

## **A Low Power Consumption Frequency Adaptation Mechanism Based on the Traffic and Implementation on NetFPGA**

Lei Zhou<sup>1</sup>, Lianghua Li<sup>1</sup>, Xiaoju Liu<sup>1</sup> and Xiaojun Wang<sup>2</sup>

<sup>1</sup>*Institute of Microelectronics CAD Research, Hangzhou Dianzi University, Hangzhou 310018, China*

<sup>2</sup>*The Prince Institute, School of Electronic Engineering, Dublin City University, Dublin 09, Ireland*

*yflilianghua@163.com, wangx@eeng.dcu.ie*

### **Abstract**

*Both in academic and industrial research, building a green energy-saving network has become a hot topic. Combined with hardware division and adaptive algorithm, this paper proposes an adaptive frequency control mechanism based on traffic to reduce power consumption. By calculating real-time traffic, the network device can adjust its operating frequency. On the premise of maintaining network performance, this mechanism could reduce the power consumption significantly. Besides, the adaptive frequency mechanism has been verified and realized in the NetFPGA Reference Router. Experimental data show that under different ranges of traffic loads and without affect its performance, the Router embedded this mechanism can reduce power consumption by more than 20%.*

**Keywords:** *NetFPGA, Energy Conservation, Frequency Adaptation, Adaptive, Rapid*

### **1. Introduction**

With the surveys by related research organization, there are about 3 billion network interface devices which are utilized for the global LAN in the Ethernet wired communication [1]. It is estimated that European Internet Service Provider (ISP) has consumed about 21.4 billion KWh electricity which is used for all the network equipments in 2010. Besides, by 2020, the consumption is predicted to dramatically increase to about 35.8 billion KWh if green energy-saving technology cannot be made use of [2]. Thus, both study in effective utilization of energy and research to reduce power consumption aiming at network equipment are becoming increasingly significant and necessary in the modern society which admires eco network construction.

The average utilization bandwidth of current internet backbone network reaches 30% and the highest average utilization of the bandwidth is less than 45% [3]. However, the current network devices are generally working in full load condition during a whole day to guarantee network traffic to circulate smoothly in unpredictable network environments, which causes great electricity waste. In response to this situation, two basic methods are proposed by network equipment providers and researchers to make the network devices operate at the appropriate working state. One is to make the network device to switch between operation and dormancy, and the other is to realize the adaptive adjustment on operating frequency based on a certain frequency control mechanism [3]. For the network data flow is difficult to predict, it may cause a great impact on the network performance when the network devices switch between sleeping and working modes. Fortunately, an efficient adaptive frequency adaptation

mechanism can solve this problem very well, thus realizing the effective combination between performance and power consumption.

The proposed adaptive frequency control mechanism based on traffic (AFCMBOT) is to use the network traffic as an indicator to adjust network equipment's operating frequency automatically. The verification of AFCMBOT selects the NETFPGA (Network Field Programmable Gate Array) development platform of Stanford University [4]. The whole NetFPGA development platform is composed of software and hardware. The part of hardware is made up of developed card and integrated engineering design which is programmed with the hardware description language. Software section includes the related software corresponding to engineering design and device driver based on the LINUX operating system. With the drivers in the PC, users can easily communicate between the host computer and hardware board.

Corresponds to the traffic, the frequency of network device can be switched to a proper pattern steadily and quickly by AFCMBOT, which not only reduces the power consumption of network device, but also ensures its performance. In this paper, main contributions and works are as follows:

- (1) Proposing the adaptive frequency mechanism based on traffic-detection and adopting the buffered traffic threshold to avoid the clock jitter;
- (2) The correct statistics on network traffic based on the HDL (hardware description language) to achieve the hardware acceleration;
- (3) Keeping a larger data throughput to ensure the Router performance while AFCMBOT is added into the Reference Router [4] hardware design<sup>1</sup>;
- (4) With building power test model in LabView platform and collecting power data by the National Instruments' Data Analysis Instruments, the Router embedded AFCMBOT can reduce power consumption by more than 20%.

The rest of the paper is organized as follows, in section 2 an overview of frequency adaption technology is provided followed by a detailed introduction of adaptive frequency control mechanism based on traffic in section 3. In section 4 the implementation of AFCMBOT on the NetFPGA Platform is presented. With added AFCMBOT, the power consumption and the packet delay of the Router is tested in a set of experiments in section 5. Finally section 6 discusses and concludes the implications of the frequency adaptation technology.

