# LETTER Cycle Time Improvement of EtherCAT Networks with Embedded Linux-Based Master\*

Hyun-Chul YI<sup>†</sup>, Nonmember and Joon-Young CHOI<sup>†a)</sup>, Member

**SUMMARY** We improve the cycle time performance of EtherCAT networks with embedded Linux-based master by developing a Linux Ethernet driver optimized for EtherCAT operation. The Ethernet driver is developed to establish a direct interface between the master module and Ethernet controllers of embedded systems by removing the involvement of Linux network stack and the New API (NAPI) of standard Ethernet drivers. Consequently, it is achieved that the time-consuming memory copy operations are reduced and the process of EtherCAT frames is accelerated. In order to demonstrate the effect of the developed Ethernet driver, we set up EtherCAT networks composed of an embedded Linux-based master and commercial off-the-shelf slaves, and the experimental results confirm that the cycle time performance is significantly improved.

key words: EtherCAT, embedded Linux, EtherCAT master, cycle time

# 1. Introduction

Industrial Ethernet networks have been widely used for automation applications in place of traditional fieldbus networks because of their various benefits such as cost-effective implementation, flexible topology, high bandwidth, and wide compatibility [1]. Among the Industrial Ethernet protocols, EtherCAT is one of the prominent real time protocols and shows a desirable feature that each slave processes the addressed data *on the fly* as the frame moves downstream, which makes the theoretical maximum effective data rate even higher than 100 Mbps using the full duplex mode of Fast Ethernet, and enables a fast operation with cycle times even shorter than 100  $\mu$ s [2].

Nonetheless, faster performance is recently demanded by distributed real-time precision control applications such as multi-axis motion control systems including robots and computerized numerical control (CNC) machines, and power control systems including modular multi-level converter (MMC) and modular multi-level inverter (MMI) [3]. Moreover, open source master running on embedded Linux platform is also demanded because it is cost-effective for both hardware and software, space saving in factory, and efficient in performance.

\*This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2015R1D1A1A01056843).

a) E-mail: jyc@pusan.ac.kr (Corresponding author)

DOI: 10.1587/transinf.2018EDL8132

Fortunately, EtherLab is providing the IgH EtherCAT master, an open source EtherCAT master for Linux, which supports native EtherCAT-capable Ethernet drivers that is optimized for EtherCAT operation [4]. However, the support is provided only for several specific Ethernet controllers, and most of embedded processors must use standard Ethernet drivers for Linux, which leads to considerable degradation in cycle time performance.

In this letter, we investigate the cause of cycle time degradation when using standard Ethernet drivers for the IgH EtherCAT master. Then, we develop an EtherCATcapable Ethernet driver for Ethernet controllers of embedded systems so that the cycle time performance is improved by overcoming the causes of cycle time degradation. The validity of the developed driver is verified by extensive experiments of EtherCAT networks composed of IgH Ether-CAT master and commercial off-the-shelf slaves.

### 2. Analysis and Design

The software architecture of IgH EtherCAT master using standard Ethernet drivers is depicted on the left side of Fig. 1 as the original architecture, where we can find the generic Ethernet driver module between EtherCAT master module and Linux network stack [5]. This generic driver module enables arbitrary standard Linux Ethernet drivers and Ethernet controllers to be used for EtherCAT operation without any modifications. However, the generic driver module may cause a poor performance in cycle time for EtherCAT operation due to the following two main factors.

The first degradation factor is that the generic driver module and Linux network stack considerably give rise to unnecessary memory copy operations in processing the Ethernet frames. The second degradation factor is the New API (NAPI) for networking devices in the Linux kernel that is intended to reduce the overhead of packet receiving by deferring incoming frame processing until a sufficient amount of frames are piled up [6]. This approach is desirable for general network devices, but leads to considerable delay in processing of EtherCAT frames or even to a disruption of EtherCAT networks because each incoming frame to master must be processed within a preconfigured cycle time for the normal operation of EtherCAT networks.

In order to solve these problems, we design an architecture depicted on the right side of Fig. 1, where the EtherCAT master module directly interfaces to Ethernet controllers of embedded systems through a direct Ethernet driver. The di-

Manuscript received June 20, 2018.

Manuscript revised September 11, 2018.

Manuscript publicized October 11, 2018.

<sup>&</sup>lt;sup>†</sup>The authors are with the Department of Electronics Engineering, Pusan National University, 2 Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan, 46241, Rep. of Korea.



Fig. 1 Architecture of EtherCAT master and ethernet driver.

rect Ethernet driver is developed so that the master module directly controls Ethernet controllers without Linux network stack nor NAPI, which has the advantages of avoiding unnecessary memory copy and timely processing of EtherCAT frames. Consequently, the developed direct Ethernet driver for embedded systems achieves the similar functions to native EtherCAT-capable Ethernet drivers supported by Ether-Lab [5].

## 3. Experiment

#### 3.1 Setup

In order to evaluate the cycle time performance of the developed EtherCAT-capable driver, we build EtherCAT networks consisting of the Linux-based IgH EtherCAT master and commercial off-the-shelf EtherCAT slaves. An embedded system board, Variscite VAR-DVK-AM33 based on TI AM3354 MPU, and Linux kernel 3.2.0 with a real-time patch, PREEMPT\_RT-rt10 are employed for the IgH Ether-CAT master stack. For the EtherCAT slaves, we choose TI's TMDSICE3359 and Infineon's XMC4800 slave boards.

