Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Evaluation of Real-Time Behaviour in Virtual Automation Networks

J. Beran*, F. Zezulka*

**Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno, Czech Republic, (e-mail:beranj@feec.vutbr.cz) ,*
*** Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno, Czech Republic, (e-mail:zezulka@feec.vutbr.cz) ,*

Abstract: Ethernet-based fieldbuses continuously penetrate industrial automation and many of them have been standardized (PROFINET, EtherCAT, SERCOS III, etc.). This fact fosters utilization of established Internetworking technologies, such as TCP/IP protocol suites and web services with the aim of widening of the communication scope and seamless interconnection with higher levels of industrial automation. However, industrial applications with their special requirements especially in terms of real-time, functional safety and security call for proper extensions of the existing Ethernet-based fieldbuses. This paper presents state of the art in development of a test bed which will serve for evaluation of real-time parameters of a given point-to-point connection based on the established QoS metrics.

1. INTRODUCTION

Some features of office networks and telecommunication technologies are not tailored to automation needs. Hence, it is necessary to spend significant effort on extensions to meet the requirements of industrial applications. These are mainly preserving real-time behaviour, preserving functional safety, extending system security, and integration of wireless technologies.

Virtual Automation Networks (VAN), an integrated project of 6th Framework Programme (6FP), tries to achieve integration of existing Ethernet-based fieldbuses with both office networks and telecommunication infrastructures with a stress to the abovementioned aspects.

A part of the activities is dedicated to preserving real-time parameters of industrial communication when extending the communication scope from a single LAN to multiple LAN communication. In such a case the communication requires routable protocols, e.g. UDP/IP.

Our approach to this problem is introduction of established methods of Quality of Service (QoS) resulting in preserving determinism and availability of communication.

The aim of our work is to evaluate influence of different network components on the communication determinism, and observe its improvement when employing DiffServ approach. The expected result is to draw a general conclusion of determinism feasibility and provide a solution suitable for industry. Technically, the process was divided into two parts: qualitative analysis whose results should retrieve sources of determinism disturbance and methods how to cope with them, and quantitative analysis, whose results will provide enumeration of the determinism disturbance and possible improvement using DiffServ approach.

Providing quantitative analysis required development of a specialized test bed capable of very precise measurement of QoS metrics, e.g. latency, jitter, bit-error-rat, and packet loss.

2. REAL-TIME SYSTEMS AND REAL-TIME COMMUNICATION

In the following, relations among terms such as real-time system and real-time communication have to be clarified to have a common view in this field.

2.1 Real-Time Systems

B.P Douglass (Douglass, 2000) defines real-time systems with regard to system response to an event. Utility functions are used to evaluate real-time requirements of a system. The utility function depicts usefulness of system response as a function of time. If the utility function of a specific system drops with time, it means that the timing of the response is to be taken into account and the system is real-time. Moreover:

If the utility function drops step-wise from 1 to 0, the system is called Hard Real-time (HRT). This means that any response delayed more than required is useless. This is usually followed by the system shut down or damage.

If the utility function drops smoothly, the system is called Soft Real-time (SRT). This means that response delayed more than required degrades the data but the data can still be used. However, performance of such a system is decreased according to the delay.

If the utility function does not drop significantly with regard to the time constants of the system, the system can be called Non-real-time (NRT). It is difficult to imagine a system which is not sensitive to delay at all. Therefore, a NRT system is such whose time constants are significantly higher than the time constants of the communication.

2.2 Real-Time Communication

A system can be modelled as a set of objects which interchange information among each other based on their functionality. Communication links among objects can be modelled as communication objects. These communication objects can possess various communication parameters. Bandwidth, latency, and jitter will be considered in the following (Lorentz, 2005).

If the communication parameters are qualitatively better (higher bandwidth, lower latency and jitter) than the system requires, we can neglect them, for not having any influence on the system behaviour. This can be for instance an autonomous system with a fast internal bus and CPU.

If the communication parameters are comparable to the system's requirements, we have to evaluate the system's real-time requirements and introduce appropriate real-time communication. This applies typically in distributed systems.

Consequently, real-time systems require real-time communication.

