EtherCAT Protocol Used in High Performance Real-Time Servo Drive and Time Critical Applications

Mostafa AlNaimi, Ammar Abdullah, Karthik Keshava, Hussein Issa, Hady Bandar, and Housam Fares

ECE 5995
College of Electrical & Computer Engineering
Wayne State University

ABSTRACT:

Given the increased complexity of the control applications used in the various engineering systems, as well as, the need for a reliable, fast, low cost communication solutions in real-time industrial applications brought the attention to the Ethernet based communication protocols (more specifically EtherCAT). Moreover, the In-Vehicle control applications demands a real-time embedded environment with high bandwidth, low latency, and safety measures. The potential for the active/passive driver assistance applications used in the automotive solutions and complex communication networks are becoming more challenging to meet all the system application requirements. By using EtherCAT as the as the main/sub communication layer in a real-time embedded system will satisfy many strict timing and data requirements with more flexible topologies are possibly could be implemented to suit the system design specifications.

I. INTRODUCTION

In general, Ethernet provides the base requirements of an industrial communication protocol since Ethernet features high levels of bandwidth and data integrity. However, Ethernet is basically considered a communication medium rather than a complete industrial communication solution. In addition, the strict requirements of the control applications used in various industry aspects have more potential considerations including cost, latency, redundancy, and topology. All the mentioned above requirements demand a reliable complete communication system which offers a sustainable environment to control exchanged data between distributed nodes with a very low expected latency.

There are couple main reasons on why the industry and the system design developers have more interests in using Ethernet based communication solutions in high-performance distributed embedded networks which are listed below:

- low cost of network device components (switches, hubs, routers). For EtherCAT, there is no need for network control devices which means more savings.

- Various supported topologies which allow more flexibility with the system design requirements.

- Tremendous options of hardware components and software solutions to buy from vendors which help with cost and savings.

- High speed and throughput for large data.

This paper opens up the discussion for a potential use of EtherCAT protocol which supports the basis of real-time embedded communication networks by providing high speed, low latency, and low cost of hardware compared with other legacy serial communication protocols (CAN, LIN, and FlexRay). The discussion involves reviewing EtherCAT related articles and compare findings on overall hardware/software performance.

II. GOALS

The goals we want to achieve in this literature review paper could be summarized with the main highlighted points below:

1. Evaluate low latency & jitter, accurate hardware synchronizing system that can accommodate for less than 150us cycle time at a low hardware cost of Master and Slave devices.

2. Discuss flexible topologies and possible implementation could be used in distributed real-time systems/subsystems.

3. Explore EtherCAT concepts and "Process on the fly" terminology for cyclic and event driven types of messages are transferred in the EtherCAT network.

4. Expand the work done for the review with a practical example of an EtherCAT network between a Master and Slave devices to evaluate and analyze network performance.

III. REVIEW ON ETHERCAT CONCEPT & TERMINOLOGY

EtherCAT communication protocol is an Ethernet based real-time fieldbus system originally developed by Beckhoff Automation in 2003 [1]. The name of EtherCAT is an acronym for Ethernet for Control Automation Technology. The industrial network was designed around providing a low jitter and very accurate hardware synchronizing system that can achieve very short cycle time at low hardware cost. Another major benefit is that its master to slave communication is transmitted over regular ethernet cables with RJ-45 connectors which are very common and relatively inexpensive and robust. The hardware and software architecture make EtherCAT a suitable solution for hard and soft real-time system requirements in automation technology. Primarily EtherCAT is used in test, measurement, and Servo Drive applications throughout various industries.