## 2. Overview of Frequency Adaption Technology

In view of the adaptive frequency adjustment of network devices in NetFPGA platform, there are also some related researches at home and abroad, mainly in two methods. One is based on the cache of current data stream to achieve the device's frequency adaptation [5], the other is controlled by the upper software configuration [6]. By these two methods, the effect of frequency adaptation can be achieved in theory, but there are still certain deficiencies in the actual application. Firstly, the frequency adaptation is configured by software which needs to communicate with the NetFPGA board, while those reference designs such as routers, network cards, etc, are all hardware-based implementation; the frequency dynamic adjustment can not be implemented fast and efficiently by the upper software. As well, it relates to data interchange with each other between the hardware and software so that to achieve frequency adaptation is

---

<sup>1</sup> In our paper, we called Router without AFCMBOT as Reference Router, and called Router added AFCMBOT as Router.

relatively complicated and slow. In practical tests and in theory, we found that the cache based adaptive frequency control also has some drawbacks. As we all known, at a specific frequency, the specific cache size can be used as an indicator of the device's working state, but when at different frequencies, there is no obvious meaning that the specific cache size is utilized to indicate the device operating status. Besides, this approach would result in the adjacent frequency switching repeatedly in actual application when the cache changes frequently, which would affect the stability of network device. Furthermore, in the reduplicative switching process, a large number of data packets are lost and it also needs some extra power consumption. Though some algorithm (dual-threshold or multi-threshold) can solve this problem to some extent, there are still some potential threats to the stability of the network.

### 3. Adaptive Frequency Control Mechanism Based on Traffic

In this section we discuss superiority of AFCMBOT and make a detailed introduction about its algorithms to control the frequency.

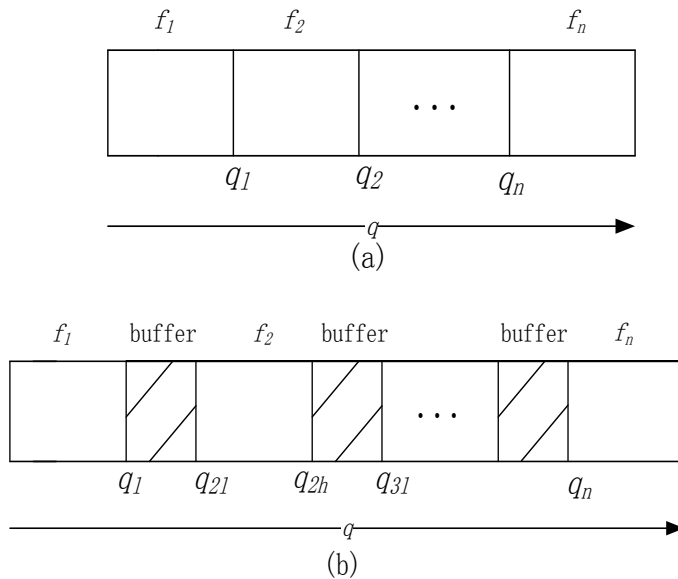
#### 3.1. Superiority of the AFCMBOT

In this paper, the AFCMBOT is aiming at the size of the network data stream to make frequency switch. The received packets of network device are calculated at a specific time (setting a specific and available time threshold value) under a constant frequency, and then the current network traffic is reflected based on the received packets' bytes in a given time. Setting the time threshold directly affects the sensitivity of the frequency control system, so the appropriate time threshold would greatly improve the stability, sensitivity of the device's performance when it is used as an indicator to switch the frequency. The size of the network traffic only depends on the real-time data coming from its ports, in that it is not limited by the frequency of network devices and independent with previous given time to calculate packets' bytes, which avoids both the defects of adopting cache to achieve frequency adaptation radically and the cumbersome and slow of software frequency configuration.

#### 3.2. Matching Mechanism between Traffic and Frequency of the AFCMBOT

The traffic load of the devices can be reflected by the size of the data streams which is detected by the front of the network device, so the range E of the data streams can be divided into several closed subintervals  $Q_1, Q_2, \dots, Q_n$ , where  $E = \bigcup_{k=1}^n (Q_k)$ ,

$Q_k = [q_k - q_{k-1}, t_0 + k \hat{1} N^+]$ ,  $q_k$  is the traffic rate, shown in Figure 1(a). These enclosed intervals are corresponding to the scope of the respective frequency ability to process data. It is emphasized that the devices will rapidly switch to the appropriate frequency adaptively as long as the value of the calculated traffic falls into the corresponding traffic interval. When it is necessary (considerable data stream changes), the devices' frequency can be switched to not only adjacent frequency, but also cross-frequency (directly switch to another frequency which is not adjacent).