Real-time communication is classified differently from real-time systems. A basic classification of real-time communication acknowledges four real-time classes as follows (Collective, 2005). They are based on implementation aspects of the communication:

- Class 0: Non-real-time (NRT) – class without any special time constraints.
- Class 1: Soft Real-time (SRT) - scheduling on top of UDP/TCP with scalable cycle time; used in factory floor and process automation (local and enterprise domain).
- Class 2: Hard Real-time (HRT) - scheduling on top of MAC/LLC layer (L2); cycle time 1…10 ms. Used for direct process control (local domains only).
- Class 3: Isochronous Real-time (IRT) - with precise time synchronisation and scheduled network traffic; requires dedicated HW components. Cycle time 250 μs…1 ms; jitter less than 1 μs. Used especially for motion control (local domain).

It can be seen that the higher determinism required, the less complex the stack has to be and the more proprietary components are necessary.

3. QUALITY OF SERVICE IN IP NETWORKS

QoS in IP networks represents ability of a network to provide better transmission quality to certain network traffic in heterogeneous networks. The considered metrics are latency, jitter, out-of-order packets, transmission errors, and bit-error rate (BER). QoS represents rather a set of precautions taken by network components and traffic management policies than a single general mechanism providing the expected result. However, two general approaches can be applied:

The first approach of QoS fulfilment is IntServ (Integrated Services). This approach, utilizing, for instance, RSVP (Resource Reservation Protocol), relies on reservation of network resources. However, due to enormous enlargement of contemporary networks, administration of dozens of thousand of reservations is not applicable. Therefore, this approach is considered not efficient for Internet any longer.

The latter QoS approach is called DiffServ (Differentiated Services). It is based on classification of transmitted data into flows based on predefined criteria with the aim of treating different data flows differently based on the application requirements and their time critical nature.

The core idea is to recognize time-critical (real-time) traffic and assign it appropriate priority treatment. Best-effort traffic (non-real-time) can then occupy the leftover bandwidth available.

Most of the following information was taken from (Ford, 1998). Cisco is the world leading producer of network components such as routers, switches and strongly contributes to development and standardisation of Internetworking protocols.

DiffServ approach concerns several mechanisms: Classification, Traffic Shaping, Congestion Management, and Congestion Avoidance. As a first step of providing QoS to time-critical flows, Classification and Congestion Management mechanisms are considered. Traffic shaping and Congestion Avoidance have not been addressed yet. However, some of their parts can be added later on to support the core mechanisms. The following subchapters will introduce the core DiffServ mechanisms.

3.1 Classification

Before data can be treated in an appropriate manner, they have to be classified. The classification process consists of two steps. Firstly, the incoming packet has to be identified and secondly, the packet has to be marked. There are several classification techniques like Priority Based Routing (PBR), Committed Access Rate (CAR), and Network Based Application Recognition (NBAR). A common denominator of these techniques is that their instances reside in the nearest QoS capable component; either a L3-switch or an access router. The protocols' struggle results in setting of an appropriate code in DSCP field in IP header of the examined packet.

However, our classification approach differs. Industrial applications are extremely strict with QoS requirements. Fortunately, as a trade-off, industrial traffic can be well scheduled in advance. In time-critical applications, transmission intervals, data volumes and communication partners are known in advance. Consequently, if the given network is administrated consistently, we can shift the burden of classification directly to network devices generating the flow. If the abovementioned presumption is fulfilled, we gain a fair saving of time normally consumed by the classification protocols when recognizing the flows.

3.2 Congestion Management – Advanced Queuing

A packet or datagram arriving at router has to be examined on network layer (L3) to determine the next hop. In other words, a MAC address of the next hop has to be determined based on the IP address introduced in the packet. This procedure, however optimized, is of low determinism. Therefore, incoming traffic to be processed must be queued and waits for processing.

Establishment of multiple queues is a step forward to provide flows with differentiated treatment. Consequently, the question remains: How should the queues be served to deliver optimal behaviour? Partial answer to this is that it is the criteria of optimality that has to be defined first. Based on its definition, appropriate queuing algorithm can be chosen. Therefore, none of them is implicitly considered ultimately dominant over others. The fundamental algorithms will be introduced in the following summary:

- First-in-first-out (FIFO) – using this algorithm, no flow is favoured over others. The queues are served symmetrically.
- Custom Queuing (CQ) – dedicates a predefined portion of bandwidth to each queue and serves them in a round-robin principle. The algorithm is byte-wise fair. A drawback of this method is that there is rather a dominance of time-critical traffic over best-effort traffic than a guarantee of determinism.
- Priority Queuing (PQ) – prioritizes queues strictly by priorities. The queue with the highest priority will be served until it is empty followed by the next highest, etc. It is suitable for transmission of low duty-cycle time-critical data. A drawback is that if the time-critical data is voluminous, best-effort packets can be dropped.
- Weighted Fair Queuing (WFQ) – is bit-wise fair, favouring lower volume traffic over the large and gives flows of higher priority proportionally higher bandwidth. It has been made a generic queuing mechanism in many routers for its salient suitability for most applications.

