8

Wired and Wireless Reliable Real-Time Communication in Industrial Systems

Magnus Jonsson and Kristina Kunert
Halmstad University
Sweden

1. Introduction

In modern factory automation systems, data communication plays a vital role (Moyne and Tilbury, 2007). Different nodes like control systems, sensors and actuators can communicate over a wireless or wired industrial network. The data traffic generated is often scheduled for periodic transmission, where each single message or packet must arrive in time. For this real-time communication, methods have been developed to support communication services with a guaranteed throughput and delay bound for such periodic traffic, but merely under the assumption of error-free communication. However, the possibility for errors in the transmission still exists due to, e.g., noise or interference. A node receiving sensor values from a sensor in the system might then be forced to rely upon an older sensor value from the latest period, possibly leading to inaccuracies in control loops which can compromise the functioning of the system. In safety-critical systems, redundant networks or communication channels are frequently added to cope with errors, leading to more expensive systems. In this chapter, we will describe an alternative approach where erroneous data packets are retransmitted in a way that does not jeopardise any earlier stated real-time guarantees for ordinary transmissions. Using our framework, the reliability of real-time communication can be increased in a more cost-efficient way. We describe in this chapter an overview of our framework for reliable real-time communication, while details of our approach can be found in (Jonsson & Kunert, 2008; Jonsson & Kunert, 2008b; Jonsson & Kunert, 2009). In the light of the emerging use of wireless communication (Willig et al., 2005; Willig, 2008), the framework proves to be especially useful due to the high bit error rate inherent to the wireless medium. However, the framework is naturally also attractive for wired communication systems.

The rest of this chapter is organized as follows. In Section 2, several basic retransmission schemes are described in order to introduce basic concepts in retransmission protocols, while Section 3 summarizes some of the related work in the area of real-time communication and QoS (Quality of Service) provision by the usage of retransmissions. Section 4 introduces the concept of logical real-time channels used in our framework, followed by a description of our layered approach in Section 5. The retransmission scheme is explained in Section 6, while timing and scheduling analysis are briefly discussed in

Section 7. Section 8 is dedicated to results from our simulation studies before Section 9 concludes this chapter.

2. Basic retransmission schemes

Retransmission schemes are often referred to as ARQ (Automatic Repeat reQuest) protocols and can be divided into three categories (Lin et al., 1984; Tanenbaum, 2003):

- Idle RQ (also called stop-and-wait)
- Go-back-n
- Selective repeat

The latter two variants are of sliding window type (also called Continous RQ), while the first variant is a simpler kind of retransmission scheme. An Idle RQ protocol only allows for one packet at a time to be handled. After the packet has been transmitted, the transmitting node waits for an acknowledgement packet (ACK) from the other node, or for a retransmission timer to expire. A simplified finite state machine describing the function of the sending node using Idle RQ is shown in Fig. 1. Starting in the Idle state, a state transition to the "Wait for ACK" state is made when a request to send a packet is received from the layer above. The packet is sent immediately. If negative acknowledgement (NAK) is supported, retransmission can be explicitly requested by the other node. When an ACK packet is received by the transmitting node, the node returns to Idle state.

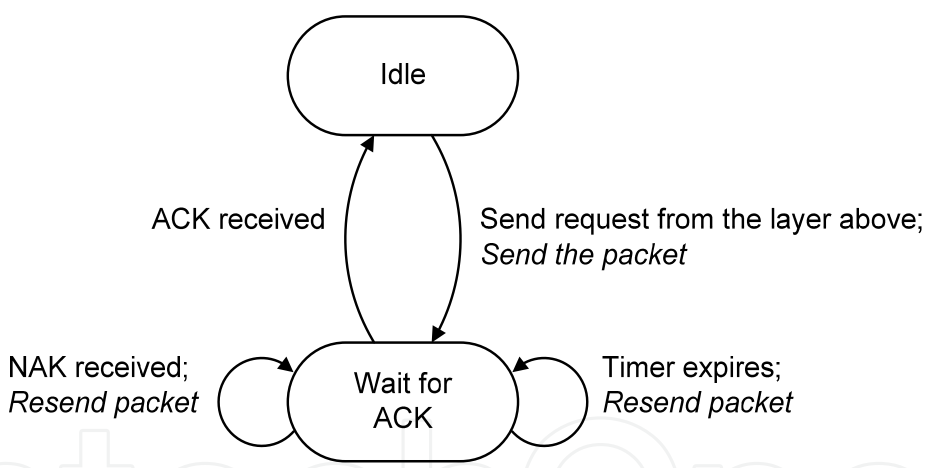

Fig. 1. Finite state machine describing the function of Idle RQ.