**Figure 1. Buffered and Non-buffered Frequency Control Mechanisms**

However, it was found that this adaptive frequency switch can be achieved largely by this frequency adaptation mode in tests. But when the data stream size jitters just at the end of two subinterval, the frequency would switch between two adjacent frequencies to cause the frequency jitter and effect on the stability of the network devices. In order to avoid the shortcoming of using the closed threshold to make frequency adaptation, the buffered dual threshold to control mechanism is adopted into our design. As shown in Figure 1 (b). a suitable buffered space is left between the adjacent traffic intervals, which means  $\text{gap} T_k = q_{(k+1)l} - q_{(k)h} > 0$ . To point out, this kind of circumstance of jitter is different from the jitter which uses the cache to adapt frequency, whose frequency jitter is produced by the internal frequency adaptation mechanism; while AFBMCOT frequency judder is coming from the external network environment.

Compared to the previous moment, the trend of network data flow can be divided into two kinds of circumstances: increase and decrease, so there are two cases of frequency adaptation.

1. When  $q$  increases, the relation between the frequency and flow values as shown below;

$$F = \begin{cases} f_1, & q < q_{2l}; \\ f_k, & q_{kl} < q < q_{(k+1)l}; \\ f_n, & q > q_n; \end{cases} \quad ; \quad q = \int_0^m v dt \quad ; \quad (1)$$

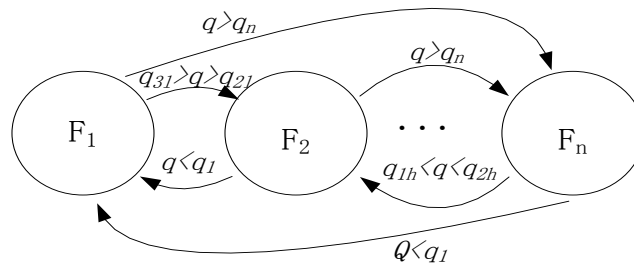
2. When  $q$  decreases, the relation between the frequency and flow values as shown below;

$$F = \begin{cases} f_1, & q < q_1; \\ f_k, & q_{(k-1)h} < q < q_{kh}; \\ f_n, & q > q_{(n-1)h}; \end{cases} \quad q = \int_0^m v dt; \quad (2)$$

Among them,  $F$  denotes the working frequency,  $m$  is the symbol of the time values for calculating traffic,  $f_k$  is the  $k_{th}$  set of frequency, and  $f_{k-1} < f < f_{k+1}$ ,  $k \in \mathbb{Z}^+$ ;  $q$  is the received traffic value,  $q_{kl}$  is  $k_{th}$  set smaller flow rate value,  $q_{kh}$  is the  $k_{th}$  set in a large flow rate value,  $q_{(k-1)h} < q_{kl} < q_{kh}$ ,  $n = k + 1$ .

The frequency adaptation state machine of dual threshold is shown in Figure 2. When the module is operating at the frequency  $f_k$  and the data traffic  $q$  grows gradually to more than  $q_{(k+1)l}$ , the operating frequency is switched to the high frequency mode  $f_{k+1}$  (if the received data stream is large enough to  $q_{(k+2)l}$ ,  $q_{(k+3)l}$  etc. in the given time  $m$ , the frequency can be directly switched to  $f_{k+2}$ ,  $f_{k+3}$ , until the maximum frequency  $f_{max}$ , which is called the cross-frequency switching); Similarly, when the module operates at frequencies  $f_k$  and data traffic is reduced from large to small below the  $q_{(k-1)h}$ , the frequency will be switched to the low frequency mode  $f_{k-1}$  (if the received data stream rapidly decreases to  $q_{(k-2)h}$ ,  $q_{(k-3)h}$ , etc. in  $m$ , the frequency can be directly switched to  $f_{k-2}$ ,  $f_{k-3}$ , until the minimum frequency  $f_{min}$ ).

When  $q_{(k-1)h} < q < q_{kl}$ , which means the traffic value falling into the buffer gap between two adjacent intervals, the device maintains at low frequency  $f_{k-1}$  if the current frequency is at the low frequency  $f_{k-1}$ . As well, if the current frequency is at the  $f_k$ , the device also maintained at  $f_k$ ,  $f_{k-1}$ , and  $f_k$  are adjacent each other. This effectively avoids frequency jitter when the network environment is unstable. Regardless of the current frequency  $f_{k-1}$  or  $f_k$ , frequency adaptation devices are able to work properly. It should be noted that when the device is operating at a high frequency  $f_k$ , it would bring a little higher energy consumption relative to  $f_{k-1}$ .



