EtherCAT Technology for Reliable Industrial Communication Networks

Vincent Rutagangibwa[1]
[1]P.G Scholar & Assistant Lecturer
[1]Gujarat Technological University (GTU) India & Mbarara University of Science and Technology (MUST) Uganda

Abstract— The modern developing communication industry requires increasingly faster, more throughput and reliable com-munication networks which are able to work in a hard real-time situation. This paper presents the EtherCAT communication technology that does not only meet the above requirements but also allows market to reduce implementation costs in order to have affordable and reliable services. The paper further explains the EtherCAT operation principles with the most important mechanisms such as Frame processing on-the-fly, logic ring topology and slave synchronization among other mechanisms. The comparison of EtherCAT with the traditional most pop-ular industrial based networks is also presented to show the leading position of this upcoming technology in the Industrial communication networks in terms of reliability, performance, implementation and cost.

Key words: EtherCAT, Communication Networks, Indus-trial Automation

I. INTRODUCTION

EtherCAT is currently a market leading technology in the Real-time Ethernet domain offering high real-time performance in combination with a wide set of functionalities. The initiator and main developer company Beckhoff has become a significant vendor of advanced equipment for industrial automation. The EtherCAT technology now has broad support and involvement by many companies [1]. Industrial Ethernet and EtherCAT in particular are hot topics in the global automation industry. Traditional fieldbus systems are showing their limitations today as new technologies are appearing that provide more performance and robust support for Internet Protocols. Automation systems developers have to decide if, when and how to adopt these next generation network technologies in order to keep with the market demands.

EtherCAT is an open source, relatively high speed fieldbus system based on Ethernet basics. The motivation for EtherCAT development was to improve Ethernet capabilities which are applicable in automation applications which require real time operations and low component costs. Its hardware, software and network design has been optimized to provide a deterministic high performance capability for moving data around on the embedded device network. EtherCAT holds considerable potential in embedded networking applications, particularly in motion control and factory automation.

Applications requiring high bandwidth or high update rates like CNC, robotics and packaging are all candidates for EtherCAT technology. It is also consider for highly distributed applications, especially those with a large number of nodes, like materials handling and data acquisition systems [2].

Originally developed by Beckhoff Automation GmBH and introduced to the public in November 2003, EtherCAT is now in the final stages of international standardization with ISO and IEC (primarily in IEC 61158-4-12, IEC 61784-2, and ISO 15745). The trade group for EtherCAT is the EtherCAT Technology Group (ETG) [3]. Beckhoff provides an extensive range of fieldbus components for all common I/O and fieldbus systems. The wide choice of I/O components implies that the bus system best suited for a particular application can be chosen [4].

The outstanding performance, flexible topology and simple configuration characterize EtherCAT Technology. The real-time Ethernet technology sets new standards where conventional fieldbus systems reach their limits. 1,000 distributed I/Os in 30us, almost unlimited network size, and optimum vertical integration courtesy of the current Ethernet and Internet technologies. With EtherCAT, the costly Ethernet star topology can be replaced with a simple line or tree structure – no expensive infrastructure components are required. All types of Ethernet devices can be integrated via a switch or switch port.

Firldbuses have become an integrated component of automation technology. They have been tried and tested and are now widely established. It was fieldbus technology that enabled the wide-scale application of PC-based Industrial Control Systems. While the performance of controller CPUs particularly for IPC is increasing rapidly, conventional fieldbus systems tend to represent bottlenecks that limit the performance control systems can achieve. An additional factor is the layered control architecture consisting of several subordinate (usually cyclic) systems; the actual control task, the fieldbus system and perhaps local expansion buses within the I/O system or simply the local firmware cycle in the peripheral devices. Reaction times are typically 3-5 times higher than the controller cycle time – an unsatisfactory solution. Above the fieldbus system level, i.e. for networking controllers, Ethernet has already been the state of the art for some time in communication networks. What is relatively new is its application at the drive or I/O level. The main requirements for this type of application are high real-time capability, suitability for small data quantiles, and naturally cost-effectiveness. What is promising is that EtherCAT meets these requirements and at the same time makes Internet technologies available at the I/O level.

