERGY STAR*, 2011.

[51] Marla C Sanchez, Richard E Brown, Carrie Webber, and Gregory K Homan. Savings estimates for the United States environmental protection agency's ENERGY STAR voluntary product labeling program. *Energy policy*, 36(6):2098–2108, 2008.

[52] Gale Boyd, Elizabeth Dutrow, and Walt Tunnessen. The evolution of the ENERGY STAR energy performance indicator for benchmarking industrial plant manufacturing energy use. *Journal of cleaner production*, 16(6):709–715, 2008.

[53] Maruti Gupta and Suresh Singh. Using low-power modes for energy conservation in Ethernet LANs. In *INFOCOM*, volume 7, pages 2451–2455, 2007.

[54] K. Christensen and F. Gulledge. Enabling power management for network-attached computers. *International Journal of Network Management*, 8(2):120–130, 1998.

[55] Bruce Nordman and Ken Christensen. Improving the energy efficiency of Ethernet-connected devices: A proposal for proxying. *white paper, version*, 1, 2007.

[56] Madhu Gupta and Sushil Singh. Dynamic Ethernet link shutdown for energy conservation on Ethernet links. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 6156–6161. IEEE, 2007.

[57] Thomas D Burd and Robert W Brodersen. Energy efficient CMOS microprocessor design. In *System Sciences, 1995. Proceedings of the Twenty-Eighth Hawaii International Conference on*, volume 1, pages 288–297. IEEE, 1995.

[58] Mark Horowitz, Elad Alon, Dinesh Patil, Samuel Naffziger, Rajesh Kumar, and Kerry Bernstein. Scaling, power, and the future of CMOS. In *IEEE InternationalElectron Devices Meeting, 2005. IEDM Technical Digest.*, pages 7–pp. IEEE, 2005.

[59] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall. Reducing network energy consumption via sleeping and rate-adaptation. In *NSDI*, volume 8, pages 323–336, 2008.

[60] L. Benini, A. Bogliolo, G. Paleologo, and G. De Micheli. Policy optimization for dynamic power management. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 18(6):813–833, 1999.

[61] E. Chung, L. Benini, A. Bogliolo, Y. Lu, and G. De Micheli. Dynamic power management for nonstationary service requests. *Computers, IEEE Transactions on*, 51(11):1345–1361, 2002.

[62] M. Srivastava, A. Chandrakasan, and R. Brodersen. Predictive system shutdown and other architectural techniques for energy efficient programmable computation. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 4(1):42–55, 1996.

[63] A. Miyoshi, C. Lefurgy, E. Van Hensbergen, R. Rajamony, and R. Rajkumar. Critical power slope: understanding the runtime effects of frequency scaling. In *Proceedings of the 16th international conference on Supercomputing*, pages 35–44. ACM, 2002.

[64] Z. Ren, B. Krogh, and R. Marculescu. Hierarchical adaptive dynamic power management. *Computers, IEEE Transactions on*, 54(4):409–420, 2005.

[65] R. Golding, P. Bosch, and J. Wilkes. Idleness is not sloth. In *USENIX Winter*, pages 201–212, 1995.

[66] S. Gochman, R. Ronen, I. Anati, A. Berkovits, T. Kurts, A. Naveh, A. Saeed, Z. Sperber, and R. Valentine. The Intel Pentium M processor: microarchitecture and performance. *Intel Technology Journal*, 7(2):21–36, 2003.

[67] R. Chabukswar. Maximizing performance and energy-efficiency on Intel@ CoreTM microarchitecture using multi-threading. *Intel White paper*, 2006.

[68] S. Herbert and D. Marculescu. Analysis of dynamic voltage/frequency scaling in chip-multiprocessors. In *Low Power Electronics and Design (ISLPED), 2007 ACM/IEEE International Symposium on*, pages 38–43. IEEE, 2007.

[69] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen. Reducing the energy consumption of Ethernet with adaptive link rate (ALR). *Computers, IEEE Transactions on*, 57(4):448–461, 2008.

[70] G Le Lann and P Rolin. The 802.3 D protocol: A variation on the IEEE 802.3 standard for real-time LANs. *INRIABP 105, F*, 78153, 1987.

[71] Pedro Reviriego, José Alberto Hernández, David Larrabeiti, and Juan Antonio Maestro. Performance evaluation of energy efficient Ethernet. *IEEE Communications Letters*, 13(9):697–699, 2009.

[72] R. Bolla, R. Bruschi, and A. Ranieri. Green support for PC-based software router: performance evaluation and modeling. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–6. IEEE, 2009.

[73] R. Bolla, R. Bruschi, F. Davoli, and A. Ranieri. Energy-aware performance optimization for next-generation green network equipment. In *Proceedings of the 2nd ACM SIGCOMM workshop on Programmable routers for extensible services of tomorrow*, pages 49–54. ACM, 2009.