**Figure 2. Buttered Frequency Control State Transition Diagram of Dual Threshold**

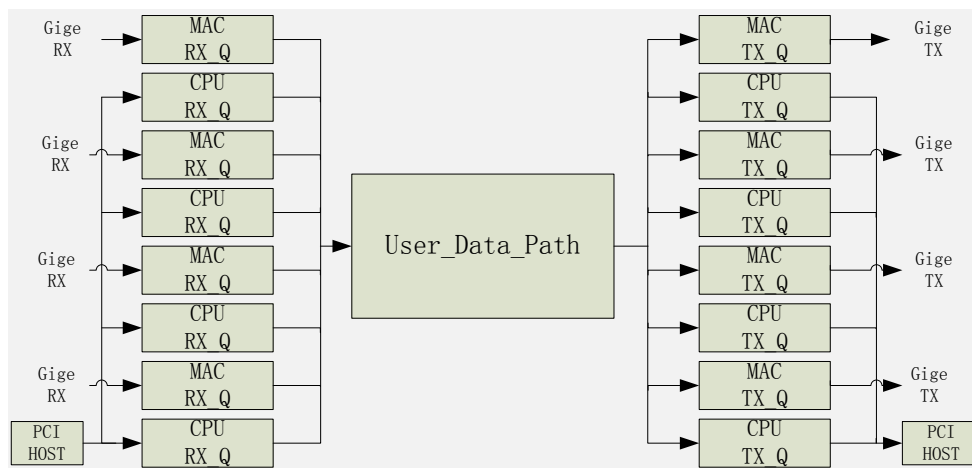
Selection of the dual threshold needs to consider two factors: 1. Data processing capability. Since different frequency to handle the data traffic capacity is not the same, and setting the individual frequency value reasonably is particularly important as to reduce power consumption. 2. The buffer gap between the adjacent subintervals cannot be large. Effectively reducing the frequency jitter and the power consumption are both considered. Since in most cases the data rate and frequency are one to one correspondence, the probability of  $q$  falling in the buffer gap is relatively low. So a combination of the above two factors, both in the energy consumption and the stability of frequency switching, device can achieve a good results.

#### 4. Implementation of AFCMBOT on the NetFPGA

With the emergence of NetFPGA, researchers can quickly build a prototype of a high-speed complex network and put up a real and reliable application in the NetFPGA platform to resolve emerging issues about network technology. New designed about web applications can be easily added to the network for testing, which not only simplifies the development process, but also shortens the development cycle [7]. This design embedded AFCMBOT in the Referenced Router project of the NetFPGA, aiming at achieving frequency dynamic adaptive adjustment and saving power consumption.

##### 4.1. The Realization Principle of Referenced Router

The realization principle of Reference Router is that hardware board gets packets from internet access and realizes the hardware-based routing look-up function. The software implements the complex routing protocols, which forms specific packets and sends out through different ports by the hardware card [8]. In addition, we can configure the route table, observe the traffic information on different ports and other functions through the GUI interface. The overall structure of Referenced Router is shown in Figure 3.



**Figure 3. Design of Reference Router's Pipelines**

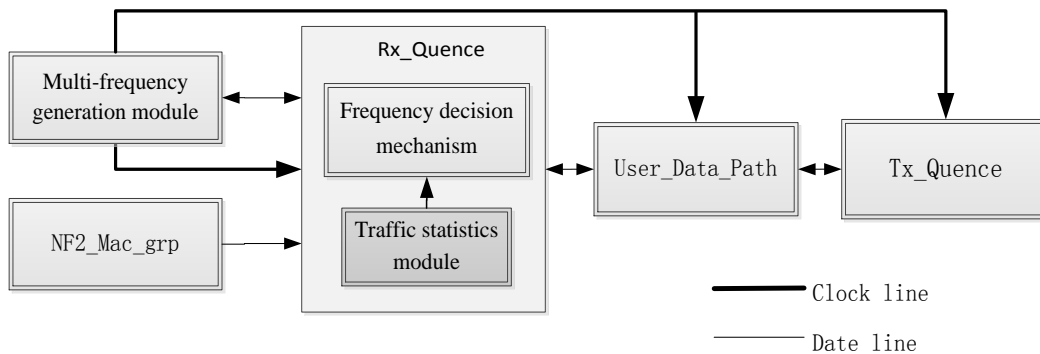
The NetFPGA Referenced Router can handle four Ethernet ports' data flow whose bandwidth is 4Gbps, and it also processes four CPU queue ports which communicate with the host computer. Ethernet ports communicate with the outside via Ethernet cable; and CPU port is connected with the PCI-e bus, which corresponds to the operating system kernel and the device driver for the board. The User\_Data\_Path(in the Reference Router

design) is the core part of the whole design, with rotation method to receive and send network packets as well as realize the routing function.

#### 4.2. The Implementation of AFCMBOT in Reference Router

To ensure AFCOMBOT can accurately calculate the current traffic, it must be embedded in the front of Reference Router. Based on this, we can ensure that the rate of traffic calculation will not be affected by the subsequent processing module.

In the Referenced Router project of this paper, the module to receive data flow queue (Rx\_Queue) was added with AFCOMBOT. Rx\_Queue module lies between the link processing module and the network processing module, which is used to receive input data flow from link layer and package data frame into packets. Then after a FIFO (First In First Out) buffer, packets are forwarded to the network module (User\_Data\_Path) for routing process.