We expect that proper classification together with suitable queuing algorithm will increase communication determinism in multiple LAN topologies (described in 4.2).

4. INCREASING DETERMINISM IN IP-BASED INDUSTRIAL NETWORKS

To improve communication determinism, one must perform qualitative analysis of sources of latency and jitter in a node-to-node connection. The more complex the topology is, the more sources of stochastic influence exist on the path. Let us classify the topologies into three classes which factor into in future VAN networks: (order of complexity).

4.1 Single LAN Topology

This topology has been established in PROFINET IO switched networks. The sources of latency and jitter are the following: device stack, line propagation, and switch processing (Moyne, 2005).

Propagation latency can be neglected in comparison to others. Switch latencies and jitters are minimal if no significant congestions take place, which means order of dozens of microseconds for latency. In case of congestions, they can be minimized by using VLAN tagging with differentiated priorities. Finally, the greatest deal of latency is caused by end device stacks. Each ISO/OSI layer adds extensive latency when passed through. PROFINET IO fights against the problem by employing mere L2 communication for run-time data exchange, eliminating UDP/IP layers. It can be afforded because of no L3 components in the given topology. Even IRT communication can be provided by introducing specialized switches.

4.2 Multiple LAN Topology – Enterprise Network

This topology, contrary to the previous, is extended by routers providing wider communication scope. In terms of latency and jitter, routers affect the communication in the following points:

- They draw in greater complexity of device stacks caused by necessity of a routable datagrams to be used. So far there is no well established technology to tackle this. However, RToUDP (Real-Time over UDP) technology is very promising in this sense.
- They draw in additional latency caused by datagram processing on routers up to L3. Fortunately high-end routers employ highly optimized data processing.
- Additional best-effort traffic is dragged in as the domain is no longer separated from the rest of the inter-LAN traffic.

Using proper DiffServ mechanisms, determinism can improve; DSCP classification and Congestion Management have to be used. This assumption applies provided that all network components in the given topology can be administrated. Presumably, the administration scope ends at enterprise boundary. However, it strongly depends on internal IT policies of each company.

4.3 WAN Topology – Public Networks

Let us consider that public network is a complement to enterprise network. In the VAN project, public networks are considered to provide interconnection among spatially distributed enterprise networks. These can be either telecommunication infrastructure or Internet backbones.

Seeing the fact that we usually cannot reach configuration of the QoS parameters of the network components, which are yet hidden from us, we have to rely on Service Layer Agreement (SLA) agreed on with the service provider.

5. TEST BED ARCHITECTURE AND DEVELOPMENT

Having introduced the requirements of industrial communication and QoS measures to increase determinism, provided qualitative analysis of improvement of determinism in IP based industrial networks.

The main objective of the test bed is to verify our presumptions and provide quantitative analysis of determinism improvement. The following chapters will describe the test bed architecture and its main features. Finally, first measurement results will be presented.

5.1 EB200 and EDD

The test bed is based on two EB200 (Evaluation Board 200). EB200 is a board accommodating ERTEC (Enhanced Real-Time Ethernet Controller). ERTEC chip is designed to perform communication based on PROFINET IO. EB200 is delivered with MS Windows XP drivers on top of which EDD (Ethernet Device Driver) operates. EB200 is PROFINET IO controller and 2-port switch. However, it can be also used as an end-device, as is done in our case.

EDD is service-oriented interfaces between EB200 and user applications. It provides NRT, SRT and IRT communication channels. All these interfaces are available at L2 except for RToUDP operating at L4.

NRT channel provides acyclic communication. User passes Ethernet payload together with configuration parameters to EDD. EDD provides transmission of the frame. Similar applies to reception. This channel is intended to be bound with the TCP/IP stack.

