Journal of Electrical and Electronic Engineering
2016; 4(3): 68-72
http://www.sciencepublishinggroup.com/j/jeee
doi: 10.11648/j.jeee.20160403.15
ISSN: 2329-1613 (Print); ISSN: 2329-1605 (Online)

Measure and Evaluate Delay Time for Wakeup on Lan (Wol) Method of OpenFlow Switch

Tran Hoang Vu

Danang College of Technology, The University of Danang, Danang, Vietnam

Email address:
thvu@dct.udn.vn

To cite this article:
Tran Hoang Vu. Measure and Evaluate Delay time for Wakeup on Lan (Wol) Method of OpenFlow Switch. *Journal of Electrical and Electronic Engineering*. Vol. 4, No. 3, 2016, pp. 68-72. doi: 10.11648/j.jeee.20160403.15

Received: May 18, 2016; **Accepted**: June 1, 2016; **Published**: June 17, 2016

Abstract: Currently, there are many ways to save energy for the switch such as change the operating modes to low power modes: *SLEEP PORT* and *SLEEP SWITCH* when there is no traffic flowing through the Ethernet ports. The switch will be waked up to normal modes when traffic flows pass through it. However, the changes of the modes of the OpenFlow Switch will create DELAY-TIME. If we know DELAY-TIME, then we will make routing algorithms and get appropriately mechanisms to ensure no impact on the transmission of data and loss packets. In this paper, we propose a remedy to measure and evaluate DELAY-TIME when we use the Wakeup On Lan (WOL) method for OpenFlow switch based on NetFPGA platform.

Keywords: OpenFlow Switch, NetFPGA, Wakeup On Lan, Delay Time, Data Center Network, Green Networking

1. Introduction

Power consumption in the ICT infrastructure is a pressing concern. Electricity used by data centers worldwide increased about 56% from 2005 to 2010, accounting for between 1.1% and 1.5% of total electricity use [1]. Many system components in the data center contribute to the overall power consumption, including servers, power, cooling, storage, networking equipment, etc. Nowadays, many network devices have tended to integrate additional functionalities to sleep when not being used and automatically wake up when receiving requests. By shutting down wireless network card of a handheld device when no internet action is being taken place, the battery lifetime can be increased. In this case, a network card is powered only when an incoming call is received. This is known as "wake-on-wireless" [2]. Moreover, IEEE 802.1 X Wake on LAN (WOL) [3] supporting feature allows dormant PCs to be powered up when the switch receives a specific Ethernet frame, known as the "magic packet" [4].

Following this trend, some power management methods for networking devices have been proposed to reduce network power consumption, can be generally divided into two categories: the sleep mode supporting devices [5, 6] and rate adaptation mechanism supporting devices [7, 8]. Sleeping scheme powers off idle devices or components into sleep states [6] for a pre-estimated duration, and wakes up the sleeping devices or components when new packets arrive. On the other hand, this scheme is fragile against burst traffic [8].

In paper [9], we propose methods for intelligently controlling the power consumption of OpenFlow switches used in data centers by adding SLEEP modes and WOL mechanism to the switches. However, the paper [9] has not considered the method to measure Delay Time when waking up OpenFlow Switch from SLEEP PORT mode and SLEEP SWITCH mode to normal modes. Therefore, in this paper, we present a solution to measure Delay Time for WOL method of OpenFlow Switch and build the test-bed system for our experiments.

The main contributions of our work are the following:

We propose the remedy to measure the delay time for WOL method of OpenFlow Switch.

We design the test-bed system measure and evaluate Delay time when we wake up OpenFlow Switch from SLEEP PORT mode (DTSP) and Sleep switch mode (DTSW).

The rest of the paper is organized as follows. Section 2 describes related works. Section 3 presents the measurement of delay time for WOL method. Section 4 describes test-bed system and experimental results. Conclusions are drawn in section 5.

2. Related Works

2.1. Sleep Modes for OpenFlow Switches

In [10] we designed a clock controller module integrated in the core of the NetFPGA based OpenFlow switch which can adjust the working clock frequency of the switch from 125 MHz down to 3.90625 MHz as in Fig. 1.

