Enabling Experiments for Energy-Efficient Data Center Networks on OpenFlow-based Platform

Nguyen Huu Thanh, Pham Ngoc Nam, Thu-Huong Truong, Nguyen Tai Hung, Luong Kim Doanh
Hanoi University of Science and Technology
Hanoi, Vietnam
{thanhnh, pnnam-fet, huong.truong, hungtai-fet}@mail.hut.edu.vn

Rastin Pries
University of Würzburg
Würzburg, Germany
pries@informatik.uni-wuerzburg.de

Abstract—**Energy efficiency is becoming increasingly important in today's networking infrastructure, especially in data center networks. Recent surveys have shown that the energy utilized by computing and networking components in a data center considerably contributes to its operation costs. Therefore performance and energy issues are important for the design of large multi-tier data centers that can support multiple services. However, the design, analysis, and experiments of such a large and complex system often suffer from the lack of appropriate experimental environments. With this motivation, we present a platform for in-depth analysis of energy-aware data center networks. The platform is a combination of hardware testbed and emulation. Based on real traffic patterns measured in a data center and with the energy model obtained from realistic measurements on NetFPGA-based OpenFlow switches, the experimental platform is designed to capture details of energy consumed by all network components such as links, ports, and switches under different scenarios.**

Keywords-energy-efficient networking; data center networks; OpenFlow; testbed; Future Internet

I. INTRODUCTION

Cloud computing and other distributed data services in the recent years have become increasingly popular due to the evolution of data center and parallel computing paradigms. Besides the increase in computing capacity, huge amount of energy consumed by data centers as well as the consequent greenhouse gas emission have drawn special concerns from the community. Large volume of energy consumption considerably contributes to the current data center operational expenses and has negative environmental impacts. This leads to the fact that the design of high performance, energy-efficient data centers is becoming an important issue. The concept of "Green Data Center" focuses on different research issues, including designing smart cooling systems, migrating virtual machines across physical machines, optimizing energy consumption of servers and network components, etc. Among these issues, the networking research community has paid special interest in optimizing energy consumption of network and server components. However, the design and analysis of such large and complex energy-aware data center systems often suffer from the lack of experiment environments. In the article, we present a platform for in-depth analysis of energy-aware data center networks, which is under the framework of the project Reducing Energy Consumption in Data Center Networks based on Traffic Engineering (ECODANE[1]). The main contributions of our work are the following:

- The test platform can be used to measure and analyze detailed energy consumption on different parts of a data center network, including switches, links, ports.
- Traffic load at different links and ports can be captured and monitored. Thus the relationship between traffic load and energy-consumption can easily be investigated.
- We extend the OpenFlow protocol and Controller to support traffic and energy-aware functionalities.
- The deployment of OpenFlow offers new possibilities to deploy and analyze different energy-aware topology optimization and routing algorithms.

The rest of the article is organized as follows. Section II addresses the requirements for energy-efficient networking experiments and some related works. In Section III, we analyze traffic data-sets measured in a data center and its modeling. Section IV discusses the hardware energy measurement and modeling NetFPGA-based OpenFlow switch. Based on the energy and traffic modeling in the previous sections, Section V addresses our testbed architecture and its energy-aware functionalities as well as some performance studies. Section VI concludes our work.

II. BUILDING INFRASTUCTURE FOR ENERGY-EFFICIENT NETWORKING EXPERIMENTS

A. Requirements

Providing an experimental infrastructure for the research of energy-efficient data center networks is challenging. Before building such an experimental infrastructure, the following issues should be considered: (1) How the energy consumption of different network components under varying conditions can be modeled and deployed in the test environment; (2) how realistic traffic patterns can be applied; and (3) which factors have an impact on the energy consumption of a network (e.g., topology, traffic load, clock frequencies, etc.) and how these relations can be investigated. Considering these issues will lead

---

[1] ECODANE is a research project co-sponsored by the Ministry of Science and Technology (Vietnam) and the Federal Ministry of Education and Research (Germany)

978-1-4673-2493-9/12/$31.00 ©2012 IEEE

to the following requirements of experimental infrastructures for energy efficient data center networks:

- *Scalability*: the test infrastructure should scale well to relatively large and complex data center scenarios.
- *Flexibility*: the test infrastructure should be able to flexibly apply different energy models of switches, routers, network interface, etc. On the other hand, various routing paradigms, optimization algorithms, and topologies should easily be integrated, managed, and investigated.