Idle RQ performs not very well in terms of link utilization at high bit-rates and/or long distances (large propagation delays), since a lot of the time then will be spent waiting for ACK packets. Continuous RQ protocols solve this problem by allowing several packets to be transmitted without having received ACK packets for the first packets already sent. The difference between go-back-n and selective repeat is in the retransmission strategy. When experiencing an erroneous packet transfer using go-back-n, the transmitter is restarting transmission starting with the packet after the last packet up to which all packets have been correctly transferred. In this way, even correctly transferred packets might be retransmitted. Using selective repeat instead, exclusively erroneous packets are retransmitted. In summary this means that selective repeat in most cases results in better link or network utilization, while go-back-n generally leads to simpler protocol implementations.

In order to make it possible to detect erroneous packets, a checksum or the like is always included in the packets. Furthermore, a sequence number is needed in both data packets, as an identifier of the data within the data stream, and ACK packets, to indicate which packet (or up to which packet) being acknowledged. Sequence numbers are also necessary in Idle RQ, since the possibility of duplicated data in case of a lost ACK packet might arise otherwise. Flow control, to avoid buffer overflow in the receiver, is solved in Idle RQ by only sending an ACK when the receiver is ready for the next transmission. Continuous RQ protocols use the concept of sliding window instead, where the size of the window corresponds to the receiver buffer size. This defines the width of the span of sequence numbers in which packets might be transmitted.

TCP (Transmission Control Protocol) (Postel, 1981) is probably the most well known ARQ protocol and is used in the transport layer to ensure reliable end-to-end transfer over the Internet. TCP is, however, not suitable when delay bounds and deterministic throughput guarantees need to be given. One obvious reason is the case when the same packet needs to be retransmitted indefinitely many times, and by that is delaying all other traffic. Additionally, the congestion control method included in TCP will decrease the packet transmission rate drastically whenever a retransmission is initiated. The problem with indefinite numbers of retransmissions, actually possible in all the three basic ARQ methods, can be solved by relaxing the demand on reliability just slightly and by using a truncated ARQ protocol instead, where the maximum number of retransmission attempts is limited. An example of a truncated ARQ method can be found in (Malkamäki & Leib, 2000). The usage of a truncated ARQ scheme, however, still demands real-time methods to be added, so that timely delivery of delay-sensitive traffic can be ensured.

3. Related work

The area of real-time communication is vast and well-studied, exactly as the research on the provisioning of QoS by retransmitting erroneous packets, as e.g. implemented by ARQ protocols. Although both topics have attracted interest in many years, a holistic view encompassing both approaches in order to *guarantee* timely treatment of real-time traffic and calculate the necessary delay bounds has remained relatively unstudied.

Amongst earlier work dealing with ARQ protocols for communication with real-time demands, solutions can be found which merely settle for a certain average performance, studying e.g. the average delay of delay-sensitive packets. In (Pejhan et al., 1996) it is shown, that using retransmissions is an effective way of error control for soft real-time traffic, i.e. traffic where occasionally missing the deadline is acceptable without serious implications on system performance. The article studies several different retransmission schemes used in connection with multicast protocols for real-time multimedia applications, which are typical examples of soft real-time applications. No hard real-time traffic is studied by the authors. Unfortunately, by the usage of statistical QoS parameters no deterministic calculations can be made and consequently no guarantee can be provided that no packet will miss its deadline.

The research presented in (Bilstrup et al., 2004) is suffering from a similar downside. The presented approach for a Bluetooth-based network provides a possibility of guaranteeing a certain degree of QoS. However, these guarantees are merely based on probabilistic calculations including average estimates of the quality of the wireless channel, and therefore a deterministic guarantee of the end-to-end delay bound cannot be given. Additionally, the presented statistical calculations are based upon an Idle RQ approach, often resulting in poorer link utilization than Continuous RQ-based schemes like ours.

An approach explicitly assuming hard real-time traffic has been published by Butt (Butt, 2006). His approach is similar to ours in that it aims to decrease the bit error rate experienced by hard real-time traffic by retransmitting erroneous packets until their deadline has been reached. However, similarly to (Bilstrup et al., 2004), also this retransmission scheme is based upon Idle RQ, hazarding the consequences of decreasing system performance. By limiting the suggested solution to a packet-level description, without developing an analysis of the queuing delay or providing details on the scheduling analysis, the provision of end-to-end delay-bound guarantees is not possible. In contrast, our approach with end-to-end real-time channels makes it possible to include an end-to-end delay bound analysis (including a queuing delay analysis) and by that provide guarantees of bounded end-to-end delay.