There are many different approaches that try and provide real-time capability for Ethernet; for example, the CSMA/CD media access procedure is disabled via higher level protocol layers and replaced by the time slice procedure or polling [5], other propositions use special switches that distribute Ethernet packets in a precisely controlled timely manner. Whilst these solutions may be able to transport data packets more or less quickly and accurately to the connected Ethernet nodes, the times required for the redirection to the outputs or drive controllers and for reading the input data strongly depend on the implementation and hence non-deterministic challenges result. A logical addressing scheme is defined which permits small-sized process data to be packed. As a consequence, communication efficiency is very high (up to 90%) which achieves very short cycle times [6]. This makes this solution particularly attractive for connecting decentralized peripherals (i.e. remote I/O devices) to the application master (either a real or a soft-PLC) [7]

II. ETHERCAT OPERATING PRINCIPLES

The basic operating principle of EtherCAT Technology is that all nodes in an EtherCAT network can read from and write data to the EtherCAT telegram as it passes by with only a short constant delay in each slave device (independent of the size of the packet). Typically, a larger number of slaves can be accommodated using only one telegram, thereby optimizing bandwidth usage. This very efficient usage of bandwidth is enabled by a concept called Logical Addressing [6]. EtherCAT networks are characterized by a daisy-chain topology and a master/slave architecture. The master node periodically transmits standard Ethernet frames containing several telegrams (Fig. 1), while the slaves process the frame on-the-fly (Fig. 2) and read and write data in the telegrams. When a slave receives a byte, this byte is processed and transmitted to the next slave without waiting for the entire frame to arrive. The last slave in the chain has to transmit back the frame to the master and then the cycle starts once again [8].

EtherCAT uses a 100 Mbit/s full duplex (thus collision free) link between nodes, which are physically connected using 100Base-TX (over usual CAT5e/CAT6 cable), 100Base-FX, or LVDS links. An EtherCAT network uses the Master node communicating with many slave nodes paradigm. This requires foreknowledge of the types of slaves, their cabling order, and master-initiated communication. The payoff is deterministic behaviour and high performance, much less bandwidth wasted on overhead, and cycle times measured in microsecond and milliseconds, but not in seconds. A master is usually a PC-class workstation with an ordinary Ethernet NIC. An Ether-CAT slave will typically consist of two Ethernet PHYs (IN and OUT)., EtherCAT slave controller (providing Ethernet MAC functionality), and a Microcontroller. As an alternative, MCU-less digital I/O is also provided directly from the slave controller, allowing for simple, low-cost slaves without any slave-side coding [2]. Practically, EtherCAT slave controllers show a very creative departure from the traditional Ethernet MAC in a way that to address a slave, instead of specifying a unique MAC address (which they don‘t posses), one of the following address modes is set in the EtherCAT header;

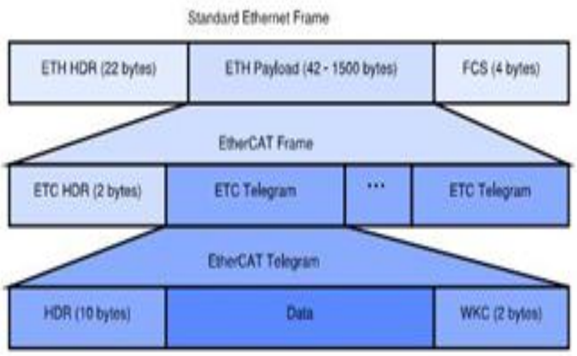

Fig. 1: EtherCAT Frame Structure

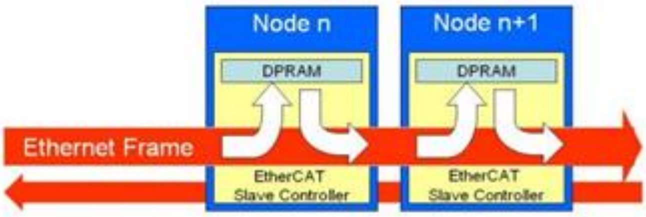

Fig. 2: The EtherCAT Frame Processing On-the-Fly [9]

– A slave‘s unique, centrally assigned node number,
– Its position in the chain (e.g. 5th slave from the master), or
– The slave‘s assigned portions of a system-wide 32-bit address space.