B. Related Works

There are some approaches for the analysis and performance evaluation of a system, namely *simulation*, *emulation* (or *virtual testbed*), and *hardware testbed*. With regards to the Green Data Center, there are several works published recently. GreenCloud [2] is a simulator for energy-aware cloud computing designed to capture details of the energy consumed by data center components. Another simulation tool is the CloudSim Toolkit [3], which offers a wide range of possibilities in modeling, simulation, and experimentation of a cloud computing infrastructure. With regards to emulation or testbed implementation, MDCSim [1] and the OpenFlow-based testbed shown in [4] can be used for data center and cloud computing infrastructure setups. Nevertheless the works in [1, 3, 4] are not specifically designed for energy-aware experimental purposes.

C. The ECODANE Experimental Infrastructure

To the best of our knowledge, there are until now only few platforms that are able to perform experiments for energy-efficient data center networks. In the project, we opt to build the experimental platform based on the combination of hardware testbed and emulation due to some of its advantages:

- In comparison to a pure hardware testbed, emulation environments are more scalable and flexible in the sense that a complex network with a large number of nodes and different network scenarios can be tested.
- In contrast to simulation tools, the platform makes use of such real network components as traffic generators or the OpenFlow Controller. Thus, we are able to develop new energy-aware functionalities based on current network protocols and components and investigate their performance under realistic conditions.
- In our approach, a NetFPGA OpenFlow switch is used to measure the energy consumption under different traffic loads, clock frequencies, etc. The realistic energy-model inferred from the hardware measurement results is deployed later on in the emulation part of the testbed.

Figure 1 shows the concept of our testbed. The testbed is based on OpenFlow [6], a technology that separates the control plane and data plane, thus offers a new flexibility to deploy and investigate new energy-aware routing and topology optimization paradigms.

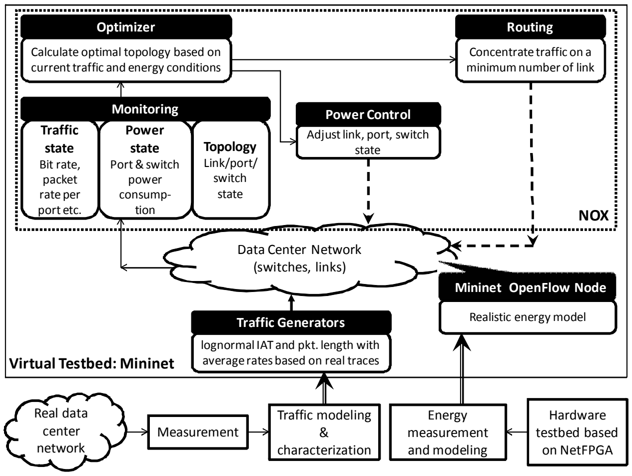


Figure 1. Testbed concept

To quantify the performance of the data center network under realistic traffic patterns, we firstly measure and analyze data sets from the data center of Viettel Corp., a large telecom provider in Vietnam. We then model the traffic patterns and deploy the models in the testbed. Details of traffic measurement and modeling are described in Section III. Furthermore, we build a small hardware NetFPGA-based OpenFlow testbed for measurement and modeling of the energy consumption and investigate the impact of several factors on energy consumption, including traffic load and clock frequencies. Section IV discusses this issue in more detail. Our emulated, virtual testbed is built based on Mininet [10], which includes different network components, such as the traffic generators, the emulated OpenFlow switches, and the OpenFlow controller. Details of the emulated testbed infrastructure are described in Section V.