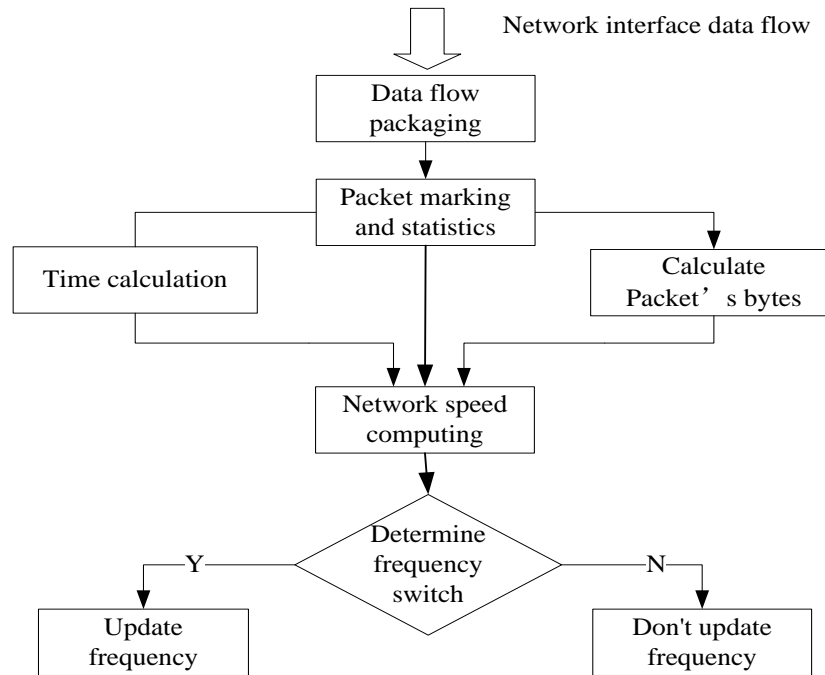
**Figure 4. The Architecture of AFCMBOT in the Referenced Router**

Figure 4 is the architecture implementation of the AFCMBOT in the Referenced Router. Firstly, the Traffic statistics module calculates the traffic of Router's input, and then select different frequency through the calculation results. Finally, the frequency is transmitted to each sub-module by the global clock lines.

For details, Frequency decision mechanism firstly sets the corresponding frequency adaptation flag signal on the basis of network traffic which is counted by the Traffic statistics module. Multi-frequency generation module is used to generate various frequencies and select an appropriate frequency according to the flag signal. And then it transmits the selected frequency to the Router's modules through the global clock lines. In our design, five clock frequencies can be selected in the multi-frequency generation module. To clearly distinguish the Router processing capability and power requirements of each frequency, the relationship among the five frequencies in this design is doubled, with the sizes of 7.813MHz, 15.625MHz, 31.25MHz, 62.5MHz and the highest frequency 125MHz [5].

Frequency decision mechanism possesses five corresponding flag signals, which are corresponding to select different frequencies. The Traffic statistics module is used for counting received data traffic, and the principle of traffic statistics module in Rx\_Queue is that when receives data frame, the data frame is encapsulated into a data packet and calculate the packet size, at the same time, a flag signal whose length is a clock cycle is generated for each packet, and then in a certain time  $m$ , flag signals and the bytes of each

packet are caught during the traffic rate calculation, the process of frequency adaptation is shown in Figure 5.



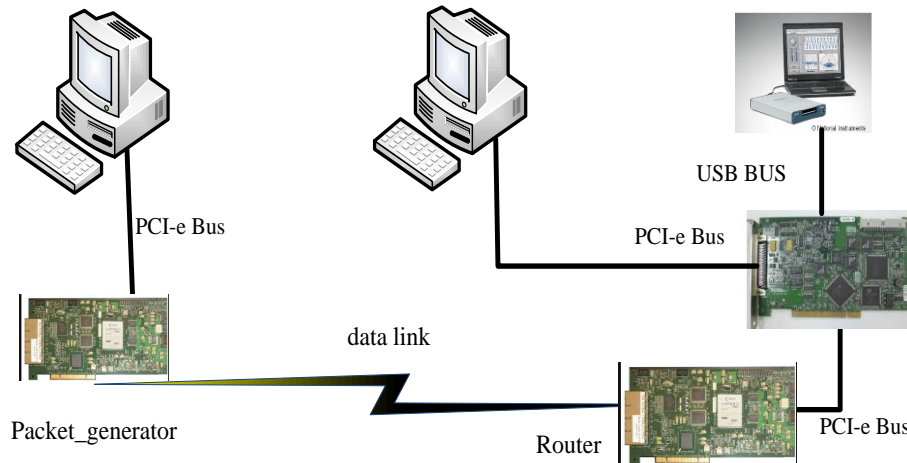
**Figure 5. AFCMBOT Processing Flow Chart**