This unique feature is implemented in HW using a Functional Memory Mapping Unit. Portions of this address space may be mapped into each slave‘s dual-port RAM, with granularity down to one bit. In this way, the end-user application could access the entire network as a single, contiguous memory space.

EtherCAT uses a high performing mode of operation, in which a single frame is usually sufficient to send and receive control data to and from all nodes. The EtherCAT master sends a telegram that passes through each node. Each EtherCAT slave device reads the data addressed to it on-the-fly, and inserts its data in the frame as the frame is moving downstream. The frame is delayed only by hardware propagation delay times. The last node in a segment or branch detects an open port and sends the message back to the master using Ethernet technology's full duplex feature. 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 greater than 100MBits/s. The EtherCAT Master is the only node within a segment allowed to actively send an EtherCAT frame; all other nodes merely forward frames downstream. This concept prevents unpredictable delays and guarantees real-time capabilities.

The master uses a standard Ethernet Media Access Controller (MAC) without an additional communication processor. This allows a master to be implemented on any hardware platform with an available Ethernet port, regardless of which real-time operating system or application software is used. EtherCAT slave devices use EtherCAT Slave Controller (ESC) to process frames on-the-fly and entirely in hardware, making network performance predictable and independent of the individual slave device implementation [3]. This interesting feature makes EtherCAT suitable for Industrial Real-time Applications.

III. ETHERCAT FEATURES

A. *Protocol:*

The EtherCAT protocol is optimized for process data and is transported directly within the Ethernet frame. It may consist of several EtherCAT telegrams each serving a particular memory area of the logical process images that can be up to 4 Gigabytes in size. The data sequence is independent of the physical order of the Ethernet terminals in the network; addressing can be in any order. Broadcast, multicast and communication between slaves are possible. Direct Ethernet frame transfer is used in situations where maximum performance is required.

EtherCAT embeds its payload in a standard Ethernet frame. The EtherCAT frame is identified with the Identifier (Ox88A4) in the EtherType field. Since the EtherCAT protocol is optimized for short cyclic process data, the use of bulky protocol stacks, such as TCP/IP or UDP/IP, can be eliminated as shown in fig. 3. EtherCAT applications are not limited to a single subnet, but EtherCAT UDP packages the EtherCAT protocol into UDP/IP datagram 3. This enables any control with Ethernet protocol stack to address EtherCAT systems. Communication across routers into other subnets is also possible. In this variant, system performance obviously depends on the response times of the EtherCAT networks itself are hardly restricted at all; the UDP datagram only has to be unpacked in the first station.

In addition to data exchange according to the master/slave principle, EtherCAT is very suitable for communication between controllers (master/master). Freely addressable network variables for process data and a variety of services for parameterization, diagnosis, programming and remote control cover a wide range of requirements. The data interfaces for master/slave and master/master communication are identical. For slave to slave communication, two mechanisms are available. Upstream devices can communicate to downstream devices within the same cycle and thus extremely fast. EtherCAT only uses standard frames, according to [5], the frames are not shortened. EtherCAT frames can thus be sent from any EtherCAT MAC, and standard tools (e.g. monitor) can be used.

Fig. 3: EtherCAT in a standard Ethernet frame (according to IEEE 802.3)

The EtherCAT frame contains one or more Datagrams. The Datagram header indicates what type of access the master device would like to execute:

- Read, write, or read-write
- Access to a specific slave device through direct address-ing, or access to multiple slave devices through logical addressing (implicit addressing)

B. Topology:

EtherCAT topology network can be one/and or a combination of Line, tree or star. EtherCAT supports almost any topology. The bus or line structure known from the field bus also becomes available for Ethernet, without the quality limitations implied by cascaded switches or hubs. The combination of line and branches or stubs is particularly useful for system wiring. The required interfaces exist on many devices (e.g. on I/O modules),; no additional switches are required. Naturally, the classic switch-based Ethernet start topology can also be used. Wiring flexibility is further maximized through the choice of different cables. Flexible and inexpensive standard Ethernet patch cables transfer the signals in 100BASE-TX mode. Plastic Optical Fiber (POF) complements the system for special applications.