III. TRAFFIC MEASUREMENT AND MODELING

A. Measurement Scenarios

This section shows the analysis of data-sets from Viettel data center. Viettel is one of the largest telecom providers in Vietnam, who provides mobile, fixed telephone services as well as broadband Internet. Viettel data center offers web services as well as video on demand. The data center is interconnected with the NGN core router which transmits IP-based converged services, including voice, video, and data. SNMP data for 6 days from 26th Oct 2011 (Wednesday) to 31st Oct 2011 (Monday) were collected. The data-sets were captured at the edge layer, at which Alpine 3804 switches from Extreme Networks are deployed. It is worth mentioning that only traffic utilization in long time scale 5-minute intervals categorized into web and video were allowed to be captured. In the paper, we denote *receiving traffic* as the traffic coming to the router and *transmitting traffic* is the traffic going from the router. Network topology of the Data Center was not known.

We study data collected at the GM-4Xi module of the router. GM-4Xi has four Gigabit Ethernet ports, using standard Gigabit Interface Connectors (GBICs). Our data includes web traffic, video traffic, and aggregation traffic. *Web traffic* and *video traffic* are the traffic that exchange data between the application servers and the edge switch; *aggregation traffic* is the traffic that exchanges data between the edge switch and the aggregation switch. We aim at answering the following questions: (1) what is the utilization of links of different applications in a data center; (2) how do traffic volumes vary over time; and (3) how do the distribution of traffic rates and the distribution of link utilization vary over time.

B. *Measurement Results*

1) Time-of-day traffic dependencies

We first examine the link utilization of receiving video traffic. Figure 2 shows how the link utilization of receive video traffic changes within the data center from 26th Oct 2011 to 31st Oct 2011. We observe that the traffic shows a little variation over a one week period. For example, the traffic always decreases from 0:00 to 5:00 of all days in a week, then increases from 5:00 to 11:00 etc. Therefore, we examine the distribution of link utilization of receiving video traffic in 5 time periods of a day, which have the same traffic utilization: 0:00-5:00; 5:00-11:00; 11:00-16:00; 16:00-20:00 and 20:00-24:00. For each period, we collected data-points in all days of a week and then studied the best-fitted distribution.

TABLE I. DISTRIBUTION FOR EACH PERIOD TIME-OF-DAY – RECEIVING VIDEO UTILIZATION

Time-of-day	Distribution of receiving video utilization	Mean of util. (%)	Standard deviation (%)
0:00 – 5:00	Logistic	19,83339	9,46915
5:00 – 11:00	DoubleBoltzman	31,2786	17,3518
11:00 – 16:00	Logistic	53,91224	6,94337
16:00 – 20:00	Logistic	49,25308	7,88609
20:00 – 24:00	Logistic	50,75055	7,47359

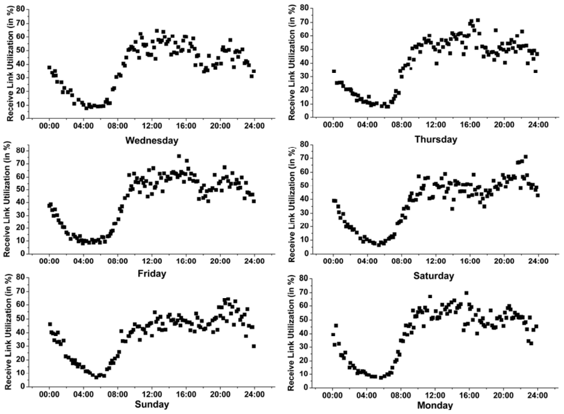


Figure 2. Utilization of receiving video traffic from 26th Oct – 31st Oct 2011

Table I represents the statistic distribution of these periods with the corresponding mean utilization and standard deviation from mean values. It is observed that the volume of transmitting video is very low, thus the measurement results of transmitting will be neglected in the paper.

Finally, we present the distribution fitting results for the average rates of receiving web traffic and the distribution fitting results for the link utilization of receiving and transmiting of aggregation traffic, which are shown in Table II, III, and IV, respectively.

