Survey on Present Real-Time Ethernet Solutions

Stefan Krywult, Christian Steiner

Research Report 12/2004

Abstract

Ethernet has become the standard for local area networks all over the world. Ethernet controllers are mass products and therefore quite cheap. The industry tries to use cheap standard technology in their factories instead of expensive proprietary equipment. As a consequence Ethernet finds its way into the factory automatization. This paper starts with a short introduction to the main concepts of Ethernet. Then five techniques of real-time communication based on the Ethernet standard are analyzed.

Keywords: Ethernet, real-time, switched Ethernet, IEEE 1588 Precision Clock Synchronization Protocol, Powerlink, PROFInet, EtherCAT

Contents

1 Introduction

Ethernet prevailed in local area networks (LANs) for both, private and commercial use. Almost every company deploys Ethernet based networks to interconnect their servers and their employees' personal computers (PCs). Various flavors of Ethernet exists providing a communication speed up to 10 gigabits per second, which qualifies Ethernet to be used even for wide area networks (WANs).

However, Ethernet has a lot of shortcomings that do not allow using Ethernet in a hard real-time environment (e.g. automatization). This paper gives a short overview of the history of Ethernet, discusses the main advantages and disadvantages of Ethernet and provides an insight into five different techniques trying to make Ethernet suitable for the use even in time-critical applications.

2 Ethernet

In this section the basic concepts of Ethernet are explained roughly. It is based on [Dem01], where some much more detailed information about Ethernet can be found.

2.1 A brief history of Ethernet

Ethernet was born in 1979. A consortium consisting of Digital, Intel and Xerox released the Ethernet standard. This early form of Ethernet is also known as Standard Ethernet (10BASE-5) or "thick wire" or "yellow cable" because the data is transferred with a speed of 10 Mbps using thick coaxial cables with a yellow sheath.

The transceiver that is needed to connect the network card to this cable is rather expensive. To minimize the cost of an Ethernet installation a new standard was released. In the Cheapernet (10BASE-2) standard, that is also known as "thin wire", a thin coaxial cable is connected to the Ethernet cards directly. The speed of 10 Mbps and the bus architecture were inherited from the "thick wire".

In the bus architecture the bus is a huge single point of failure. Even if only one of the termination resistors of the bus is missing, the communication between all network members is impeded. This and the fact, that the coaxial technology is still rather expensive, leads to development of Ethernet based on a twisted pair cable (10BASE-T). Again the speed was not changed. The 10baseT Ethernet is based on the star topology. All network members are connected to a central repeater (a "bus in a box"). The effort of cabling increased but the twisted pair CAT3 cable is very cheap and widespread because it is also used for telephony. There is still one single point of failure, the repeater which is much easier to check and exchange than a whole bus connecting all network members.

With the Fast Ethernet standard (100BASE-TX), the speed was raised to 100Mbps. As media the only slightly more expensive Cat5 twisted pair cable is used. This standard also defines a full duplex communication and an automatic negotiation of the transmission speed between network interface cards and repeaters.

There are also Ethernet standards that define a communication at 10Mbps over fibre optics (10BASE-F), at 100Mbps over Cat3 twisted cable (100BASE-T4/100BASE-T2) and at 100Mbps over fibre optics (100BASE-FX), but these have never become widely accepted.

The next successful Ethernet standard is Gigabit Ethernet. A communication at 1Gbps is defined in this standard either using high quality Cat7 twisted pair cable (1000BASE-CX, 1000BASE-T) or fibre optics (physical layer similar to *Fibre Channel*, 1000BASE-SX and 1000BASE-LX). Gigabit Ethernet does not use CSMA/CD, the medium access strategy that is typical for Ethernet.

The newest Ethernet standard, the 10 Gigabit Ethernet is designed for WANs (Wide Area Networks). It supports half duplex communication only, and does not use CSMA/CD.

The Standard Ethernet was slightly modified and released by the *IEEE* as the standard *IEEE 802.3 CSMA/CD*

All this classes of Ethernet use different physical media and different coding techniques. But most of them use the same media access strategy (CSMA/CD) and same MAC frame format.