C. Distributed Clocks:

Accurate synchronization is particularly important in cases where spatially distributed processes require simultaneous actions. This may be the case for example, in applications where several servo axes carry out coordinated movements simultaneously. The most powerful approach for synchronization is the accurate alignment of distributed clocks, as described in the IEEE, 1588 standard [1]. In contrast to fully synchronous communication, where synchronization quality suffers immediately in the event of a communication fault. Distributed Aligned clocks have a high degree of tolerance versus possible fault-related delays within the communication system. With EtherCAT, the synchronization of the devices is fully based on a pure hardware machine. Since the communication utilizes a logical ring structure, a timestamp can be sampled within each device for the incoming and the outgoing frame. With these timestamps, the master can determine the propagation delay offset to the individual slave clocks simply and accurately. The distributed clocks are adjusted based on this value, which means that a very precise network-wide time base with a jitter of significantly less than 1 msec is achievable (Fig. 4) [9].

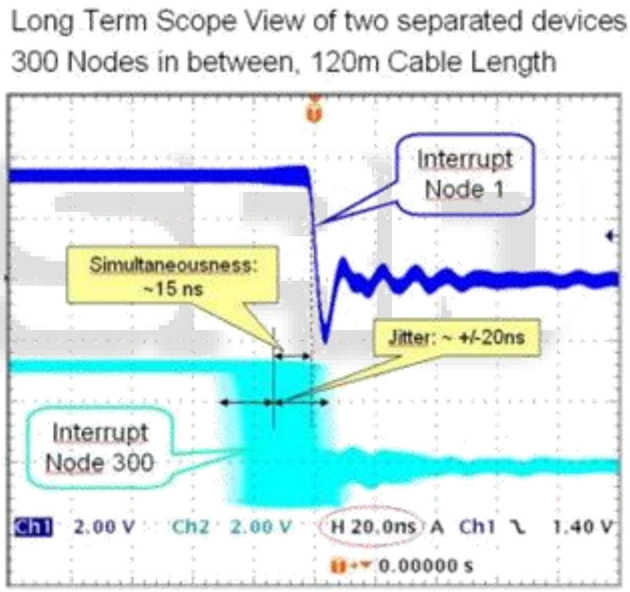

Fig. 4: EtherCAT Clock Distribution Accuracy Example [9]

With EtherCAT, the data exchange is completely hardware based on "mother" and "daughter" clocks. Each clock can simply and accurately determine the other clocks run-time offset because the communication utilizes a logical and full duplex Ethernet physical ring structure. The distributed clocks are adjusted based on this value, which implies that a very precise network-wide time base with a jitter of less than 1us is achievable (Fig. 2). This accuracy is ideal for synchronized motion control applications and for integration of measurement tasks within the same network.

D. Performance:

EtherCAT reaches new dimensions in network performance thanks to hardware integration in the slave and direct memory access to the network controller in the master. The complete protocol processing takes place within hardware and is fully independent of the run-time of protocol stacks, CPU performance or software implementation. The update time for 1000 I/Os is only 30us including I/O cycle time. Up to 1486 bytes of process data can be exchanged with a single Ethernet frame – This is equivalent to almost 12000 digital I/Os. The transfer of this data quantity only takes 300us. The detailed performance analysis of EtherCAT Technology can be found in [6].