TABLE II. TIME-OF-DAY DISTRIBUTION OF RECEIVING WEB TRAFFIC

Time-of-day	Distribution of receiving web utilization	Mean of util. (bps)	Standard deviation (bps)
0:00 – 8:00	LogNormal	530460, 41967	219447,77516
8:00 – 16:00	ExpAssoc	2,06577E6	2,67416E6
16:00 – 24:00	BoltzIV	1,21705E6	582523,86268

TABLE III. TIME-OF-DAY DISTRIBUTION OF TRANSMITTING AGGREGATION TRAFFIC UTILIZATION

Time-of-day	Distribution of receiving video utilization	Mean of util. (%)	Standard deviation (%)
0:00 – 5:00	Logistic	12,37866	7,00438
5:00 – 11:00	DoubleBoltzman	19,23621	10,0563
11:00 – 16:00	Logistic	31,52311	4,50159
16:00 – 20:00	Logistic	28,54375	5,59683
20:00 – 24:00	Logistic	32,34108	4,73992

TABLE IV. TIME-OF-DAY DISTRIBUTION OF RECEIVING AGGREGATION TRAFFIC UTILIZATION

Time-of-day	Distribution of receiving video utilization	Mean of util. (%)	Standard deviation (%)
0:00 – 5:00	Logistic	6,96035	3,36795
5:00 – 11:00	DoubleBoltzman	19,74767	10,06409
11:00 – 16:00	Logistic	27,85558	4,96727
16:00 – 20:00	Logistic	24,7109	5,12353
20:00 – 24:00	Logistic	21,66302	3,90391

2) Day-of-week traffic dependencies

In this subsection, we focus on the distribution of traffic in different days of a week. We follow a similar analysis method as in the previous subsection. Once again, we use the data collected from one of the application servers for each run. As expected, the utilization of web and video traffic is the lowest on Sunday and Saturday, while it reaches the peak in the middle of the week. Due to the space limitation, we will not show statistic results for traffic of other applications.

TABLE V. DAY-OF- WEEK DISTRIBUTION OF RECEIVING WEB UTILIZATION

Day-of-week	Dist. of receiving web util.	Mean util. (in bps)	Standard (in bps)
Wednesday, 26 Oct	Gaussian	1,56173E6	1,84992E6
Thursday , 27 Oct	Exponential	770597,76181	1,07373E6
Friday, 28 Oct	LogNormal	676079,30612	926034,40581
Saturday, 29 Oct	Sweibull2	399764,64566	229524,21542
Sunday, 30 Oct	Slogistic1	352617,22113	131445,92936
Monday, 31 Oct	Extreme	1,93749E6	2,21991E6

IV. HARDWARE-BASED ENERGY MEASUREMENT AND MODELING

In our hardware testbed, NetFPGA based OpenFlow switches developed by Stanford [7] are used. The NetFPGA consists of a programmable Xilinx Virtex II- Pro FPGA based core with four Gigabit Ethernet interfaces as shown in Figure 3. The NetFPGA card is connected to a host computer which plays the role of the control part of the OpenFlow switch via a PCI interface. The function of the NetFPGA card is to handle all the packet processing tasks in hardware like commercial routers [17].