2.2 CSMA/CD

CSMA/CD stands for *Carrier Sense Multiple Access with Collision Detection*. That means every node of the network that has to send some data, first listens to the network. If no traffic is noticed, the node starts sending. If two nodes start sending simultaneously, the packets collide on the shared medium. The nodes are able to detect this collision, stop sending, and retransmit their packet of data at a random point of time in the future.

The CSMA/CD algorithm in detail (see figure 1, header only):

1. wait until medium is free
2. send data and check for every bit, if a collision has occurred, in case of a collision continue with 4
3. if no collision has occurred, data has been sent successfully finished.
4. send unique jam signal so that every other node can detect this collision
5. if the maximum number of retries is reached, give up, the data cannot not be sent

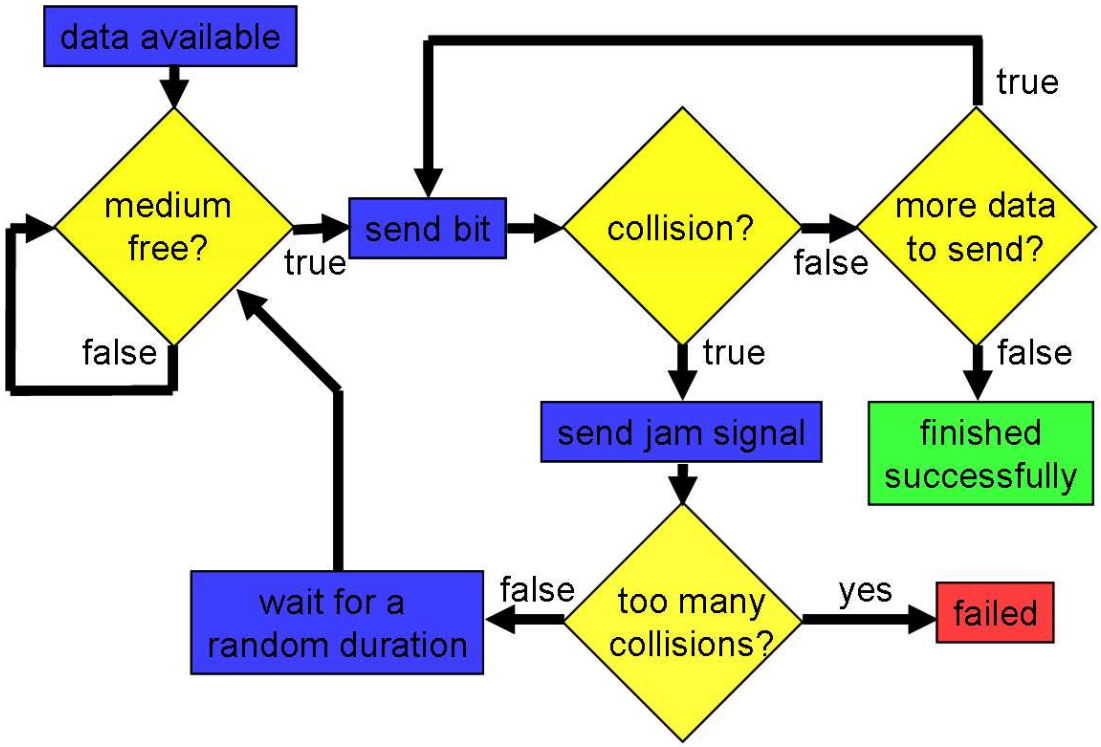

Figure 1: CSMA/CD algorithm

6. wait for a random duration; the maximum time to wait for depends on the number of the current retry

7. retry and go to 1

2.3 Frame Format

Ethernet frame format: There are two different Ethernet frame formats defined: one in the Ethernet standard and one in the IEEE 802.3 standard.