E. Diagnosis:

Experience with fielbus systems shows that availability and commissioning times crucially depend on the diagnostic capability of the communication network, only faults that are detected quickly and accurately can be rectified quickly. Therefore, special attention was paid to exemplary diagnostic features during the development of EtherCAT. During com-missioning, the actual configuration of the nodes (e.g. drives or I/O terminals) should be checked for consistency with the specified configuration. The topology should also match the configuration. Due to the built-in topology recognition down to the individual terminals, this verification does not only take place during system start-up but automatic reading-in of the network is also possible (i.e. configuration upload). The automatic evaluations of the associated error counters enable precise localization of the critical network sections. A gradual or changing source of error such as EMI influences defective connectors or cable damage are detected and located, even if they do not yet over-strain the self-healing capacity of the network.

F. High Availability:

Increasing demands in terms of system availability are centered for with optional cable redundancy that enables devices to be replaced on a live network. Adding redundancy is very inexpensive: the only additional hardware is another standard Ethernet port in the master device and the single cable that turns the line topology into the ring topology. Switchover in case of device or cable failure only takes one cycle, so even demanding motion control applications survive a cable failure without problems. EtherCAT also supports redundant masters with hot standby functionality since the EtherCAT slave controllers immediately return the frame automatically if an interruption is encountered, failure of a device does not lead to the complete network being shut down.

G. Safety over EtherCAT (SoE):

In the interest of realizing safe data communication over EtherCAT, the SoE protocol has been disclosed within the EtherCAT technology Group. EtherCAT is used as a single-channel communication system for transferring safe and non-safe information. The transport medium is regarded as a black channel and not included in safety. A safety frame containing the safe process data and the required data backup is included in the EtherCAT process data and this container is safely analyzed in the devices at the application level. Beckhoff currently offers three safe I/O components;

- An input terminal with 4 safe inputs
- An output terminal with 4 safe outputs and
- A logical terminal with configurable safety logic and 4 local safe outputs.

Safety-related parameterized of the devices can be implemented via a safe configuration tool integrated in the standard programming environment (TwinCAT) and then the resulting safe parameter set is uploaded (password-monitored) to the safe logical terminal. During each start-up, the logic terminal distributes the safe application parameters to the configured input and output terminals. This enables simple exchange of input and output terminals without having to adapt or reload the configuration. EtherCAT is used as a black channel, and thus the communication system plays no part in the safety considerations.

H. Ethernet over EtherCAT (EoE):

The EtherCAT technology is not only fully Ethernet-compatible, but also characterized by particular openness by design. The protocol tolerates other Ethernet-based services and protocols on the same physical network usually even with minimum loss of performance. There is no restriction on the type of Ethernet device that can be connected within the EtherCAT segment via a switch port. The Ethernet frames are tunneled via the EtherCAT protocol, which is the standard approach for Internet applications (e.g. VPN, PPPoE (DSL), etc.). EtherCAT devices can additionally feature other Ethernet protocols and hence act like a standard Ethernet device. The master acts like a layer 2 switch that redirects the frames to the respective devices according to the address information. All Internet technologies can therefore also be used in the EtherCAT environment integrated web server, email, FTP transfer etc.

I. Infrastructure:

Since no hubs and switches are required for EtherCAT, costs associated with these devices including power supply, installation, etc. are avoided. Standard Ethernet cables and standard low cost connectors are used, if the environment conditions permit this. For environment requiring increased protection sealed connectors according to IEC standards are specified.

IV. Implementation Aspects

The EtherCAT Technology was developed with very low cost devices in mind, like I/O terminals, Sensors, and Embed-ded Controllers. EtherCAT only uses standard Ethernet frames according to IEEE 802.3 standard. These frames are sent by the master devices and the Slave devices extract and/or insert data on the fly.

A. The EtherCAT Master:

EtherCAT communicates a maximum of 1486 Bytes of distributed process data with just one Ethernet frame. So unlike other solutions, where the master devices in each network cycle has to process, send and receive frames for each node, EtherCAT systems typically only need one or two frames per cycle for the entire communication with all nodes. Therefore EtherCAT masters do not require a dedicated communication processor. The master functionality puts hardly any load on the host CPU which can handle this task easily besides processing the application program. So the EtherCAT can be implemented without special and expeand mediumsive active plug-in card by just using a passive NIC card or the on-board Ethernet MAC. Implementation of an EtherCAT master is very easy, particularly for small and medium-sized control systems and for clearly defined applications.