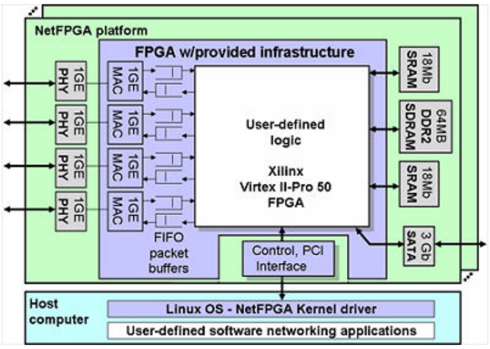


Figure 3. The NetFPGA platform

A. *Energy Measurement based on NetFPGA*

To obtain the energy model for the NetFPGA, we have done a number of energy measurements using Xilinx power estimation tool Xpower. The Verilog source code of the reference OpenFlow switch from [17] was synthesized by the Xilinx ISE 10.1 tool to obtain .ncd and .pcf files. These files were then input to the Xpower tool to obtain the power estimation. It should be noted that we only estimate the power consumption by the FPGA chip with the Xpower tool. In our experiments, we investigated the change in power consumption of the FPGA chip when the working frequency of the switch changes and when the data rate changes. Figure 4 shows the power consumption of the FPGA chip as a function of working clock frequency. As can be seen from the figure, the power consumption of the switch can be reduced significantly by reducing the working clock frequency. To see how the power consumption of the FPGA chip changes as function of the data rate, we used a script to generate different traffic loads for the switch and input to the Xpower analyzer to obtain the results as shown in Figure 5. It is shown that the power consumption increases very slowly as the network traffic increases, which is in-line with observations made by other researchers in [8], [9].

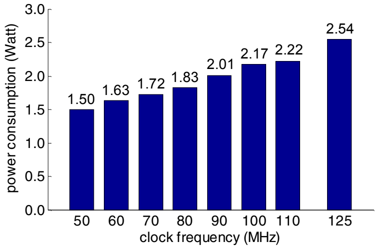


Figure 4. Power consumption (W) versus clock frequency (MHz)

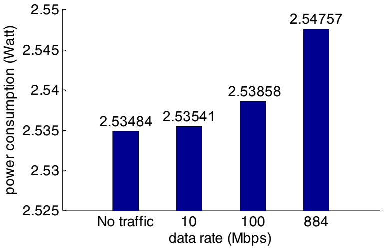


Figure 5. Power consumption (W) versus data rate

B. *Energy Modeling*

From our energy measurement results presented in Section 4.A and from the results in [9], we can model the power consumption of a NetFPGA card as follows:

$$P(f, r) = P_{co} + P_{FPGA}(f, r) + KP_E + P_P(r) \qquad (1)$$

where $P(f,r)$ is the power consumption of NetFPGA dependent of working clock frequency f and data rate r. P_{co} is a constant base-line power consumption of all electronic components except the FPGA chip on the NetFPGA card (without any Ethernet ports connected). $P_{FPGA}(f, r)$ is the power consumed by the FPGA chip at working clock frequency f and data rate r. $K \in [0; 4]$ is the number of connected Ethernet ports. P_E is the power consumed by each Ethernet port (without any traffic flowing). P_E=1.102 W [9]. $P_P(r)$ is the power needed to process packets at a data rate of r (this component is about 4% of the total power consumption [9])

V. EMULATED NETWORK INFRASTRUCTURE

A. *The Emulation Infrastructure*

An emulation environment was developed to test our ideas of controlling the system in favor of energy saving. The emulator is called Mininet [10] and provides a simple and inexpensive network testbed for developing OpenFlow applications. Mininet is used as the emulation tool, which is able to emulate a real network with switches, servers, and links. The emulation platform allows turning links, switches, and servers on and off. The benefit of Mininet is that the code developed and tested under Mininet, for an OpenFlow controller, modified switch, or host, *can move to a real system with no changes*, for real-world testing, performance evaluation, and deployment.