As the experiment scenarios, we consider two network configurations: one consists of one up to four TI slaves, and the other consists of one up to eight Infineon slaves. At each experiment scenario, the same type of slaves are used to eliminate the influence due to the heterogeneous device configuration. The distributed clock (DC) method is enabled for the synchronization of the EtherCAT networks.

According to the EtherCAT protocol, the cycle time of an EtherCAT network is defined as a fixed time period in each of which the master sends an EtherCAT frame and the frame returns back to the master after passing through all slaves. The cycle time is set as an operation parameter in the master application. In order to evaluate the cycle time performance, we first find the minimum cycle time for each network configuration that maintains the stable operation of the EtherCAT network without error.

Then, in order to measure the actual operational cycle time, we call the function clock\_gettime and store the Linux system time at each starting point of the master's cyclic operation in the master application. The actual operational cycle time is estimated based on the stored Linux system times by calculating the time differences between each two consecutive starting times of master's cyclic operations. For each round of experiment, we carry out a cold reset to eliminate any influence from the previous experiment.

# 3.2 Results

We carry out experiments for both the standard and developed Ethernet driver and compare the results in order to verify the performance improvement achieved by the developed driver. The minimum cycle time is obtained by gradually reducing the cycle time at each round of experiment until the master stops the stable operation and generates error messages. The obtained minimum cycle times for all network configurations are listed in Table 1, which obviously shows that the minimum cycle times of the developed driver are significantly improved in comparison to those of the standard driver.

For each minimum cycle time in Table 1, we measure 1,200 cycle time samples in the master application based on the Linux system clock after waiting 10 minutes for the system to stabilize from the cold reset. The results for one number of slave are shown in Figs. 2 and 3, which verify

0



400 600 800 1000 1200 600 800 1000 1200 200 200 400 Cycles Cycles (a) Standard Ethernet driver (b) Developed Ethernet driver

Fig. 3 Measured cycle times with a TI slave.

that both the measured cycle times and their oscillation frequencies of the developed driver are significantly reduced in comparison to those of the standard driver. Moreover, it is observed in Figs. 2 (b) and 3 (b) that the flat parts of the graphs occur repeatedly over the entire 1,200 cycle period, and the flat parts represent constant cycle times for quite a number of successive cycles. Note that the oscillation of measured cycle times is mainly caused by the jitter of the Linux system clock.

Since the results for multiple slaves exhibit the similar shapes as in Figs. 2 and 3, we omit the plots for multiple slaves but list the averages and standard deviations of the measured cycle times in Table 2. From Table 2, it is verified that both the averages and standard deviations of the measured cycle times generated by the developed driver are significantly reduced in comparison to those generated by the standard driver. Namely, the developed driver improves not only the cycle time but also the jitter of the cycle times.

Comparing Tables 1 and 2, it is revealed that the minimum cycle time set as a parameter is almost identical to the average of the measured cycle times, and the slight differ-

Table 2 Average (AVG) and standard deviation (STD) of measured cycle times

Slave board	Number	Standard driver		Developed driver	
	of	AVG	STD	AVG	STD
	slaves	(µs)	(µs)	(µs)	(µs)
Infineon	8	199.81	15.19	130.18	13.48
	4	200.32	15.13	125.05	9.71
	1	199.81	15.19	124.72	9.85
TI	4	199.84	15.19	100.05	13.68
	2	200.17	15.15	100.02	13.66
	1	199.99	15.71	100.15	13.73

ence is caused by the jitter of the Linux system clock used for the measurement of actual cycle times in the master application.

#### 4. Conclusion

We developed an Ethernet driver to improve the cycle time performance of EtherCAT networks with embedded Linuxbased master. The improvement was achieved by designing a direct interface between the master module and the Ethernet controller without the Linux network stack nor the NAPI of standard Ethernet drivers, which effectively reduces the time-consuming memory copy operations and increases the processing speed for EtherCAT frames. Extensive experiments were carried out for several configurations of Ether-CAT networks that were composed of the embedded Linuxbased master and commercial off-the-shelf slaves. The experimental results verified that the cycle time performance with the developed driver is significantly improved in comparison to the standard Ethernet driver.

#### References

- [1] D. Orfanus, R. Indergaard, G. Prytz, and T. Wien, "Ethercat-based platform for distributed control in high-performance industrial applications," IEEE Conference on Emerging Technologies & Factory Automation (ETFA), pp.1-8, Sept. 2013.
- [2] G. Prytz, "A performance analysis of EtherCAT and PROFINET IRT," IEEE Conference on Emerging Technologies & Factory Automation (ETFA), pp.408-415, Sept. 2008.
- [3] G. Cena, I.C. Bertolotti, S. Scanzio, A. Valenzano, and C. Zunino, "On the accuracy of the distributed clock mechanism in EtherCAT," IEEE International Workshop on Factory Communication Systems (WFCS), pp.43-52, May 2010.
- [4] IgH EtherCAT Master for Linux. [Online] http://www.etherlab.org.
- [5] IgH EtherCAT Master 1.5.2 Documentation. [Online] http://www. etherlab.org/download/ethercat/ethercat-1.5.2.pdf.
- J.H. Salim, R. Olsson, and A. Kuznetsov, "Beyond softnet," Proc. [6] 5th Annual Linux Showcase & Conference (ALS '01), pp.165-172, Oakland, CA, USA, Nov. 2001.
- [7] S.-M. Park, H. Kim, H.-W. Kim, C.N. Cho, and J.-Y. Choi, "Synchronization improvement of distributed clocks in EtherCAT networks," IEEE Commun. Lett., vol.21, no.6, pp.1277-1280, June 2017.