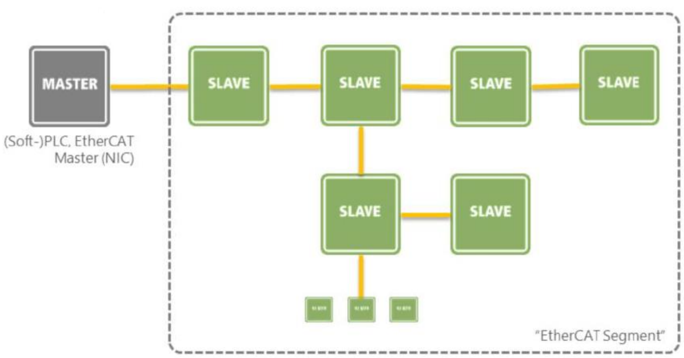

Figure 1: Possible EtherCAT Network [1]

EtherCAT is an internationally standardized open technology for a Master-Slave protocol. EtherCAT has a unique method of processing frames, results in it the fastest industrial Ethernet based technology on the market. It boasts accurate synchronization as jitter is less than 1us. These speeds are 100x faster than traditional CAN and 20x faster than CAN FD coming in at 100mbps. These speeds are only achievable by processing the data on the fly by the quick switching FPGA's in thee hardware of the device. EtherCAT is also Full-Duplex meaning information can flow BOTH ways at the same time (Full-duplex). Since the system is based on Ethernet all inherent collision avoidance standards CDMS/CD are built in through the IEE 802.3 standard. Another important point to EtherCAT is that it can support up to 65535 nodes per segment, and that means one master can handle that many slave devices.

IV. REVIEW ON ETHERCAT FRAME & COMMUNICATION

A. EtherCAT Frame:

The EtherCAT frame contains the frame header and a single or multiple Datagrams. The network embeds its payload into a standard Ethernet frame. In the datagram header information, it specifies what type of access the master is intending to do in terms of read/write or both accesses.

To access to a specific slave device, the system can use direct address mapping, or accessing multiple Slave devices through implicit addressing, or a broadcast frame to all slave devices [1].

The Master device could access cyclical mapped PDO messages through implicit addressing. Each datagram could address a specific PDO mapped area in the EtherCAT segment, where it could allocate up to 4 Gbytes of address space. During the network startup, each slave is assigned an address by the Master device. The Master device could access multiple slave devices in a single Datagram if they share the same region of the address space. A block diagram of the EtherCAT frames is shown in the figure below. SDO (event driven) messages will be accessed through mailbox communication where they do not have to be mapped into a global address space.

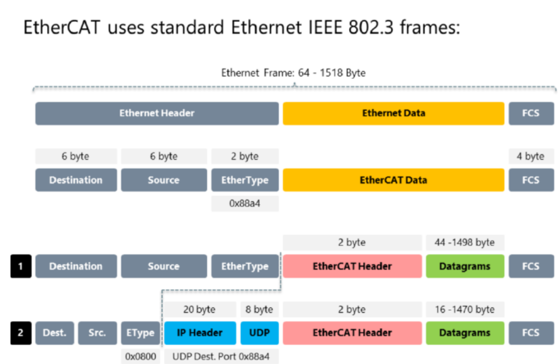

Figure 2: EtherCAT Frame [1]

B. Process on the Fly & FMMU:

The EtherCAT master device transmits a telegram (Datagram + Ethernet header) that go through each device. Each EtherCAT slave will read the data addressed to it “on the fly” and writes its data in the frame as the frame is going downstream to the next slave. The only delay is considered here is the hardware propagation delay.

The telegram’s maximum effective data rate increases to over 90 %, and due to the utilization of the full duplex feature, the theoretical effective data rate is even higher than 100 Mbit/s (> 90 % of two times 100 Mbit/s) [1].

EtherCAT technology overcomes these challenges and limitations of other Ethernet based solutions: the Ethernet packet is no longer received, then processed and copied as process data at every connection level. EtherCAT offers developed FMMU in each I/O terminal reads the data addressed to it, while the telegram passes through the device. At the same time, input data are inserted into the frame as it passes through. The telegrams are usually only delayed by couple nanoseconds [2].