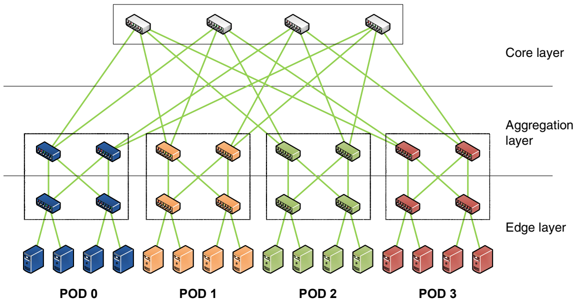


Figure 6. The Fat-Tree data center architecture

For our data center management, we use the Elastic-Tree network [11] which is based on the Fat-Tree topology as shown in Figure 6. The Fat-Tree architecture [12], [13] was developed to reduce the oversubscription ratio and to remove the single point of failures of the hierarchical architecture. The Elastic-Tree was proposed for dynamically adapting the energy consumption of a data center network, i.e., its network topology adjusts to the traffic requirements. As similar switches are used on all layers of the architecture, the cost for setting up a Fat-Tree data center can be kept low. The architecture is not achieving complete 1:1 oversubscription in reality, but offers rearangeable non-blocking paths with full bandwidth. The figure shows a 4-ary fat-tree which is build up of k=4 Performance Optimized Data Centers (PODs), each containing two layers of $k/2$ switches. The switches in the edge layer are connected to $k/2$ servers and the remaining ports of the edge switches are connected to the aggregation layer as shown in Figure 6. The core layer consists of $(k/2)^2$ k-port core switches where each of them is connected to each of the k PODs [12]. A Fat-Tree data center architecture built with k-port switches supports $k^3/4$ servers. In our emulation, we developed some logical modules, which are described in the followings and are shown in Figure 1.

1) *Data center network*

In our data center context, the network is emulated with the Fat-Tree topology. By using Mininet, the topology can scale up to a large number of nodes. In our experiment configurations, the number of ports of each switch k=4 or 6 is normally used, corresponding to the number of servers (16 or 54) as shown in Figure 6. Virtual OpenFlow-capable nodes are already integrated in Mininet. In the emulation, we extend the functions of the node to mimic behaviors of a real switch, such as the shutdown or start-up latency.

2) *NOX controller*

Figure 1 depicts the developed NOX controller with 3 main functional modules: *Optimizer*, *Power Control*, and *Routing*. NOX is an open source OpenFlow Controller that is used to control OpenFlow switches [14].

a) *Optimizer*

The Optimizer's role is to find the minimum power network subset (minimum numbers of switches and links) that satisfies current traffic conditions, while still offering good Quality of Service (QoS). The module is developed using the NOX controller [14], being able to provide network traffic statistics via OpenFlow messages. Its needed inputs are a network topology, a traffic matrix, a power model for each switch, and desired fault tolerance properties. Traffic statistics are gathered with the port-counter field of the OpenFlow switch. Fault tolerance is handled with defining spare switches or spare capacity for a link. In our testcase, a bandwidth threshold of traffic transmitted on a link is set to 70% to spare 30% capacity of the link for safety margin reason. Topology-aware heuristics [11] are developed for the Optimizer module to compute a set of active components (switches, ports, and links). Every 500ms, the Optimizer outputs a subset to the Power Control and Routing modules. Based on our proposed power model (Section IV) of each commercial switch, port and link, the energy reduction in the whole network is estimated as numbers of switches, ports, and links can be turn off or be put into sleep mode.

b) *Power Control*

The power model derived from the hardware NetFPGA-based OpenFlow switch (see Section IV) is deployed in the testbed and the Power Control module of the NOX. The module toggles the power states of ports, linecards, and entire switches through OpenFlow messages and Python APIs of Mininet to "tell" switches "off or on" or change to an appropriate power saving mode (e.g., changing clock frequencies).

c) *Routing*

The module is in charge of optimizing the routes in the data center. It is implemented in the NOX controller as a NOX module. Besides some routing mechanisms supported by the current NOX such as Dynamic All Pairs Shortest Path, Spanning Tree, a hierarchical load-balancing routing algorithm is, in our implementation, selected to guarantee the QoS requirements. The load-balancing task is hierarchically decentralized into sub-trees for different traffic classes. The algorithm computes a minimum-load path for all packets of the same flow. Following this criterion, load distribution is in balance all over the network; that in turn increases the reliability of a data center. Moreover, the hierarchical model is used to reduce the traffic passing in the network.