[74] Greg Semeraro, Grigorios Magklis, Rajeev Balasubramonian, David H Albonesi, Sandhya Dwarkadas, and Michael L Scott. Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling. In *High-Performance Computer Architecture, 2002. Proceedings. Eighth International Symposium on*, pages 29–40. IEEE, 2002.

[75] Kihwan Choi, Ramakrishna Soma, and Massoud Pedram. Fine-grained dynamic voltage and frequency scaling for precise energy and performance tradeoff based on the ratio of off-chip access to on-chip computation times. *IEEE transactions on computer-aided design of integrated circuits and systems*, 24(1):18–28, 2005.

[76] Johan Pouwelse, Koen Langendoen, and Henk Sips. Dynamic voltage scaling on a low-power microprocessor. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 251–259. ACM, 2001.

[77] Padmanabhan Pillai and Kang G Shin. Real-time dynamic voltage scaling for low-power embedded operating systems. In *ACM SIGOPS Operating Systems Review*, volume 35, pages 89–102. ACM, 2001.

[78] Kihwan Choi, Ramakrishna Soma, and Massoud Pedram. Dynamic voltage and frequency scaling based on workload decomposition. In *Proceedings of the 2004 international symposium on Low power electronics and design*, pages 174–179. ACM, 2004.

[79] Arindam Mallik, Bin Lin, Gokhan Memik, Peter Dinda, and Robert P Dick. User-driven frequency scaling. *IEEE Computer Architecture Letters*, 5(2):16–16, 2006.

[80] Yung-Hsiang Lu, Luca Benini, and Giovanni De Micheli. Dynamic frequency scaling with buffer insertion for mixed workloads. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 21(11):1284–1305, 2002.

[81] Tran Hoang Vu, Tran Thanh, Vu Quang Trong, Nguyen Huu Thanh, and Pham Ngoc Nam. Energy saving for OpenFlow switch on the NetFPGA platform using multi-frequency. *Int J Comput Netw Technol*, 1:9–15, 2014.

[82] Ralf Huelsermann, Matthias Gunkel, Clara Meusburger, and Dominic A Schupke. Cost modeling and evaluation of capital expenditures in optical multilayer networks. *Journal of Optical Networking*, 7(9):814–833, 2008.

[83] Gangxiang Shen and Rodney S Tucker. Energy-minimized design for ip over wdm networks. *Journal of Optical Communications and Networking*, 1(1):176–186, 2009.

[84] Jayant Baliga, Robert Ayre, Kerry Hinton, and Rodney S Tucker. Energy consumption in wired and wireless access networks. *IEEE Communications Magazine*, 49(6), 2011.

[85] Biswanath Mukherjee. *Optical WDM networks*. Springer Science & Business Media, 2006.

[86] Satish Sharma. A review of passive optical networks, 2013.

[87] Frank Effenberger, David Cleary, Onn Haran, Glen Kramer, Ruo Ding Li, Moshe Oron, and Thomas Pfeiffer. An introduction to PON technologies [topics in optical communications]. *IEEE Communications Magazine*, 45(3):S17–S25, 2007.

[88] Ken-ichi Sato. Optical technologies that enable green networks. In *2010 12th International Conference on Transparent Optical Networks*, pages 1–4. IEEE, 2010.

[89] Josep Prat et al. Next–generation FTTH passive optical networks. *Springer Science+ Business Media BV*, 2008.

[90] C Siva Ram Murthy and Mohan Gurusamy. *WDM optical networks: concepts, design, and algorithms*. Prentice Hall, 2002.

[91] Filip Idzikowski, Edoardo Bonetto, Luca Chiaraviglio, Antonio Cianfrani, Angelo Coiro, Raúl Duque, Felipe Jiménez, Esther Le Rouzic, Francesco Musumeci, Ward Van Heddeghem, et al. TREND in energy-aware adaptive routing solutions. *Communications Magazine, IEEE*, 51(11):94–104, 2013.

[92] Paweł Wiatr, Jiajia Chen, Paolo Monti, and Lena Wosinska. Green WDM-PONs: Exploiting traffic diversity to guarantee packet delay limitation. In *Optical Network Design and Modeling (ONDM), 2013 17th International Conference on*, pages 229–234. IEEE, 2013.

[93] Paolo Monti, Cicek Cavdar, I Cerutti, J Chen, Ajmal Mohammad, L Velasco, P Wiatr, and L Wosinska. Green optical networks: Power savings versus network performance. *Green Communications: Principles, Concepts and Practice*, pages 291–308, 2015.