EtherCAT masters have been implemented on a wide range of RTOS. The availability of open-source

EtherCAT master products, make it possible to build a full low cost industrial control system based only on open source software components [10]. The practical feasibility of such a solution was demonstrated by the results presented in [11]. Master stacks are available as open source projects, as sample code and as commercial software. Implementation services are also avail-able from a variety of vendors and for a variety of hardware platforms. Information about this fast growing offering can be found within the product section of the EtherCAT website [3].

B. The EtherCAT Slave:

A cost effective EtherCAT slave controller is used in the slave devices. With EtherCAT, the slave does not need a microcontroller at all, simple devices that get by with an I/O interface can be implemented only with the EtherCAT Slave Controller (ESC) and the underlying PHY, magnetics and the RJ45 connector. The process data interface (PID) to the slave application is a 32-bit I/O interface. This slave without configurable parameters needs no software or mailbox protocol. The EtherCAT state machine is handled in the ESC and the boot-up information comes out of the EEPROM that also supports the identity information of the slave. More complex slaves that are configurable have a host CPU on board that is connected to the ESC with na 8-bit or 16-bit parallel interface or via SPI. The performance of the host CPU is determined by the slave application. The EtherCAT protocol software can be run additionally and the stack manages the state machine and the communication protocol.

Several manufactures provide EtherCAT slave controllers and their functionality can be implemented in a very cost effective way on FPGAs for which binary code is available as a buy-out license. The slave controllers typically feature an internal DPRAM and offer a range of interfaces for accessing this application memory. Information regarding choice of EtherCAT slave controllers suitable for your application can be found on [3]

V. SUMMARY

EtherCAT Technology makes Ethernet down to the I/O level technically feasible and economically achievable. Full Ethernet compatibility, Internet technologies, maximum bandwidth utilization, real-time characteristics at low cost are all outstanding features of this new and promising EtherCAT Technology. EtherCAT Technology is thus the most suitable, inexpensive and reliable technology for modern Industrial Communication Networks.

VI. ACKNOWLEDGMENT

Great thanks to Gujarat Technology University (GTU), Mbarara University of Science and Technology (MUST) for the support. Special thanks to Mr. Rajesh Sola for his expertise and contribution towards this research work.

REFERENCES

[1] “Ieee 1588-2002: Ieee standard for a precision clock synchronization protocol for networked measurement and control systems.”
[2] M. Diu, “Ethercat: Technology spotlight,” in Current: Nuvation.
[3] “Ethercat technology group (etg),” in http://www.ethercat.org/.
[4] Beckhoff, “Beckhoff fieldbus components: Beckhoff main website, http://www.beckhoff.com/.”
[5] “Ieee 802.3: Carrier sense multiple access with collision detection (csma/cd) access method and physical layer specifications.”
[6] G. P. Norway, “A performance analysis of ethercat and profinet irt,” in 13th IEEE International conference on emerging technologies and factory automation.
[7] “Evaluation of ethercat distributed clock performance,” in IEEE, vol. 8, no. 1, February 2012
[8] L. L. Bello, G. Patti, G. Alderisi, V. D. Patti, and O. Mirabella, “A flexible mechanism for efficient transmission of aperiodic real-time messages over ethercat networks,” in IEEE, 2014.
[9] M. Rostan, J. E. Stubbs, and D. Dzilno, “Ethercat enabled advanced control architecture,” in IEEE, 2010.
[10] M. Cereia, S. Scanzio, and C. D. degli Abruzzi, “A user space ethercat master architecture for hard real-time control systems,” in IEEE, 2012.
[11] M. Cereia, I. C. Bertolotti, and S. Scanzio, “Performance of a real-time ethercat master under linux,” in IEEE Transactions on Industrial Informatics, vol. 7, no. 4, 2011, p. 679.