SRT channels provide both acyclic and cyclic communication. Acyclic channel does not differ significantly from the previous approach but using VLAN tagging and PROFINET IO EtherType 0x8892. This provides a certain improvement in determinism. Cyclic channel improves determinism a lot as this communication channel is scheduled without user intervention providing periodic process image updates.

IRT channel is the most efficient interface in terms of determinism. IRT is a time-driven interface with precise timing and scheduling of traffic in the Ethernet segment. All the components in the network have to be IRT aware.

However appreciated all these features are, the test bed utilizes only NRT channel. The reason for employing EB200 in our test bed is access to 100 MHz free running counter. The services of EDD used for sending and receiving Ethernet frames contain parameters TxTime and RxTime, respectively. These 4-byte parameters contain value of the free-running counter of the EB200 taken at a precisely defined moment of the frame transmission and reception.

5.2 Test Principle

We want to measure one-way latency between Port 1 of Board A and Port 1 of Board B. The latency is caused solely by propagation delay and the network infrastructure on the path.

Direct results are packet latency and packet-loss rate. Further results are inferred. Determinism changes can be observed while varying particular parameters. Figure 1 summarizes the parameters of interest as a result of the qualitative investigation.

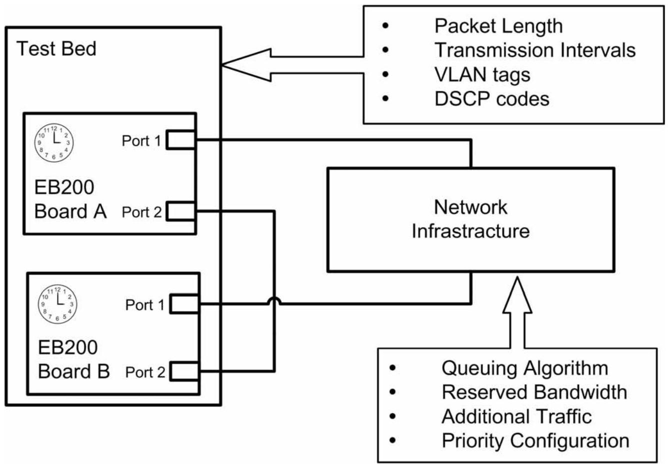

Fig. 1. Test Bed Principle

Each board has its own free running counter. We have chosen to use a comparative method of latency measurement. The procedure of the measurement is described in the following

- Frame 1 is compiled and sent via Port 1 of Board A. When being sent, timestamp Tx1 is stored. Frame 1 passes the tested infrastructure.
- Frame 2, is sent via Port 2 of Board A. When being sent, timestamp Tx2 is stored. Frame 2 passes only through a shortcut.
- Frame 2 is received by Port 2 of Board B. When being received, timestamp Rx2 is stored. This packet will always arrive first unless the tested infrastructure's latency is equal to 0.
- Frame 1 is received by Port 1 of Board B. When being received, timestamp Rx1 is stored.

Let us assume that we perform N measurements. Latency of n[th] measurement T_n, where $n = 1,\dots, N$, between sending Frame1 from Port 1 of Board A and receiving Frame 1 at Port 1 of Board B can be determined as follows. If there were no offset between the values of the free-running counters of the EB200 boards, the latency T_n could be evaluated according to formula

$$T_n = \Delta t_n = t_n^{Rx1} - t_n^{Tx1} \tag{1}$$

However, there is an offset of $t_{off,n}$, which needs to be compensated to baseline the values. The compensation is introduced as

$$\Delta t_n = t_n^{Tx1} - t_n^{Tx1} - \Delta t_{off,n} \tag{2}$$

The offset can be evaluated from Tx2 and Rx2 values as

$$\Delta t_{off,n} = t_n^{Rx2} - t_n^{Tx2} \tag{3}$$

Thus, the Δt is equal to

$$\Delta t = t_n^{Rx1} - t_n^{Tx1} - t_n^{Rx2} + t_n^{Tx2} \qquad (4)$$

All these values are available; hence, we can perform enumeration of the latency T_n. It is recommended that

$$t_n^{Tx2} - t_n^{Tx1} \leq 5\, ms \qquad (5)$$

to reach results with precision of dozens of nanoseconds because of the crystal drift. This prerequisite was fulfilled.