[94] Luca Chiaraviglio, Pawel Wiatr, Paolo Monti, Jiajia Chen, Lena Wosinska, Josip Lorincz, Filip Idzikowski, and Marco Listanti. Impact of energy-efficient techniques on a device lifetime. In *Green Communications (OnlineGreencomm), 2014 IEEE Online Conference on*, pages 1–6. IEEE, 2014.

[95] Luca Chiaraviglio, Pawel Wiatr, Paolo Monti, Jiajia Chen, Josip Lorincz, Filip Idzikowski, Marco Listanti, and Lena Wosinska. Is green networking beneficial in terms of device lifetime? *IEEE Communications Magazine*, 53(5):232–240, 2015.

[96] Pawel Wiatr, Paolo Monti, and Lena Wosinska. Green lightpath provisioning in transparent WDM networks: Pros and cons. In *2010 IEEE 4th International Symposium on Advanced Networks and Telecommunication Systems*, pages 10–12. IEEE, 2010.

[97] Pawel Wiatr, Jiajia Chen, Paolo Monti, and Lena Wosinska. Energy saving in access networks: Gain or loss from the cost perspective? In *2013 15th International Conference on Transparent Optical Networks (ICTON)*, pages 1–6. IEEE, 2013.

[98] Pawel Wiatr, Paolo Monti, and Lena Wosinska. Power savings versus network performance in dynamically provisioned WDM networks. *IEEE Communications Magazine*, 50(5):48–55, 2012.

[99] Shing-Wa Wong, Luca Valcarenghi, She-Hwa Yen, Divanilson R Campelo, Shinji Yamashita, and Leonid Kazovsky. Sleep mode for energy saving PONs: advantages and drawbacks. In *2009 IEEE Globecom Workshops*, pages 1–6. IEEE, 2009.

[100] Shing-Wa Wong, She-Hwa Yen, Pegah Afshar, Shinji Yamashita, and Leonid G Kazovsky. Demonstration of energy conserving TDM-PON with sleep mode ONU using fast clock recovery circuit. In *Optical Fiber Communication Conference*, page OThW7. Optical Society of America, 2010.

[101] Luca Chiaraviglio, Delia Ciullo, Marco Mellia, and Michela Meo. Modeling sleep mode gains in energy-aware networks. *Computer Networks*, 57(15):3051–3066, 2013.

[102] Antonio Cianfrani, Vincenzo Eramo, Marco Listanti, Marco Marazza, and Enrico Vittorini. An energy saving routing algorithm for a green ospf protocol. In *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pages 1–5. IEEE, 2010.

[103] Kin-Hon Ho and Chi-Chung Cheung. Green distributed routing protocol for sleep coordination in wired core networks. In *Networked Computing (INC), 2010 6th International Conference on*, pages 1–6. IEEE, 2010.

[104] M. Gupta, S. Grover, and S. Singh. A feasibility study for power management in LAN switches. In *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*, pages 361–371. IEEE, 2004.

[105] Kurt Roth, Fred Goldstein, and Jonathan Kleinman. Energy consumption by office and telecommunications equipment in commercial buildings volume i: energy consumption baseline. *National Technical Information Service (NTIS), US Department of Commerce, Springfield, VA*, 22161, 2002.

[106] Bruce Nordman and Ken Christensen. Reducing the energy consumption of network devices. *Tutorial presented at the July*, 2005.

[107] Christoph Lange. Energy-related aspects in backbone networks. In *Proceedings of 35th European Conference on Optical Communication (ECOC 2009),(Wien, AU)*, 2009.

[108] Luca Chiaraviglio, Marco Mellia, and Fabio Neri. Minimizing ISP network energy cost: formulation and solutions. *IEEE/ACM Transactions on Networking (TON)*, 20(2):463–476, 2012.

[109] J. Lockwood, N. McKeown, G. Watson, G. Gibb, P. Hartke, J. Naous, R. Raghuraman, and J. Luo. NetFPGA–an open platform for Gigabit-rate network switching and routing. In *Microelectronic Systems Education, 2007. MSE'07. IEEE International Conference on*, pages 160–161. IEEE, 2007.

[110] G. Gibb, J. Lockwood, J. Naous, P. Hartke, and N. McKeown. NetFPGA: An open platform for teaching how to build Gigabit-rate network switches and routers. *Education, IEEE Transactions on*, 51(3):364–369, 2008.

[111] J. Naous, G. Gibb, S. Bolouki, and N. McKeown. NetFPGA: reusable router architecture for experimental research. In *Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow*, pages 1–7. ACM, 2008.

[112] Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet. Product Specification. Xilinx Inc. http://www.xilinx.com/support/documentation/data_sheets/ds083.pdf. 2011. (accessed on 1 August 2016).