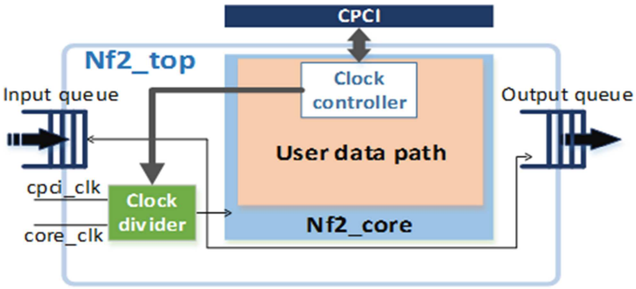

Figure 1. *Structure of the frequency divider.*

We could also change the state of any port of the switch to idle mode by modifying some bits in the control register in the Enthernet control chip.

From these results we define two new sleep modes as follows.

Sleep port mode: This mode is used to turn off one or more Ethernet port when there is no traffic flowing through the ports. In this mode, the switch is running at 125 MHz in order to maintain the operation of other Ethernet ports.

Sleep switch mode: This mode is activated only when there is not any traffic going through any Ethernet port. In this case, the switch does not need to process the data flow and can be running at the minimum clock frequency of 3.90625 MHz, while three out of the four Ethernet ports are completely turned off. In this mode, switch still maintains the operation of one Ethernet port at 10Mbps so that the switch can be waken-up immediately to normal operation by WOL method

2.2. WOL Module for OpenFlow Switches

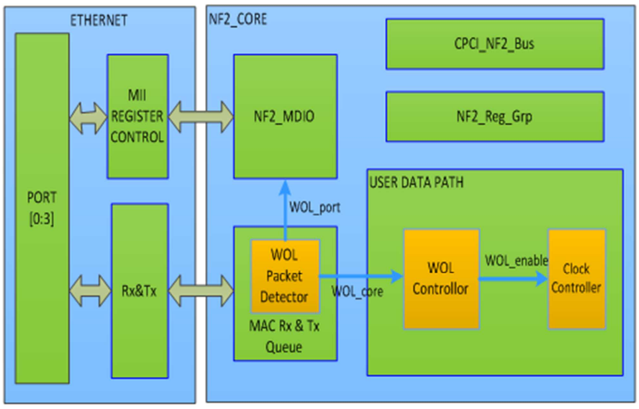

Figure 2. *WOL Packet Detector block.*

In [9], we designed a new WOL function for OpenFlow Switches as Fig. 2, which an Ethernet port or a switch can auto-wake up when they receive a broadcast packet (WOL packet) from other devices in the network.

In this method, two extended bytes are placed right after the 16 times repeats of the MAC address of the port that listens to the WOL packet. Those extended bytes for an OpenFlow Switch are used to wake up Ethernet ports or switches from a sleep mode.

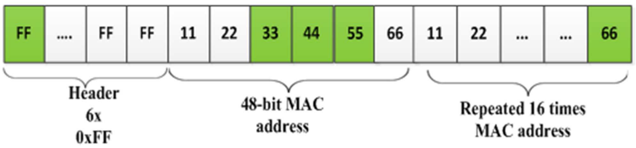

Figure 3. *The standard WOL packet.*

The port can operate at four different states includes off and three states of link rate 10Mpbs/100Mpbs/1Gbps. Therefore, each 2-bit group is used to control working modes of an Ethernet port. Besides, two extra bits are needed to change the operating frequency of the switch. Fig. 4 shows the bit descriptions of two extended bytes.

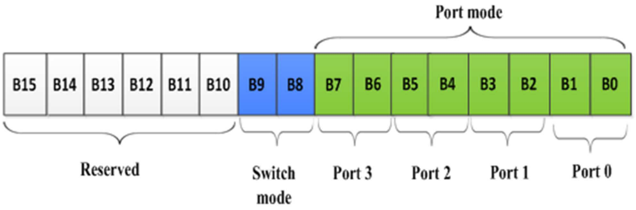

Figure 4. *Bit meanings of two extended bytes of WOL packet for OpenFlow Switches.*

When a WOL packet is sent to the switch by other devices on LAN network through the listening Ethernet port, it will be recognized over the bytes header. This packet is compared with the MAC address of the listening port. If it is the same, WOL packet is continuously compared with port mode and switch mode located at the extended bytes defined in part 2.1. WOL Packet Detector block will send WOL_port signal that corresponds to the port. This signal is connected to the NF 2_MDIO block that can change the status of the port.