Moreover, the clock which is used to calculate the data rate in the Traffic statistics module does not change in order to ensure the statistical accuracy. The same of the master clock of each function sub module ensures the operational stability of the Router and effectively restrains the various glitches which are caused by using multi-frequency in different function modules. Meanwhile, in order to observe the internal signal in the design, the corresponding registers should be added in the internal registers' pipeline, which would bring great convenience for debug and development. For example, working under different frequency, the Router's capacity to handle the size of the data flow (the estimate of  $q$ ) is obtained by adding a register in the design.

## 6. Experiment Test

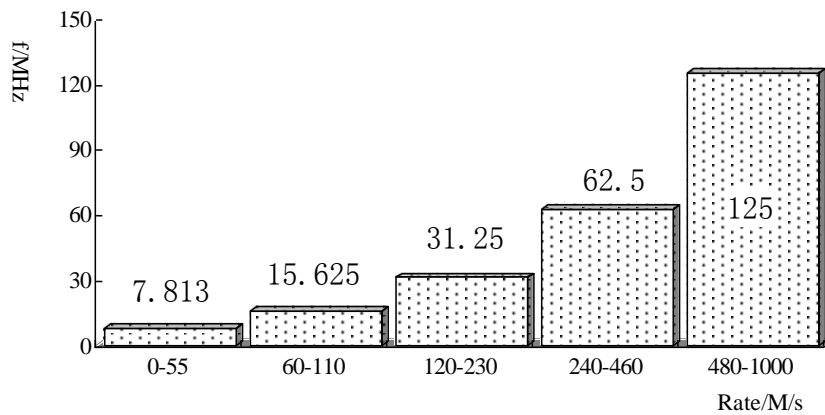
The method of our test is building with the Packet Generator [9] to send data streams at different rates to the Reference Router added AFCOMBOT. As shows in Figure 6, one computer was downloaded with project of Packet Generator, and the other is loaded with the project we propose in this paper[10]. Between the Router and the computer, a NI USB-6351 DAQ [11] was inserted to test the Router' power consumption through a PCI-e slot. We can analyze the power consumption of the Router from another computer which is connected the NI DAQ equipment through USB data cable. There is a certain time delay when the frequency switches. As well, it may cause a small amount of packets to lose during that time. So in the experiment, we should focus on the two indicators: 1.The impact on the performance of the Router with AFCMBOT, especially the influence of packets' delay to the Reference Router; 2.The maximum decreased value of the power consumption;





**Figure 6. Power Consumption Test Structures**

Figure 7 shows it can adaptively switch to the appropriate frequency when the Router receives different rate of data streams. Five rate ranges correspond to five frequency values respectively on the whole. In a short buffer gap between two adjacent rate ranges, such as gap between range 0-55 and range 60-110, the device would switch to different frequency according to the trends of the data stream, which is according to the formula (1) and formula(2). (The number of frequency values can depend on the network environment; here we chose five frequencies as a test template, and we set  $m = 1$  s).



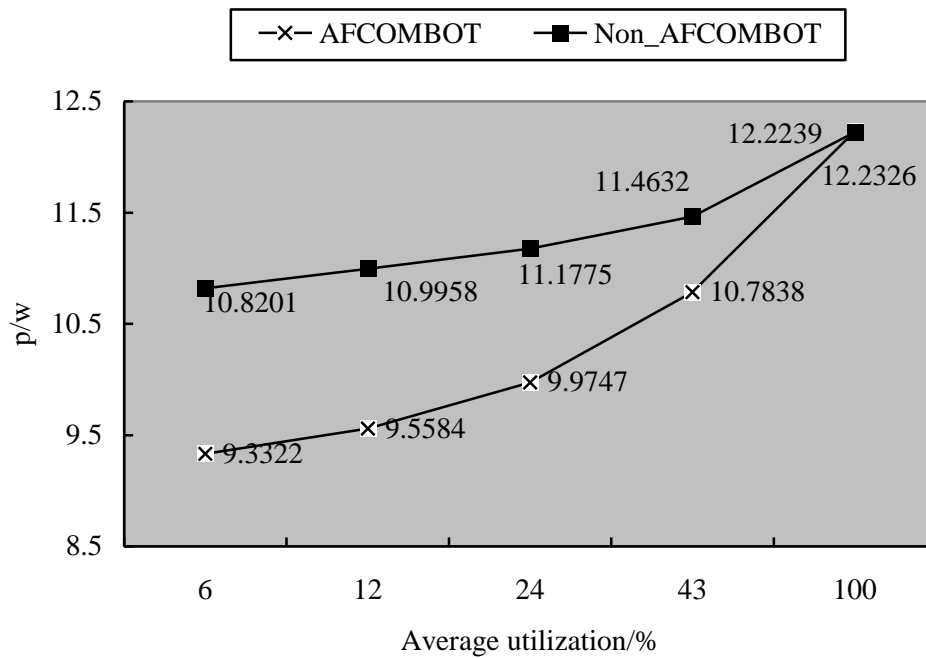
**Figure 7. Router's Frequency Corresponding to the Network Speed**