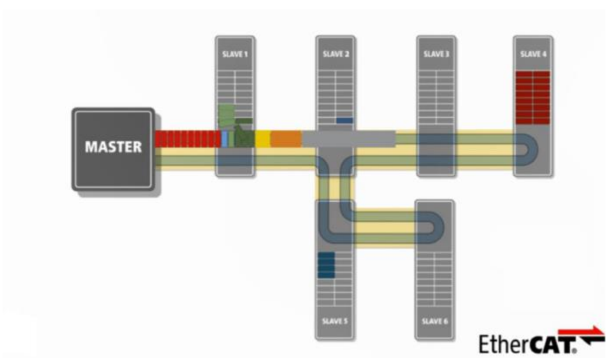

Figure 3: EtherCAT - Process on the Fly [1]

C. Working Counter:

The Working Counter is used as a safe-guard to track and validate frames and validate their accesses (Read/Write). Each EtherCAT Datagram that is sent, includes a Working Counter (WKC). The WKC tracks of all operational slaves in the network and increments after each successful read/write access. The Master will then compare WKC against the expected value to validate successful frames. The figures below depict the outline of an EtherCAT Datagram and how the WKC increments based on the command from the Master:

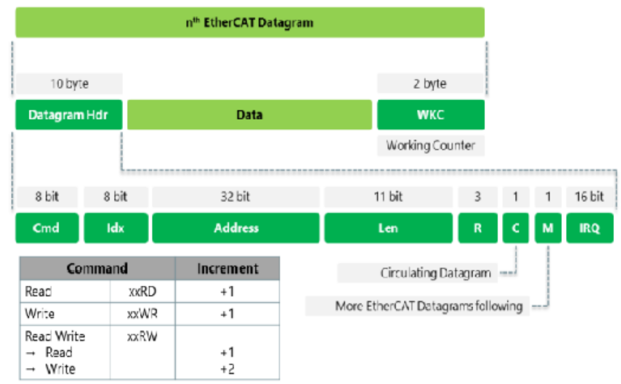

Figure 4: Communication Working Counter [1]

V. ETHERCAT TOPOLOGY

EtherCAT topology is very flexible since it comes in all different forms of networks whether that being a line, a ring, a star, and a tree network. Therefore, you are not limited to one network topology. They each have their own benefits and drawbacks. A few common topologies are listed below:

- **Line**: this is probably the easiest and most common topology to use since you do not need to worry about connection. But it does require multiple Ethernet ports on the Slave devices so the they could be daisy chained in a linear topology. The Master device will communicate with the first node (Slave); then, the frame will be transferred through the rest of the network.

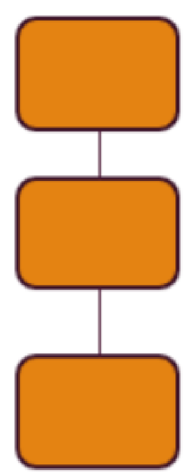

Figure 5: Line Topology

- **Ring**: this is the most preferred topology because it adds features of cable redundancy. If the cable between the two devices stops working for some connection problem, then the cable coming from the other direction will keep the EtherCAT communication going and complete the network. This topology is widely used with Real time PXI controller since it requires three or more Ethernet ports.

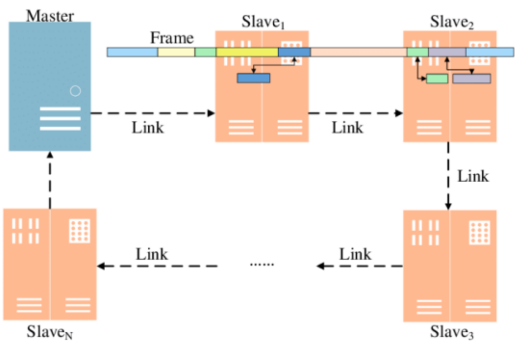

Figure 6: Ring Topology [9]

- **Star**: this is a very common topology especially to the new EtherCAT. It is used for Local Area Network (LAN) where each individual network is connected to a central connection point such as a hub or a switch. The advantage of this is that if one node breaks than it will not affect the others.