Besides, WOL Packet Detector block also sends WOL_core signal to the WOL Controller block to control the Clock Controller block which is presented in paper [10]. This block will change the operating frequency of the switch to 3.90625 MHz or to 125MHz.

3. Delay-Time Measurement for WOL Method

As the definition of two new sleep modes in section 2, Delay-Time of waking up OpenFlow Switch from SLEEP PORT mode (DTSP) is delay time when we wake up the Ethernet port (DTWP).

$$DTSP = DTWP \tag{1}$$

Furthermore, Delay-Time of waking up OpenFlow Switch from Sleep switch mode (DTSW) is the maximum value of delay times of waking up the Ethernet port (DTWP), or changing the Ethernet port state from 10Mpbs to 1Gbps (DTCP) or switching the clock frequency from 3.90625 MHz to 125 MHz (DTSC).

$$DTSW = \text{Maximum}\ (DTWP, DTCP, DTSC) \qquad (2)$$

Delay-Time Measurement Method

Delay time of waking up the Ethernet port (DTWP) or changing the Ethernet port state from 10Mpbs to 1Gbps (DTCP) is defined from the time when the switch receives WOL packet to the time when data packets are transferred.

To verify the time when data packets are transferred through switches, we find out the signal gmii_rx_dv at rgmii_io block [11]. In [11] gmii_rx_dv (Reveive Data Valid) is driven by the PHY to indicate the PHY is presenting recovered and decoded data on the RXD (7:0). It is asserted during the entire data frame, and so provides an envelope signal for a valid data frame. When rx_dv = 1 the data frame pass through the Ethernet port, and rx_dv = 0 when port is turned off.

- Measure DTWP

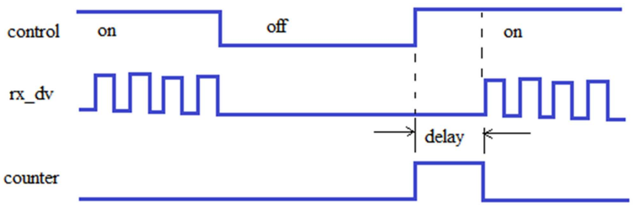

Figure 5. *Measure DTWP.*

To measure delay time of waking up the Ethernet port (DTWP), we build two counters, a counter is activated before waking up the Ethernet port, and a counter is activated after waking up the Ethernet port. DTWP is the average value of two counters. The counters will be stopped if the signal rx_dv = 1.

- Measure DTCP

To measure the delay time of changing the Ethernet port state from 10Mpbs to 1Gbps (DTCP), we also build a counter. However, rx_dv = 0 when the port is turned off while rx_dv = 1 at several times when it operates at 10Mpbs. Therefore, the parameter *-window* is defined with the condition as below:

$$\text{Maximum}\ (IPG) < window < DTCP \qquad (3)$$

In which:

IPG: interpacket gap is the time between transmissions of Ethernet packets.

When Ethernet port state is changed from 10Mpbs to 1Gbps, if IPG = window then a counter will be activated until rx_dv = 1.

Hence,

$$DTCP = window + counter \qquad (4)$$

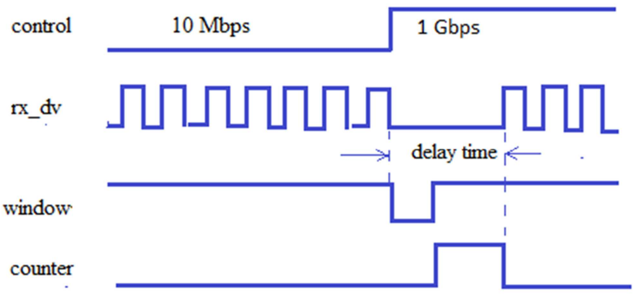

Figure 6. *Measure DTCP.*

To write and read the value of registers, we use two functions Read_Reg and Write_Reg as Table 1.