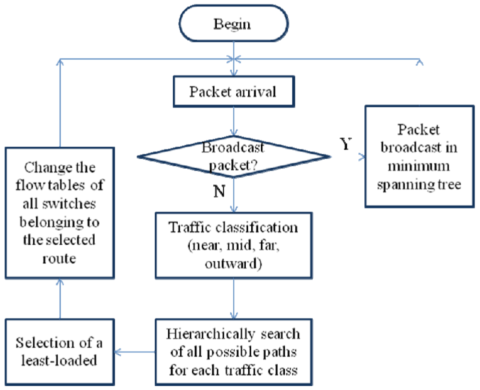


Figure 7. Traffic balancing algorithm to improve reliability

The Routing module as shown in Figure 7 is implemented separately from the Optimizer. It fetches the Optimizer's outputs for its own routing calculation. However, if the Optimizer is, by accident, out of order, all network components are toggled to operating mode.

3) *Traffic generator*

A software traffic generator is developed based on D-ITG [16] to generate network traffic from servers within a data center network. In addition to conventional traffic patterns that can be offered by D-ITG, our traffic generator is able to send traffic with common patterns in data centers [15]. Our extensions are as follows:

- *Short-time scale pattern*: the traffic pattern in short-time scale is emulated in lognormal distribution as gathered from Benson et al. [15].
- *Long*-time *scale pattern*: the min, max, and average rates of the traffic follows the rates measured in the Viettel traffic analysis as shown in Section III.

B. *Case Studies*

There are 4 traffic generation study cases called: *Near Traffic*, *Far Traffic*, *Middle Traffic*, and *Mixed Traffic* generation. As shown in Figure 8, Near Traffic generation is generating traffic that destines to servers of the same edge switch. While Middle Traffic generation is creating traffic that runs among servers of the same POD but not of the same edge switch. Far Traffic generation therefore is a context for traffic that travels between pairs of servers of 2 different PODs.

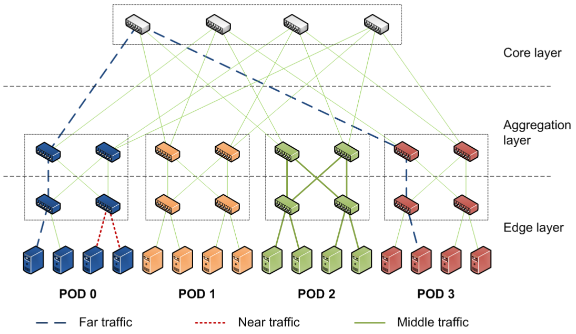
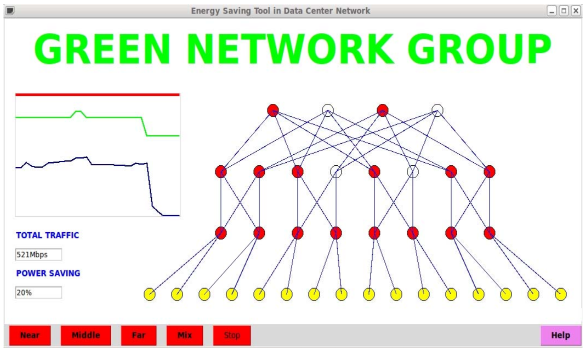


Figure 8. Near, Middle, and Far Traffic

In the mixed traffic type, each server sends its traffic to servers both of the same POD and different PODs in ratio of 9:1 accordingly. In order to check the NOX performance, we send traffic in 60 second intervals, in which intra-POD traffic is sent in the first 10 seconds, and then comes inter-POD traffic. In the duration of intra-POD traffic, many aggregation switches were observed ON while all core switches were still OFF. When the Inter-POD traffic is present in the network, the numbers of ON core switches increases.



TOTAL TRAFFIC 521Mbps

POWER SAVING 20%

Figure 9. Demonstration of Energy Reduction by turning off uneeded switches in a data center network

Figure 9 shows the GUI of our emulation. Some first results gathered with k=4 in ECODANE show the power consumption goes asymptotically with the traffic volume, which results in energy saving of between 25% and 40%, depending whether most of the traffic is transmitted locally within a rack (Near Traffic), within a POD (intra-POD traffic – Middle Traffic), or globally transmitted within the whole data center (inter-POD traffic – Far Traffic). Results show that the Near-Traffic scenario consumes least energy, while the Far-Traffic scenario consumes most energy. Also, more energy can be saved if k increases. Table VI shows the measured results at traffic load 70%.