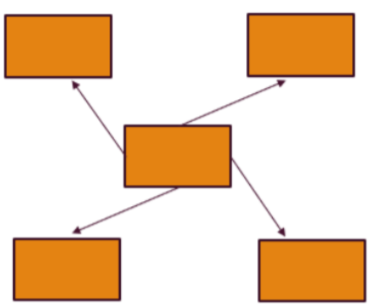

Figure 7: Star Topology

- **Tree**: this is a special type of topology because of the way it structure. The elements in it are connected in a way like the branches in a tree. They are usually used to organize the computers network. Due to the way its shape it allows for expansion of an existing network.

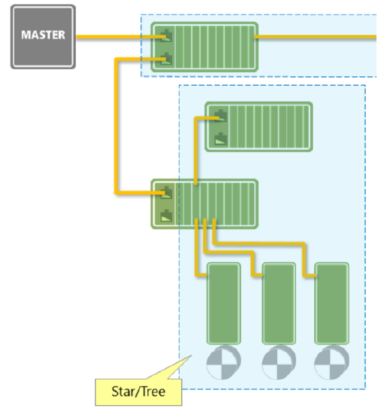

Figure 8: Tree Topology [1]

VI. HARDWARE SYNCHRONIZATION

In EtherCAT, it is important to have a proper synchronization among the master and slaves. In the EtherCAT there is a single master clock and a reference slave clock, and all other slaves have non-reference clock. The synchronization is achieved based on distributed clocks method where in the slave reference clock is tuned to the master clock and other slave clocks are tuned to the slave reference clock. Slave-slave synchronization has a high accuracy and the jitter is <12ns[6]. The Master-slave synchronization is described herein. The Master clock is called system time and the slave time is called local time. The slave keeps track of the global time, system time. The synchronization among master and slaves is done is three steps propagation delay measurement, offset and drift compensation which is summarized below.

Propagation delay measurement: Master starts the delay measurement by sending a broadcast write(BWR) datagram to all connected slaves. The slaves store the timestamps at what time the BWR arrived at each slave. The Master then is able to read the each of the Slave received time and calculate the System time delay or the propagation delay. This is a one-time measurement when the system is started.

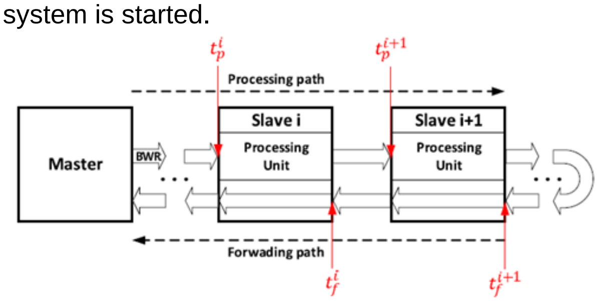

Figure 9: Propagation delay measurement [6]

Offset Compensation: Offset is the difference in master system time versus each of the slave's local time. The master is able to read the local time of slaves and then calculate offsets for each of its slaves. The master then stores these values in System_Time_offset register of the respective slaves. Like the Propagation delay measurement this calculation is performed when the system is started.

Drift compensation: Jitter of the master clock and the minute difference in the oscillators frequency of slaves causes a drift in time. In EtherCAT, reference slave drift references to difference in master's system time and the reference time (reference slave) and non-reference slave drift means difference between reference time (reference slave) and non-reference slaves system time. The figure 9[6] shows how the master triggers the compensation process by sending datagrams and how the slaves' clocks are adjusted.

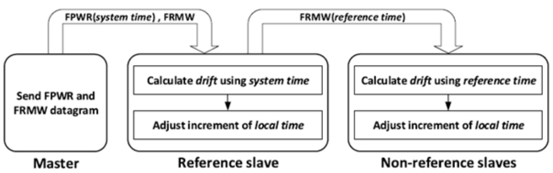

Figure 10: Drift Compensation method [6]

<u>Review of Paper [6] – Quick summary</u>

The paper [6] discusses the modifications to improve the synchronization accuracy of Master-Slave. The paper [6] uses off-the-shelf EtherCAT devices for the experimental setup. The setup consists of two configurations: i) one Master and two slaves and ii) one master and six slaves

The measurements are done to note the different system time values available at master and slaves. Performance is measured by applying the modifications (which are done at the Master application) and another time without the modifications. The modifications proposed in paper[6] target the bias components that are present in EtherCAT protocol namely a) propagation delay not being compensated between master and reference slave b) Drift compensation not accounting for the each save's drift size. The results described in paper[6] show a considerable improvement in the synchronization performance of Master-Slave.