Table 1. *Read and Write function for registers.*

Function	Structure	Describe
READ_REG	int readReg (nf2 device *nf2, unsigned int addr, unsigned int *val)	Read value of register in addr address and save this value in val register.
WRITE_REG	int writeReg (nf2 device *nf2, unsigned int addr, unsigned int val)	Write value val to register at addr address.

4. Experimental Results

4.1. Test-Bed System for Measurement

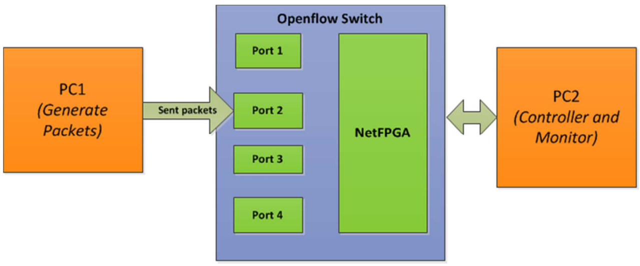

Figure 7. *Test-bed system for measurement.*

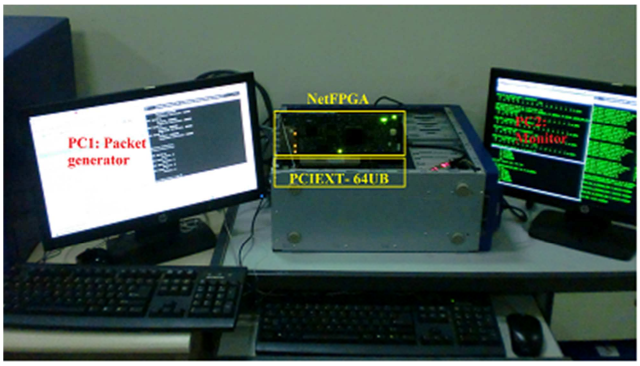

Figure 8. *Real test system for measurement.*

In order to measure DTWP and DTCP, we build a hardware test-bed as in Fig. 7.

- PC1: Generate packets and WOL packet to OpenFlow Switch.
- PC2: Control the change of Ethernet port state and measure the delay time.
- OpenFlow switch version 1.0.0.4 based on NetFPGA version 3.0.1 developed by Stanford [12] is used.

Table 2. *Configure the registers to measure DTWP.*

regwrite 0x440000 0x0900	**Turn off port 0**
regwrite 0x2001100 0xFF	Reset all counter
regwrite 0x2001100 0xAA	Enable Portx counter 0
regwrite 0x440000 0x1140	Wake up port 0
regwrite 0x2001100 0x00	Enable Portx counter 1 and 0
Read counter	
regwrite 0x2001104 0	Select Port 0 counter 0
regwrite 0x2001104 1	Select Port 0 counter 1

Table 3. *Configure the registers to measure DTCP.*

regwrite 0x2001100 0x0000F0FF	**Reset all counter**
regwrite 0x2001100 0xFFFF0000	Start counter with window filter = 65535 period
regwrite 0x440000 0x2100	Change to 1Gbps port 0
Read counter	
regwrite 0x2001104 8	Select Port 0 counter down time
regread 0x2001110	Read low DWORD
regread 0x2001114	Read high DWORD

4.2. Measuremental Results

- Measure DTWP

We measured 10 times when waking up the Ethernet port and the results are shown in the Table 4.

Table 4. *Measure DTWP.*

Time	Counter 1 (s)	Counter 2 (s)
1	1.7773	1.7761
2	1.8811	1.7793
3	1.8504	1.8789
4	1.7196	1.7279
5	1.7743	1.792
6	1.8605	1.8589
7	1.732	1.7309
8	1.8443	1.843
9	1.796	1.7959
10	1.7341	1.7327
Average	1.79696	1.79156
Average DTWP = 1.79426		

- Measure DTCP

In this experiment, we measured 5 times when changing the Ethernet port state from 10Mpbs to 1Gbps. The experimental results are shown in the Table 5.