TABLE VI. AVERAGE POWER-SAVING RATE AT TRAFFIC LOAD 70%, k=4

	Average energy-saving rate (traffic load 70%, *k*=4)
Near traffic	35%
Middle traffic	15%
Far traffic	3%

VI. CONCLUSION AND FUTURE WORK

In this paper, we present a novel platform to facilitate experiments of energy-efficient data center networks. The platform makes use of realistic traffic pattern as well as energy models. OpenFlow is deployed in our testbed to accommodate flexible implementations of energy-aware network paradigms and functionalities. We believe that such a comprehensive experimental platform is essential for providing guidelines in the research, analysis and design of energy-aware data centers.

The next step is to build a real data center testbed with 10 NetFPGA-based OpenFlow switches (a half Fat-Tree). The purpose of the testbed is to develop mechanisms to stand by or wake up a NetFPGA OpenFlow switch and investigate its performance in terms of energy saving, latency, and stability.

REFERENCES

[1] Seung-Hwan Lim, Bikash Sharma, Gunwoo Nam, Eun Kyoung Kim, and Chita R. Das, “MDCSim: A Multi-tier Data Center Simulation Platform,” in the Proceedings of the IEEE International Conference on Cluster Computing and Workshop, 2009.

[2] Dzmitry Kliazovich, Pascal Bouvry, Yury Audzevich, and Samee Ullah Khan, “GreenCloud: A Packet-level Simulator of Energy-aware Cloud Computing Data Centers,” in the Proceedings of Globecom 2010.

[3] Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N. Calheiros, “Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities,” in the Proceedings of the International Conference on High Performance Computing and Simulation, 2009.

[4] Carlos A. B. Macapuna, Christian Esteve Rothenberg, and Mauricio F. Magalhaes, “In-packet Bloom Filter based Data Center Networking with Distributed OpenFlow Controller,” IEEE Globecom 2010.

[5] Rastin Pries, Michael Jarschel, and Sebastian Goll, “On the Usability of OpenFlow in Data Center Environments,” Workshop on Clouds, Networks and Data Centers colocated with IEEE ICC, 2012.

[6] http://www.openflow.org/

[7] “NetFPGA Gigabit Router”, www.netfpga.org

[8] Bolla R., Bruschi R. and Davoli F., “Energy-Aware Performance Optimization for Next-Generation Green Network Equipment”, Proc. of the 2nd Workshop on Programmable Routers for Extensible Services of Tomorrow, in conjuction with ACM SIGCOMM’09 (PRESTO09), Barcelona, Spain, Aug. 2009.

[9] V. Sivaraman, A. Vishwanath, Z. Zhao and C. Russell, “Profiling Per-Packet and Per-Byte Power Consumption in the NetFPGA Gigabit Router”, INFOCOM Workshop on Green Communications and Networking (GCN), 2010, Shanghai, pp.331-336.

[10] http://yuba.stanford.edu/foswiki/bin/view/OpenFlow/Mininet

[11] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, “Elastic tree: Saving energy in data center networks,” in 7th USENIX Symposium on Networked System Design and Implementation (NSDI), San Jose, CA, USA, April 2010.

[12] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” in SIGCOMM ’08: Proceedings of the ACM SIGCOMM 2008 conference on Data communication, Seattle, WA, USA, August 2008, pp. 63–74.

[13] R. N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, “Portland: a scalable fault-tolerant layer 2 data center network fabric,” SIGCOMM Comput. Commun. Rev.,vol. 39, no. 4, pp. 39–50, 2009.

[14] http://noxrepo.org/wp/

[15] T. Benson, A. Akella, and D. Maltz, “Network traffic characteristics of data centers in the wild,” in Internet Measurement Conference (IMC), Melbourne, Australia, November 2010.

[16] http://www.grid.unina.it/software/ITG/

[17] “NetFPGA Gigabit Router”, www.netfpga.org