The main idea behind paper[6] is to enhance the master application without tearing the EtherCAT protocol itself. With this approach, the methods described in paper[6] can be easily applied to existing EtherCAT networks, while keeping the costs low. The one warning from the paper is that improvements are directly corelated to the reduction of jitter in master system time.

VII. TIMING / PERFORMANCE ANALYSIS

As a result of experiments done in Paper [5] using "Process on the Fly" technique in the EtherCAT frame processing, one frame will go through all the Slave devices in the network where you can imagine a train station and the data will be written/read on each Slave device. Implementing this concept in the EtherCAT network resulted the a very low latency/jitter comparing with other Ethernet based protocols.
The Maximum data could be transferred in one frame goes up to 1486 bytes and that results in a maximum frame time of 123.6 us (ignoring all hardware delays) [5]. Most likely software control loop could run at 300us per frame keeping in mind all the user-based processing. The resulted jitter was captured in the experiments fell down between 11ns – 18ns which very close to the theoretical jitter (10ns).

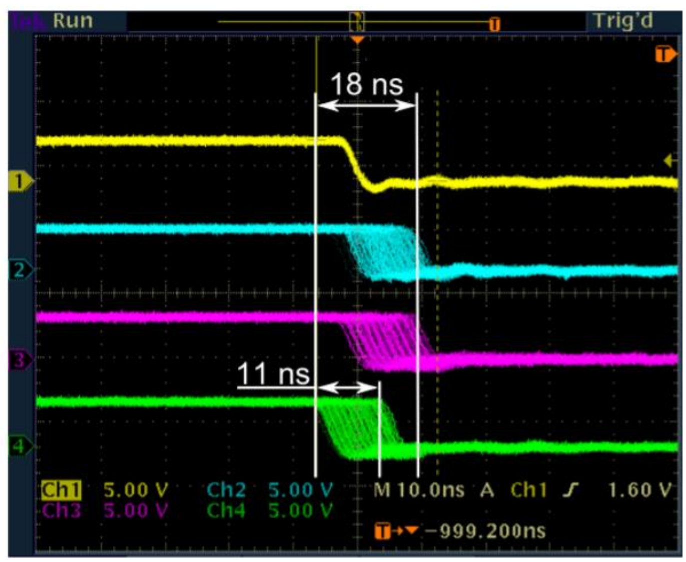

Figure 11: Capturing Jitter in the Network [5]

The resulted jitter was captured in the experiment was done in Paper [3] is at 11ns for the sync signals where in the system communication cycle, the resulted jitter is at 14ns, and that's very close to the results of experiment in was done in Paper [5]. Both results verify the powerful EtherCAT capability of providing low communication jitter.

VIII. COSTS

The Costs and the prices are being distributed in the case of the EtherCAT into many factors, it always varies between the Master device and the Slave device and It will be listed as follows:

- Device Costs
- Engineering Costs
- Setup Costs
- Maintenance Costs
- License Costs

A. **Device Cost** depends on the number of vendors supporting, and the Master device would play a crucial role for low pricing Using less Hardware of-course reduces the prices such as cables. There is no need for switches and network traffic control devices when using EtherCAT. In general, the Master does not require a specialized hardware [1]. It only requires an embedded system platform and a network interface. The Slave devices requires a specialized controller (ESC) that needs follow the generic design and specification of the EtherCAT Slave controller.

B. **Engineering Costs**: The network production should be independent from topology such as we explained above such as No switch and No Hubs used for production and give the space for saved data setting to do the troubleshooting.