- Ethernet (see figure 2, header only)
  - 8 bytes preamble (for synchronization)
  - 6 bytes destination address
  - 6 bytes source address,
  - 2 bytes packet type field
  - up to 1500 bytes user data
  - 4 bytes checksum (CRC 32)
- IEEE 802.3 (see figure 3, header only)

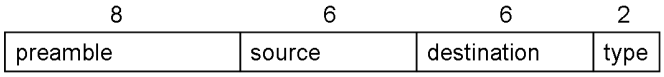

Figure 2: Ethernet frame header

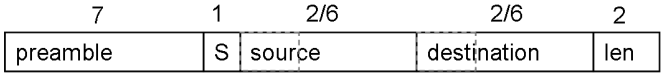

Figure 3: IEEE 802.3 frame header

- 7 bytes preamble (for synchronization)
- 1 byte start delimiter (marks the beginning of the packet)
- 2 or 6 bytes destination address
- 2 or 6 bytes source address
- 2 bytes packet length
- up to 1500 bytes user data
- up to 46 bytes padding to achieve a minimum packet length of 512 bits (in the Ethernet standard this has to be guaranteed by the layer above the Ethernet layer)
- 4 bytes checksum (CRC 32)

The IEEE 802.3 frame format with addresses that are only two bytes long are hardly used.

The two standards are interoperable because the Ethernet preamble is the same bit sequence (101010...) as the IEEE 802.3 preamble and the start delimiter. The type field of the Ethernet standard is always larger than the length field of the IEEE 802.3. Therefore Standard Ethernet and IEEE 802.3 frames can be distinguished.

In the following sections we refer to 10/100Mbit Ethernet when talking about Ethernet.

3 Ethernet and real-time communication

The main advantages of Ethernet are:

- standardized
- approved
- simple

- wide spread
- very cheap
- high speed

The disadvantages of Ethernet in view of real-time communication are:

- CSMA/CD leads to trashing (much traffic generates even more traffic)[Kop97].
- CSMA/CD is nondeterministic (it is not guaranteed that a node will be able to send its data until a particular point of time in the future).
- Since 100Mbit Ethernet, full duplex communication is available. But this feature is rarely needed in a real-time environment.
- Multicast communication is extensively used in real-time systems [Kop97], hardware should support this by providing a broadcast feature, in Ethernet networks with star topology repeaters are necessary for broadcasts.
- Star topology results in expensive installation costs and lot of space used for wiring. Therefore bus topology is preferred.
- Ethernet frames are quite large, but only very small pieces of information need to be transferred in real-time networks, so the communication is inefficient.
- Ethernet does not provide any fault tolerance mechanisms.

4 Real-Time-Ethernet solutions

To achieve a solution that fits the requirements of real-time communication, the nondeterministic CSMA/CD has to be circumvented. This can be done in three different ways. For every way various may solutions exist. The following ones are considered in this paper:

1. Topology based
    - Switched Ethernet
2. Software based
    - IEEE 1588 Precise Clock Synchronization Protocol
    - Powerlink
3. Hardware based

- PROFInet V3 (Isochronous Ethernet)
- EtherCAT

Every solution deals with other disadvantages of the Ethernet technology as well.

4.1 Switched Ethernet

As explained in the previous sections, classical ethernet comes in many flavors: the coax-based thin (*10base-5*) and thick ethernet (*10base-5*) as well as the (more sophisticated) hub-based twisted pair ethernet (*10base-T*, *100base*). Common to all flavors is the underlying bus access protocol *CSMA/CD* and its non-deterministic medium access delays. The probability of frame collisions across the wire is proportional to the number of stations in a single collision domain (plus, naturally, the general network load).

A traditional approach to minimize the number of frame collisions lies in reducing the collision domains, resulting in a network consisting of many micro-segments[1], separated by bridges. Today, bridges are increasingly replaced by switches, which provide one collision domain per port. Present-day switches are much more than simple multiport-bridges - due to their ASIC-based hardware architecture, ultra-fast simultaneous-multiple-access memory and their offering of additional IP-based services (such as configuration via *SNMP*), they easily outperform practically all existing bridge-based solutions.

As there is no standard for Ethernet switch implementation, several technologies have evolved over the years.

- Matrix-based: These switches have their roots in telecommunication switches. They provide a great number of ports, but have significant problems with simultaneous broad-, multi- and unicasts.