[113] Virtex-II Pro and Virtex-II Pro X FPGA User Guide. Xilinx Inc. http://www.xilinx.com/support/documentation/user_guides/ug012.pdf. 2007. (accessed on 1 August 2016).

[114] Erol Gelenbe, Guy Pujolle, and JCC Nelson. *Introduction to queueing networks*. Wiley Chichester, 1998.

[115] János Sztrik. Basic queueing theory. *University of Debrecen, Faculty of Informatics*, 193, 2012.

[116] Arnold O Allen. *Probability, statistics, and queueing theory*. Academic Press, 2014.

[117] David Nguyen, Abhijit Davare, Michael Orshansky, David Chinnery, Brandon Thompson, and Kurt Keutzer. Minimization of dynamic and static power through joint assignment of threshold voltages and sizing optimization. In *Proceedings of the 2003 international symposium on Low power electronics and design*, pages 158–163. ACM, 2003.

[118] Nam Sung Kim, Todd Austin, David Baauw, Trevor Mudge, Krisztián Flautner, Jie S Hu, Mary Jane Irwin, Mahmut Kandemir, and Vijaykrishnan Narayanan. Leakage current: Moore's law meets static power. *computer*, 36(12):68–75, 2003.

[119] Indranil De, Deepak Johri, Anadi Srivastava, and CM Osburn. Impact of gate workfunction on device performance at the 50 nm technology node. *Solid-State Electronics*, 44(6):1077–1080, 2000.

[120] Nadine Collaert, An De Keersgieter, Abhisek Dixit, Isabelle Ferain, L-S Lai, Damien Lenoble, Abdelkarim Mercha, Axel Nackaerts, BJ Pawlak, Rita Rooyackers, et al. Multi-gate devices for the 32nm technology node and beyond. *Solid-State Electronics*, 52(9):1291–1296, 2008.

[121] Stefanos Kaxiras and Margaret Martonosi. Computer architecture techniques for power-efficiency. *Synthesis Lectures on Computer Architecture*, 3(1):1–207, 2008.

[122] Ultraview PCIEXT-64U Live Insertion PCI Bus Extender. PCI Smart Bus Extenders. http://ultraviewcorp.com/displayproduct.php?cat_id=4. 2013. (accessed on 1 August 2016).

[123] National Instruments USB-6251 Screw Terminal. Legacy USB DAQ Device. http://sine.ni.com/nips/cds/view/p/lang/en/nid/202802. 2013. (accessed on 1 August 2016).

[124] G. Covington, G. Gibb, J. Lockwood, and N. Mckeown. A packet generator on the NetFPGA platform. In *Field Programmable Custom Computing Machines, 2009. FCCM'09. 17th IEEE Symposium on*, pages 235–238. IEEE, 2009.

[125] OWL+USB wireless electricity monitor. 2 Save Energy Ltd. http://www.theowl.com/index.php/energy-monitors/standalone-monitors/owl-usb/. 2013. (accessed on 1 August 2016).

[126] V. Sivaraman, A. Vishwanath, Z. Zhao, and C. Russell. Profiling per-packet and per-byte energy consumption in the NetFPGA Gigabit router. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 331–336. IEEE, 2011.

[127] Software Component Of NetFPGA (SCONE). Stanford University. http://github.com/netfpga/netfpga/wiki/sconewalkthrough. 2013. (accessed on 1 August 2016).

[128] ECONET Project Deliverable 2.2. Test Plant, Evaluation Criteria and Representative Test Case. 2010.

[129] Alfio Lombardo, Carla Panarello, Diego Reforgiato, and Giovanni Schembra. Power control and management in the NetFPGA Gigabit router. In *Future Network & Mobile Summit (FutureNetw), 2012*, pages 1–8. IEEE, 2012.

[130] Princeton University OIT Networking & Monitoring Services. http://www.net.princeton.edu/index.html. 2016. (accessed on 1 August 2016).

[131] Feng Guo, Olga Ormond, Martin Collier, and Xiaojun Wang. Power measurement of NetFPGA based router. In *Online Conference on Green Communications (GreenCom), 2012 IEEE*, pages 116–119. IEEE, 2012.

[132] Xing Zheng and Xiaojun Wang. Comparative study of power consumption of a NetFPGA-based forwarding node in publish–subscribe Internet routing. *Computer Communications*, 44:36–43, 2014.

[133] Raffaele Bolla, Tivadar Szemethy, Roberto Bruschi, Franco Davoli, Lorenzo Di Gregorio, Pasquale Donadio, Leonardo Fialho, Martin Collier, Alfio Lombardo, and Diego Reforgiato Recupero. The green abstraction layer: a standard power-management interface for next-generation network devices. *IEEE Internet Computing*, (2):82–86, 2013.