Power consumption test model is based on

$$p_a = C + f(r) \quad (3)$$

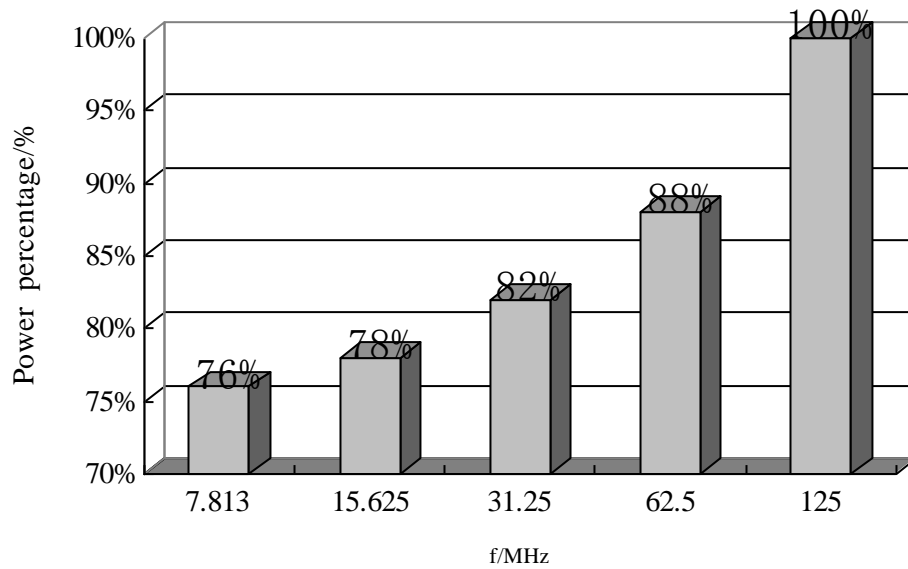
Intuitively,  $C$  can be viewed as that portion of power draw that cannot be eliminated through rate adaptation, while  $f(r)$  reflects the rate-dependent portion of energy consumption [11].

In our power consumption test model, the Router's power test is divided into two parts: power section with frequency change that corresponds to the  $f(r)$  in formula (3) and power section without frequency change corresponds to the  $C$  in formula (3). With different frequencies, corresponding power consumption is tested by NI USB-6351 DAQ and LIBVIEW software [11] based on the Graphical editing language. In order to reduce errors, 10 sets data in the same frequency were obtained respectively, then averaging. While the frequency is 7.813 MHz, the power of the Router is 9.332W, and when the frequency is at 125MHz, the frequency modulation Router's power is 12.2329W. Furthermore, even at the same frequency, the rate increase would also impact the power of the Router. Detailed data information is as shown in Figure 8.



**Figure 8. Different Average Utilization Corresponding to their Power Consumption**

As shown in the Figure 9, when the port traffic rate is less than 60Mbps, compared to the Router without AFCOMBOT, the power consumption can be reduced by 24%, which also includes the static power consumption value  $C$ .



**Figure 9. Various Frequencies and the Highest Frequencies of the Power Ratio**

The network traffic's latency would increase if the frequency of the Router lowers. Table 1 shows the number of packets that the Router transmits packets in 10 seconds. By comparison, the Router with AFCMBOT makes an acceptable difference to the Reference Router's performance in terms of packet delay and throughput.

**Table 1. Transmit Statistics of the Router**

Rate(Mbit/s)	Time(s)	Packets(AFCMBOT)	Packets(No AFCMBOT)
10	10	22769	22771
100	10	225267	227275
200	10	445975	445594
300	10	670469	668801
400	10	888272	867847
500	10	1084922	1138435
600	10	1252103	1343033
700	10	1476998	1537684
800	10	1683782	1759216

## 7. Conclusions and Discussion

Frequency adaptation (AFCMBOT) based on traffic computation can be applied to the actual network equipment to reduce power consumption. Front-end module in network devices can accurately calculate the current traffic so that it can adaptively adjust the frequency rapidly to reduce devices power consumption. Compared to other methods, this frequency adaptation can operate stably and feedback rapidly without frequency jitters. The experiments in the Referenced Router with AFCMBOT verified the operability and performance of the frequency control mechanism. More specifically, experimental data shows that it is necessary and important for the network device to possess an efficient adaptive frequency control mechanism. When the single port traffic on the referenced Router is less than 60Mbps, in comparison to the Reference Router whose frequency is 125MHz, its power can be reduced by approximately 24%. Besides, the smart AFCMBOT technology application prospect is also considerable and meaningful, by putting this technology into a common frequency synthesizer chip which can intelligently provide a driving frequency according to the load of network equipment (intelligent frequency synthesis or configuration, different from human configuration driving frequency). For hundreds of millions network equipment and the current promoting green, ecological network, the AFCMBOT technology is undoubtedly a good way to energy-saving.