- Bus-based: These switches consist of a high speed core bus (backbone) which is shared among a number of I/O ports. They natively support broadcasts, however output buffer overflows can occur when many input ports are forwarded to a single output port.

- Shared memory: Switches using the shared-memory technology store all incoming packets in memory. Each frame's destination port is determined via a MAC lookup table and buffered at the proper output port, thus avoiding the dreaded head-of-line (*HOL*) blocking[2]

[1] an ethernet segment containing only one station.

[2] HOL-blocking switches store incoming packets in a FIFO (first in, first out) queue at the input port or the switch fabric. Incoming packets, including those destined for non-congested output ports, can only be forwarded after all packets ahead of it have been forwarded, resulting in a significant performance loss.

A problem inherent with all switch implementations is the addition of the switches' internal buffer delays. In theory, a non-buffering switch can only exist if the bitrate of all output ports is at least equal to the bitrate sum of all possible input ports. Due to the essentially bidirectional communication requirements, such a switch cannot exist.

In fully switched micro-segments, *CSMA/CD* is no longer necessary and kept only for compatibility issues.

4.1.1 Performance Considerations

Several methods exist to further enhance the performance of (fully) switched ethernet networks.

- Bitrate improvements: *IEEE802.3u* (100Base-T4, 100Base-TX, 100Base-FX), *IEEE802.3z* (1000Base-CX, 1000Base-SX, 1000Base-LX), *IEEE802.3ab* (1000Base-T). These improvements offer reduced collision window sizes, reduced inter-frame spacing, an extended ethernet slot time and packet bursting (mangling of several small frames into a single frame).

- Reducing forwarding delay: Many switches provide a new *cut-through* mode in addition to the traditional *store & forward* mode. In *cut-through* mode, arriving frames are forwarded before they have completely arrived at the input port. This results in a significantly reduced frame transit time, at the cost of transmitting possibly erroneous frames (frames are forwarded before CRC checks take place).

- Congestion control: This method tries to prevent buffer overflows by introducing a new *pause* command (*IEEE802.3x*). Furthermore, some switches can slow down inbound traffic by simulating collisions by sending "jam frames". The latter technique should not be used in real-time systems, as it may introduce indeterministic message response times.

- Priority handling: This method is mandatory for real-time behavior. Ethernet frames are extended by a 3-bit priority field (*IEEE802.1p*, *IEEE802.1q*), resulting in at most 8 priority levels. However, no standard exists regarding the number of queues per port.

4.1.2 Performance Evaluation

The totally delay introduced by a single switch can be computed according to the following formula:

$$delay_{tot} = delay_{switching} + delay_{frameforwarding} + delay_{buffering}$$

- Switching delay: fixed value depending on switch ($\sim 10\mu s$)

- Frame forwarding delay: depends on switch mode (*cut-through*, *store & forward*)
- Buffering delay: empiric analysis of input traffic pattern

4.2 IEEE 1588 Precision Clock Synchronization Protocol

The *IEEE1588 Precision Clock Synchronization Protocol for Network Measurement and Control Systes*, or, in short, *Precision Time Protocol* (PTP) defines a method for the synchronization of spacially dispersed real-time clocks connected over a packet-aware network (mainly ethernet). The technique has been developed by *Agilent* (former *Hewlett Packard* department *Test and Measure*). Basically, it achieves a deterministic behavior suited for real-time systems by decoupling the application from base ethernet (*CSMA/CD*. This is accomplished by providing a synchronized real-time clock in each bus participant.

The *IEEE1588* specification defines

- a method for automated segmentation of PTP networks,
- the synchronization of interconnected PTP clocks,
- the elation and control of a master clock, and
- the PTP network management protocol.

Automated network segmentation guarantees a network consisting exclusively of acyclical connections (allowing for exact calculation of point-to-point signal propagation delays).

Clock synchronization is necessary because *CSMA/CD* can, due to its limitations (see preceding sections), not be used for normal clock synchronization (erratic collisions result in delayed/lost packets). The synchronization process itself comprises the following steps:

1. The master clock generates a *sync frame* containing the local time plus an assessment of protocol stack latency.
2. The exact time of transmission is measured by a hardware clock (if available) and sent in *follow-up frame*.
3. Each recipient can calculate the time difference between its local clock and the master clock. The local quartz drift rate is adapted to catch up with the master clock.