5.3 TestQoS Application Architecture and Design

The architecture of the test bed is introduced in Fig. 2. The whole test bed is located in a single PC equipped with two EB200 boards. The drivers are skipped for simplicity. However, boards communicate with the EDD located in a DLL via their drivers. EDD forms interface to the TestQoS application.

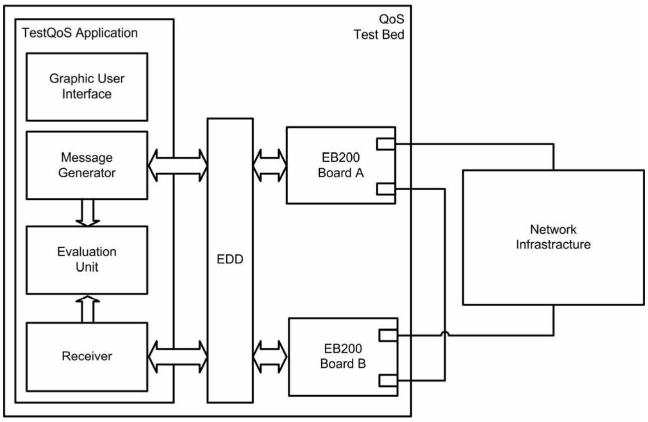

Fig. 2.Test Bed Architecture

TestQoS comprises two processes. Each process is bound to one EB200 board. The main process' task is to provide parameterisation of the test, its execution and evaluation of results. The auxiliary process' task is to store the received frames and their particular Rx1 and Rx2 timestamps and to pass them to the main process.

When the test starts, pairs of packets of predefined format and content are sent by Message Generator via Board A. The timestamps and the payloads are stored there too. Board B receives and stores the same information within the auxiliary process, which represents the Receiver. After the test ends information from the Receiver are passed via TCP/IP socket to the Evaluation Unit in the main process. The gathered data are evaluated in Evaluation Unit.

Evaluation Unit has to match packets according to sequence numbers and enumerate the latencies of each packet. Based on this, it can enumerate the jitter. If one of the packets within a pair is missing, the pair is declared lost.

User has the following setup possibilities:

- Selection of protocol and addresses. The user can use either PROFINET frames (L2), or UDP datagrams based on IP addressing and UDP ports (L4).
- Parameterisation of the test. The user can set number of sent packets, transmission interval, VLAN tag, DSCP code, and the payload content.
- Type of test output. Simplified results are visualized on screen while complete set of gathered data is stored in output file.

5.4 State of the Art in Development and First Results

Recently the test bed is capable of testing at L2; utilizing PROFINET frames. The application is capable of basic evaluation of statistics: number of transmitted packets, average latency, minimal latency, maximal latency, and jitter. Let us define a set of measured latencies as

$$T = \{T_n : n = 1, \cdots, N\}, \qquad (6)$$

where N is the number of the transmitted packets and T_n is the n-th latency. Then the average latency T_{avg} is calculated as

$$T_{avg} = \frac{1}{N} \sum_{n=1}^{N} T_n\ , \qquad (7)$$

and the jitter T_{jitter} can be calculated as

$$T_{jitter} = \max(T) - \min(T). \qquad (8)$$

First test (Test 1) was performed using L2 frames. As we needed the fastest possible application response, we used PROFINET IO frames belonging to ASRT (Asynchronous Soft Real-Time) class. Consequently, when we opened a channel with a proper filter, we were obtaining only ASRT frames. ASRT frame has EtherType 0x8892 (denoting PROFINET IO frame) and mandatory VLAN tag in Ethernet header. Ethernet payload begins with 2-byte Frame ID, in our case 0xFC00 (denoting high priority alarm frame). The measured infrastructure comprised a single 4-port office switch Gigabyte B49G. Packet length was 100 Bytes. The statistics was calculated from 1000 transmitted packets.

Second test (Test 2) was performed using L4 (UDP) datagrams. The infrastructure comprised two HP Procurve 1800-8G switches, and one Cisco 1812 router without QoS settings configured so far. Otherwise, the test was identical to the first one.

Table 1 summarizes the measurement results. Each test was performed both for offloaded infrastructure and loaded infrastructure using FTP traffic shaped at 4 MBps data flow.