## Acknowledgements

The work described in this paper was performed with the support of the ECONET project (low Energy Consumption Networks), funded by the EU through the FP7 call, and also gets support from and funded by the Microelectronic CAD Center of Hangzhou Dianzi University, Hangzhou, China. Besides, thanks the help from Dublin City University, Dublin, Ireland.

## References

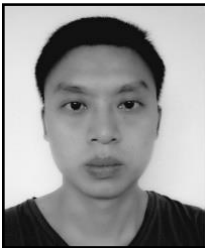
- [1] P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi and A. Maestro, "IEEE 802.3 az: the road to energy efficient Ethernet", IEEE Communications Magazine, Tampa, USA, (2010).
- [2] R. Bolla, F. Davoli, R. Bruschi, K. Christensen, F. Cucchietti and S. Singh, "The potential impact of green technologies in next-generation wireline networks: is there room for energy saving optimization?", IEEE Communications Magazine, Genoa, Italy, (2011).
- [3] M. Gupta and S. Singh, "Dynamic ethernet link shutdown for energy conservation on Ethernet link", Proceedings of the IEEE International Conference on Communications, (2007), pp. 6156-6161, Glasgow, England.
- [4] Stanford University, NetFPGA Homepage, <http://netfpga.org/index.htm> (2013).
- [5] W. Meng, Y. Wang, C. Hu, K. He, J. Li and B. Liu, "Greening the Internet using Multi-Frequency Scaling Scheme", Advanced Information Networking and Applications In 26th IEEE International Conference, (2012), pp. 928-935. Fukuoka, Japan.
- [6] F. Guo, O. Olga, L. Fialho and X. Wang, "Power consumption analysis of a NetFPGA based router", The Journal of China Universities of Posts and Telecommunications.(in press). Dublin, Ireland, (2012).
- [7] K.-K. Wang, B. Yang, T. Sun and Z.-X. Chen, "Application technology research on NetFPGA", SHANGDONG SCIENCE, Shang Dong, China, (2011), vol. 24, no. 5, pp. 93-98.
- [8] J. W. 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", Microelectronic Systems Education, IEEE Conference, (2008), pp. 160 – 161, California, USA.
- [9] G. A. Covington, G. Gibb, J. W. Lockwood and N. McKeown, "A Packet Generator on the NetFPGA Platform", Field Programmable Custom Computing Machines, 17th IEEE Symposium, (2009), pp. 235-238, California, USA.
- [10] A. Lombardo, D. Reforgiato, V. Riccobene and G. Schembra, "NetFPGA Hardware Modules for Input, Output and EWMA Bit-Rate Computation", International Journal of Future Generation Communication and Networking, June (2012).
- [11] National Instruments USB-6251 DAQ Device. <http://www.ni.com/labview/zhs>. (2014).

- [12] S. Nedeveschi, L. Popa, G. Iannaccone, S. Ratnasamy and D. Wetherall, "Reducing Network Energy Consumption", via Sleeping and Rate-Adaptation, (2008).

## Authors



**Lei Zhou**, he is a doctor of engineering has graduated from circuits and systems in Zhejiang University in June, 2009. He is currently the deputy director and associate research fellow of the Microelectronic CAD Center of Hangzhou Dianzi University. His main research direction is the internet of things technology, integrated circuit CAD technology and intelligent algorithm based on non-parametric statistics.



**Lianghua Li**, he is a graduate students in Microelectronic CAD Center of Hangzhou Dianzi University, research direction for computer systems and network equipment architecture and optimization.



**XiaojunbWang**, he received his BEng in Computer and Communications, MEng in Computer Applications, both from Beijing University of Posts and Telecommunications (BUPT), China, in 1984 and 1987 respectively. He received his PhD in Electronic Engineering from Staffordshire University, England, in 1993. He joined the School of Electronic Engineering in Dublin City University, Ireland, as a faculty member in 1992 where he is now a senior lecturer. He is interested in energy-efficient networking, information and network security.



**Xiaoju Liu**, she is a postgraduate student in Microelectronic CAD Center of Hangzhou Dianzi University, research direction for computer network optimization and power test model.