The frequency at which the *sync frame* is sent can vary between 0.125Hz and 0.5Hz (that is, clock synchronization can take place at most every two seconds).

The elation of a master clock follows the *Best Master Clock* Algorithm (BMC): each PTP clock advertises its properties in a public data set. Thus, no voting is necessary, allowing for true hotplugging of bus participants.

In conjunction with 100MBit Ethernet, PTP permits a precision $< 1\mu s$.

4.3 Ethernet Powerlink

Ethernet Powerlink was introduced in November, 2001, by *B&R*. By concept, it acts as a replacement of TCP(UDP)/IP with a deterministic protocol stack.

Traffic takes place in predefined slots; this scheme is called *Slot Communication Network Management* (SCNM). It effectively prevents the development of collisions and provides a guaranteed bandwidth for isochrone data.

Ethernet Powerlink distinguishes between a managing node (*Powerlink Manager*) and passive nodes (*Powerlink Controller*). The *Powerlink Manager* coordinates the workflow by generating the time slots and is also responsible for the configuration of all nodes in the network. Moreover, it is the only node in the network allowed to act on its own. *Powerlink Controllers* are passive bus stations that can send only when requested (i.e., polled) by the manager:

- `PollRequest`: directed at a specific node address
- `PollResponse`: sent as a broadcast

Ethernet Powerlink uses an isochrone protocol to cyclically exchange data between the nodes.

- Start period: *Start-of-Cyclic* frame, data preparation if necessary, ...
- Cyclick period: processing of all active bus participants
- Asynchronous period: *End-of-Cyclic* frame
- Idle period: remaining time between completed asynchronous period and beginning of new cycle (can be 0)

Several extensions exist that add further functionality (e.g. running TCP/IP on top of Powerlink).

Due to performance issues, only hubs are allowed in a Powerlink Network. Its main advantages are an average cycle time $< 400\mu s$ and a jitter $< 1\mu s$, which satisfy *IAONA* realtime class 3+4 specification.

4.4 PROFInet

PROFInet is developed by the PROFIBUS consortium and is described in roughly [Wen03] [Pbus1] and in a more detailed way in [Pbus2]. It is based on Fast Ethernet and defines three levels of communication:

1. communication that is not time-critical
2. time-critical communication with an update rate down to $10ms$, soft real-time (SRT)
3. time-critical communication with an update rate in the region of $1ms$, hard-real-time

For the standard communication that is not time critical the UDP and the TCP protocol are used.

TCP and UDP cause an additional delay. Therefore an optimized communication stack is used for the time-critical communication. This stack is based directly on the Ethernet layer to minimize delay.

Another delay might originate from the network traffic generated by other devices. To keep this nondeterministic delay as small as possible, packet prioritisation as defined in the IEEE 802.1q standard is used. Every network device handles those packets first, that are flagged as time-critical.

These two measures suffice to achieve update rates down to $10ms$. However, this update rate can not be guaranteed. This communication technique is adequate for soft real-time communication only.

For the hard real-time communication isochronous[3] Ethernet is used. Isochronous Ethernet uses time division multiplexing to guarantee the necessary bandwidth for the transport of the Real-Time data and a collision-free access to the shared medium: The communication is organized in rounds. In each round a time slot is reserved for every real-time communication. In this time slot only one single PROFInet controller is allowed to send data. The remaining time between the end of one communication round and the next one is used for soft real-time communication and communication that is not time-critical (see figure 4).

Special hardware is needed to perform clock synchronization among the PROFInet devices in order that all devices agree on when a new time slot starts. This hardware also controls the time division multiplexed access to the shared medium.

With this technique update rates down to $1ms$ are possible with a jitter less than 1us.