We can clearly see that while the average latency T_{avg} does not differ significantly, jitter T_{jitter} tripled at switched topology and increased by several orders in the routed topology. Fig. 3 and Fig. 4 depict the latencies T_n for 100 transmissions for illustration.

Table 1. Summary of Measured Results

Topology	Switched (Test 1)		Routed (Test 2)	
Load	Off	4 MBps	Off	4 MBps
Avg. Latency [μs]	11.21	11.41	265.94	330.07
Min. Latency [μs]	11.07	11.06	262.13	54.05
Max. Latency [μs]	11.39	12.12	274.06	911.38
Jitter [μs]	0.32	1.06	11.93	857.33

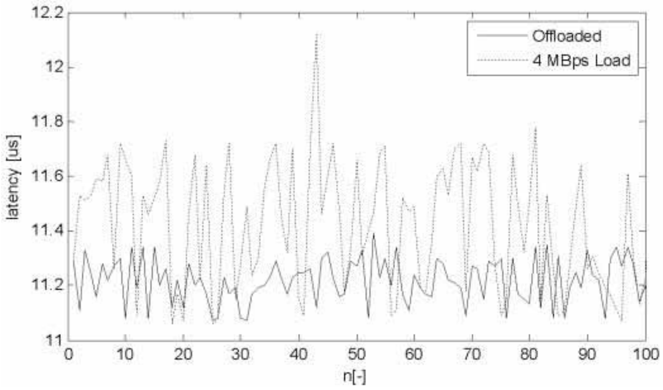

Fig. 3. Latencies measured on the offloaded and loaded switch

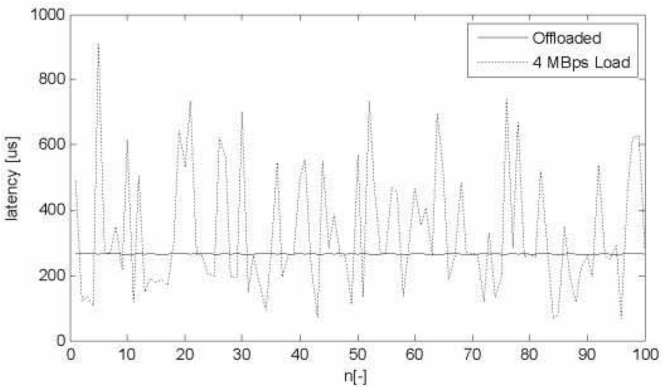

Fig. 4. Latencies measured on offloaded and loaded routed network

6. CONCLUSIONS

We have performed a detailed investigation into PROFINET IO fieldbus with a stress to real-time communication. According to requirements on extending the communication scope to multiple LAN segments, and thus migrating to routable protocols for run-time data exchange, we performed qualitative analysis of sources of additional latency.

We have specified and designed first version of a test bed giving us possibility to evaluate communication determinism in terms of latency and jitter with a 10 ns precision.

The expected result and contribution to the project is assessment of the upper boundaries of the QoS metrics given the network topology and the estimated additional best-effort traffic. Especially change of the upper boundary of latency when using the abovementioned QoS mechanisms is expected.

Consequently, if an industrial application will need a real-time communication over multiple LANs, the engineering of the application will be supported with configuration instructions and other setup recommendations to provide appropriate QoS behaviour.

7. ACKNOWLEDGEMENTS

This work has been supported in part by European Union 6th Framework Programme (Research Project FP6/2004/IST/NMP/2-016696 VAN Virtual Automation Networks) and by Brno University of Technology. Without kind support of the above-mentioned agencies and institutions the presented research and development would not be possible.

Special thanks belong to developers of Siemens AG Karlsruhe for their kind support in development of the TestQoS application.

8. REFERENCES

Collective (2005). *State of the Art and Trends in Safety, Security, Wireless technologies and real-time properties.* Virtual Automation Networks, available at: http://www.van-eu.eu/deliverables

B. P. Douglass (2000). *Doing Hard Time.* Addison Wesley,

M. Ford (1998). *Internetworking Technology Handbook.* Cisco Press

K. Lorentz and A. Lüder (2005). *IAONA Handbook – Industrial Ethernet 3rd Edition.* IAONA e. V.

J. Moyne and D. Tilbury. Performance Metrics for Industrial Ethernet. *The Industrial Ethernet Book*, **39**, available at http://ethernet.industrial-networking.com/articles/articledisplay.asp?id=1734