Table 5. *Measure DTCP.*

Time	10Mps to 1Gbps
1	1.7503
2	1.7012
3	1.7242
4	1.7239
5	1.7406
Average DTCP	1.72804

- Estimate DTSC

As you can see in the Fig. 2, when the WOL packet Detector block detects the WOL packet, the WOL_core signal is sent to WOL Controller in order to generate the WOL_enable to Clock controller. This block will change the operating frequency of the switch to 3.90625 MHz or to 125MHz.

During this process, the delay time is only approximately 4 clock cycles, which is around 32ns (frequency 125MHz). Hence, DTSC is about 32 ns.

- Calculate DTSP and DTSW

From Table 4 and Table 5, delay time of waking up the Ethernet port -DTWP is approximately 1.8 second while delay time of changing the Ethernet port state from 10Mpbs to 1Gbps –DTCP is around 1.73 second.

Therefore, according to the formula (1) and (2), we have:
DTSP = DTWP = around 1.8 second
And DTSW = Maximum (DTWP, DTCP, DTSC)
= Maximum (1.8s, 1.73s, 32 ns) = 1.8 second

4.3. Error Evaluation

There are several errors in our experiment including:

- Errors due to writing registers. To reduce this error, we designed two counters to capture the delay time. This error is about 1.5 ms.
- Error due to counter: This error is around 1 clock, approximately 8 ns, which is very low.
- Error due to the time which rx_dv = 0. This error is the interpacket gap (IPG). To reduce this error, we increase the transmission speed from PC1 to the switch. In our experiment, we generate packets with the transmission speed is 234962 packets per second, about 4.25 us.

5. Conclusions and Future Work

In this paper, we proposed the solution to measure and evaluate delay time for WOL method of OpenFlow Switch at SLEEP PORT mode and SLEEP SWITCH mode. We also built the test-bed system and measured the delay time of waking up OpenFlow Switch from Sleep port mode (DTSP) and Sleep switch mode (DTSW). These delay times are approximately 1.8 second. The time is to configure PHY blocks as Setup Pll, Auto-MDIX or Auto Negotiation.

In the future, we will research and propose remedies to improve delay time for WOL method of OpenFlow Switch, our group will implement this solution to measure and evaluate delay time for OpenFlow Switch based on NetFPGA-10G [13].

References

[1] Growth in data center electricity use 2005 to 2010, http://www.analyticspress.com/datacenters.html

[2] M. Gupta, “A feasibility study for power management in lan switches,” in Proceedings of the 12th IEEE International Conference on Network Protocols, 2004.

[3] Eugene Shih, Paramvir Bahl, Michael J. Sinclair, "Wake on wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices."

[4] "IEEE 802. 1 X Wake in LAN Support" Authentication Services Configuration Guide Cisco IOS Release 15 MT.

[5] "Magic Packet Technology" White paper, combinations thereof are trademarks of Advanced Micro Devices.

[6] M. Gupta and S. Singh, "Dynamic ethernet link shutdown for energy conservation on ethernet links," in IEEE International Conference on Communications, 2007.

[7] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the energy consumption of ethernet with adaptive link rate (alr)," IEEE Transactions on Computers, vol. 57, pp. 448–461, 2008.

[8] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in Proc. of NSDI, Berkeley, CA, USA: USENIX Association, pp. 323–336, 2008.

[9] T. H. Vu, V. C. Luc, N. T. Quan, T. Thanh, N. H. Thanh, P. N. Nam "Design Sleep mode and Wakeup method for OpenFlow Switches" in Journal of Low Power Electronics, Vol. Number 3, pp. 347-353, September 2014.

[10] T. H. Vu, T. Thanh, V. Q. Trong, N. H. Thanh, P. N. Nam "Energy Saving for OpenFlow Switch on the NetFPGA platform Using Multi-Frequency" in International Journal of Computing and Network Technology, No. 1, pp. 9-15, 1 Jan 2014.

[11] Dongwook Kim "Gigabit Media Independent Interface – Gigabit Ethernet Nework" Feb, 28 2014.

[12] Glen Gibb, John W. Lockwood, Jad Naous, Paul Hartke, and Nick McKeown "NetFPGA – An Open Platform for Teaching How to Build Gigabit-rate Network Switches and Routers" IEEE Transactions on Education, 2008.

[13] http://netfpga.org/10G_specs.html