4.5 EtherCAT

EtherCAT has been developed by the EtherCAT Technology Group and is described in [Ecat1]. The principle of EtherCAT is very similar to that of the

[3]From Greek isochronos, from is- + chronos time: uniform in time, having equal duration, recurring at regular intervals (*Merriam-Webster Online Dictionary, http://www.m-w.com).*

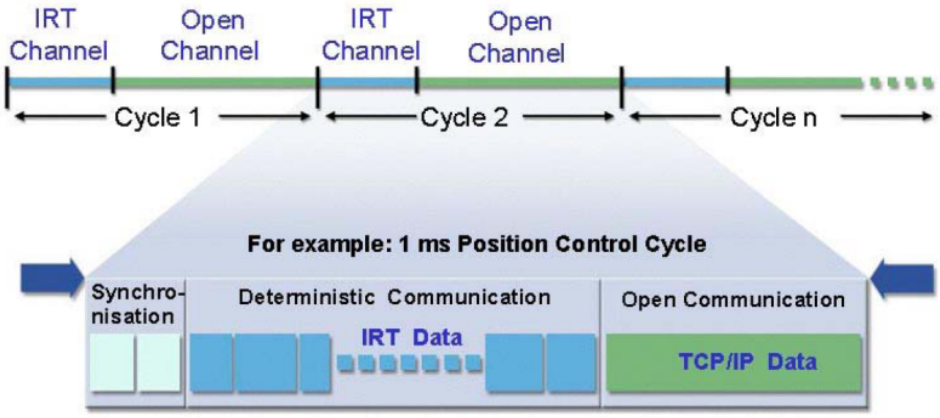

Figure 4: Communication in PROFInet system[Pbus1]

Interbus protocol.

All physical layers defined in the Ethernet standards are supported by EtherCAT. Each EtherCAT slave has two receive (RX) and two transmit (TX) interfaces. For small distances between two EtherCAT devices the E-bus protocol is used. It is based on LVDS (low voltage differential signal) and allows cost efficient transmissions over a distance of up to 10 meters.

All slaves in an EtherCAT system are connected to other slaves to form a line. A tree structure also is supported. The EtherCAT system is treated as a single Ethernet device. The first slave receives EtherCAT telegrams through its RX interface 1 sent by the master using raw Ethernet or UDP. UDP is needed, if the EtherCAT slaves are not in the same IP network as the master, but UDP has a larger overhead than raw Ethernet. The received packet is passed on from one EtherCAT device to the next using RX and TX interface 1 until it arrives at the last slave. Every device regenerates the data telegram, reads that bits in this packet it is interested in and writes its data to the packet if necessary. After the last EtherCAT device has processed the data telegram, the packet is returned to the first slave using the RX and TX interfaces 2 of every device. The first node then sends back a reply to the master.

In an EtherCAT cluster there are point to point connections only, therefore CSMA/CD is not needed. Standard Ethernet switches can be used in this clusters and a standard PC with an Ethernet controller can act as master.

The checksum of the Ethernet MAC frame is used to detect communication errors. In the EtherCAT telegrams a working counter is stared that is increased by every slave. The master can compare the counter to the number of the slaves that are expected to be online. If the successor of an EtherCAT slave fails, it closes the open loop and sends back the data telegram to its predecessor. The master is able can locate the failed node.

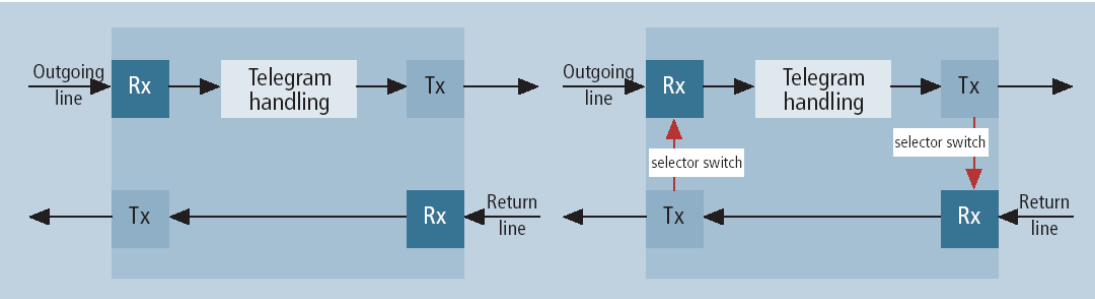

Figure 5: Interconnection of EtherCAT slaves[Ecat1]

The handling of the data is done bit-by-bit on the fly and can be realized in hardware easily. Therefore the devices are rather cheap and fast. Every slave delays the request telegram only for a few bits.