C. **Setup and Maintenance Costs** mostly depends on the Topology and the Diagnosis.

D. **License Costs** is a free charge for the Master Devices. Slave devices are usually supplied by specialized vendors with a cost range of ($50 - $1200) depending on the system requirements and complexity of the applications.

Finally, the price difference is shown in the comparison between EtherCAT and other configurations, 6% savings on the Master Cards, 12% savings on the technology of the EtherCAT Slave Controller and 2% savings on the Connector.

IX. CONCLUSION

Ethernet for Control Automation technology (EtherCAT), is a low costing industrial network invented to accommodate short cycle times. Some main advantages are its method of communication which consists of inexpensive cables and connectors, and its hardware and software architecture which make it a suitable solution for both hard and soft real-time requirements in automation technology. EtherCAT is generally inexpensive so many different products can take advantage of its system. Its master to slave communication system is simple, going through only ethernet cables and is a great option. It has unique way to process frames, makes it the fastest industrial Ethernet technology on the market. In EtherCAT, it is important to have a proper synchronization among the master and slaves. Slave-slave synchronization has a high accuracy and the jitter is <12ns[6]. This paper explains the process and material costs of EtherCAT while explaining its many advantages. Examples of experiments done to EtherCAT products and the results that come from these experiments go hand in hand. The sources and references used help explain the many advantages that come with this network.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Randy Boone, Adjunct Professor of Electrical and Computer Engineering at Wayne State University for the support and guidance throughout the semester.

APPENDIX A

DEFINITIONS, ACRONYMS, ABBREVIATIONS

ETG	EtherCAT Technology Group
SAE	Society of Automotive Engineers
IEEE	Institute of Electrical and Electronics Engineers
EtherCAT	Ethernet for Control Automation Technology
ESI	EtherCAT Slave Information
SDO	Service Data Object
PDO	Process Data Object
ESC	EtherCAT Slave Controller
FMMU	Field Management Memory Unit

REFERENCES

1. ETG. "EtherCAT *Communication Principles*". EtherCAT Technology Group presentation contains an in-depth technical introduction about EtherCAT technology. 2018. Web.
2. Beckhoff Automation. "*EtherCAT System Documentation*". 2020. Web.
3. Cummings, R., Richter, K., Ernst, R., Diemer, J. "*Exploring Use of Ethernet for In-Vehicle Control Applications: AFDX, TTEthernet, EtherCAT, and AVB*". SAE International World Congress & Exhibition - Journal of Passenger Cars, 2012. Web.
4. Loser, R. and Kleinkes, M., "*Real-Time Robot Positioning based on Measurement Feedback Control*". SAE Int. J. Mater. Manf. 2016. Web.
5. Orfanus, Dalimir, Reidar Indergaard, Gunnar Prytz, and Tormod Wien. "*EtherCAT-based Platform for Distributed Control in High-Performance Industrial Applications*". IEEE Conference on Emerging Technologies & Factory Automation (ETFA), 2013. Web.
6. SUNG-MUN PARK, HYOUNG-WOO KIM, HONG-JU KIM, AND JOON-YOUNG CHOI. "*Accuracy Improvement of Master Slave Synchronization in EtherCAT Networks*". IEEE Access, 2020. March 23,2020. Web.
7. KEVSER OVAZ AKPINAR, (Member, IEEE), AND IBRAHIM OZCELIK. *"Analysis of Machine Learning Methods in EtherCAT-Based Anomaly Detection"*. IEEE Access, December 18, 2019. Web.
8. ETG. "*EtherCAT: What are the User Benefits*". EtherCAT Technology Group seminar on Competitive Advantages through EtherCAT. 2011. Web.
9. Li, Liying & Cong, Peijin & Cao, Kun & Zhou, Junlong & Chen, Mingsong & Wei, Tongquan & Hu, Xiaobo & Hu, Shiyan. "*Game Theoretic Feedback Control for Reliability Enhancement of EtherCAT-Based Networked Systems*". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2018. Web.