The data stored in the slaves can be accessed using either physical addressing or logical addressing or multiple addressing or broadcast. Physical addressing: Each telegram is addressed to exactly one slave and reads data from and writes data in its 64kB address space. Logical addressing: At start-up the Master configures the Fieldbus Memory Management Unit (FMMU) that maps parts of the memory of the slaves with to an address in a logical, global address space. This can be done with a granularity of a single bit. Using this technique the data can be communicated more efficiently, in particular if the portions of data are small because single bits on various slaves can be read or written at the same time. Multiple addressing: The slave addressed in the telegram and all its successors read or write data at the given address, if this address lies within the physical address space of the slaves. Broadcast: Every slave reacts on the request.

In addition to these addressing modes every EtherCAT device is able to participate in the normal Ethernet communication. This allows the development of smart devices that offer any IP-based service (for instance an EtherCAT device with an integrated web server for configuration).

The main advantages of EtherCAT are:

- every device with an Ethernet controller can act as master
- efficient handling even of small pieces of data without a sub-bus
- possible cycle time $< 100\mu s$
- deterministic behaviour
- the complete Ethernet bandwidth is available
- completely compatible with the Ethernet standard, standard IP services can be implemented and EtherCAT can share the network with other Ethernet applications

5 Conclusion

Ethernet uses the indeterministic CSMA/CD medium access strategy, so it is innately not suited for time-critical applications. Three different principles are used to make the medium access strategy of Ethernet deterministic:

Topology The topology of the network can be designed so that the collision-free communication is guaranteed. Under this assumption CSMA/CD is deterministic.

Time Division Multiple Access The communication is scheduled in time slots. Every node in the network knows when it is allowed to send data. To keep the nodes in sync a clock synchronization is necessary. TDMA can be implemented either in software (e.g. IEEE 1588, Powerlink) or in hardware (e.g. PROFInet). Only one node is allowed to send at a time, therefore there are no collisions.

Time-critical subsystem It is also possible to encapsulate the complete time-critical parts of an application in zoned subsystems that provide an Ethernet compatible interface (e.g. EtherCAT). Collisions can't occur within a subsystem because Ethernet is not used inside these systems. And outside these systems the communication is either not time-critical or the determinism of the communication has to be achieved in another way.

Software solutions are cheaper and easier to integrate in an existing environment but only hardware solutions can guarantee a minimal jitter.

Real-time Ethernet makes it possible to build real-time applications using many cheap and well approved off-the-shelf Ethernet equipment. Configuration and management can be done using a web browser, a Simple Network Management Protocol (SNMP) client, Telnet or even Secure Shell (SSH). But it also involves a considerable overhead, because in many real-time applications most of the complex Ethernet functionality is not needed and implementing a TCP/IP stack and a web server on every sensor and actuator of a real-time system might be uneconomic. Moreover, none real-time Ethernet solution discussed in this paper features fault tolerance. But fault tolerance is a precondition for the use in hard real-time applications.

References

[Dem01] Demuth, Christian *Skriptum LOKALE NETZE (Computer Networks)*, TU Wien, 2001

[Kop97] Kopetz, Hermann, *Real-Time Systems*, Kluwer Academic Publishers, Norwell, Massachusetts, 1997

[Wen03] Wenk, Matthias, *PNO zndet die dritte Stufe "Isochrones Realtime Ethernet"*, IEE, 48.Jahrgang 2003, Nr.03

[Pbus1] PROFIBUS Working Group, *PROFInet Technologie und Anwendung*, Version November 2003, www.profibus.com

[Pbus2] PROFIBUS Working Group, *PROFInet - Architecture, Description and Specification*, Version 2.01, August 2003, www.profibus.com

[Ecat1] EtherCAT consortium, *EtherCAT - the Ethernet fieldbus*, www.ethercat.org