A way of providing end-to-end real-time guarantees including retransmissions can be found in (Giancola et al., 2002). The disadvantage compared to our approach is that the network capacity is analysed by the means of flow analysis. Recent work by Fan (Fan et al., 2009) has shown that flow analysis is not exploiting the available network capacity as fully as real-time scheduling does. Our framework is combining this kind of real-time scheduling analysis with the principle of retransmissions. Additionally, the exact timing details necessary to support hard real-time communication in industrial real-time systems, and used in our detailed analysis, are not included in the approach in (Giancola et al., 2002).

A related interesting solution to provide deadline guarantees for hard real-time traffic is deadline dependent coding, which on the bit level is combining error correcting codes and ARQ (Uhlemann et al., 2000; Uhlemann & Rasmussen, 2005). While this approach is presented as a major requirement for reliable wireless real-time communication, the articles treat merely point-to-point links, and no continuation towards a complete real-time scheduling analysis framework has been targeted for.

4. Logical real-time channels

The concept of logical real-time channels (RT channels), also referred to as RTVCs (Real-Time Virtual Channels), was introduced in (Ferrari & Verma, 1990). A RT channel is an abstraction of a traffic flow over a link or a network, where resources have been allocated to guarantee a certain minimum throughput and a bounded end-to-end delay. The basic parameters of a network layer RT channel i, following the notation in (Jonsson & Kunert, 2008b) (we follow this notation in the whole chapter), are the period (or minimum inter-arrival time), $P_{N,i}$, the maximum message length in bits each period, $L_{N,i}$, and the end-to-end delay bound, $D_{N,i}$, where N denotes the network layer. A RT channel can then be defined as $\tau_{N,i} = \{P_{N,i}, L_{N,i}, D_{N,i}\}$, or, in the case of a network, as $\tau_{N,i} = \{m_{s,i}, m_{d,i}, P_{N,i}, L_{N,i}, D_{N,i}\}$, where $m_{s,i}$ and $m_{d,i}$, denote the source node and the destination node, respectively. The source nodes are bound to behave according to the traffic specifications and must not violate them by e.g. sending more often.

To be able to state a deadline guarantee for a new RT channel, the worst-case delay must be analyzed for all existing RT channels plus the new one. Only if the delay bounds for this whole set of RT channels can be met, the new channel is accepted. When this procedure is

carried out online (during run-time) it is called admission control. The admission control system needs a real-time scheduling analysis to analyze the worst-case delays, which in turn must rely upon a deterministic behaviour of the network and/or the communication equipment. A shared-medium network (for example a bus or a ring network) must have a MAC (Medium Access Control) protocol with a deterministic behaviour (Malcolm & Zhao, 1995). As an example, the CSMA/CD (Carrier Sense Multiple Access Collision Detect) method used in shared-medium Ethernet is not deterministic since indefinite numbers of collisions can occur, caused by the randomness involved when resolving collisions. When two nodes send at the same time, detect a collision and backoff before retrying to send their data, they can randomly generate the same backoff time and thereafter experience a new collision.
For a point-to-point link, or a switched network made up of switches and point-to-point links, a deterministic queuing principle (or service discipline) must be used (Zhang, 1995). Even though FCFS (First Come First Served) is a deterministic queuing principle (Fan, 2005), there are queuing principles specially developed for real-time communication. Two examples of such are delay-EDD(Earliest Due Date) (Ferrari & Verma, 1990) and jitter-EDD (Verma et al., 1991). Both delay-EDD and jitter-EDD rely upon EDF (Earliest Deadline First) (Liu & Layland, 1973) scheduling, where the packets are sorted according to their deadlines. Regarding industrial communication systems appropriate for factory automation systems, the use of EDF packet scheduling in switched Ethernet networks to obtain real-time support has been proposed (Hoang et al., 2002; Hoang et al., 2002b).

5. A layered approach to reliable real-time communication

Our proposed retransmission scheme resides in the transport layer (see Fig. 2). It relies on RT channels supported by the network layer, but which are not offering any reliability in terms of retransmissions. The transport layer can, however, rely on timely delivery of packets by the network layer.

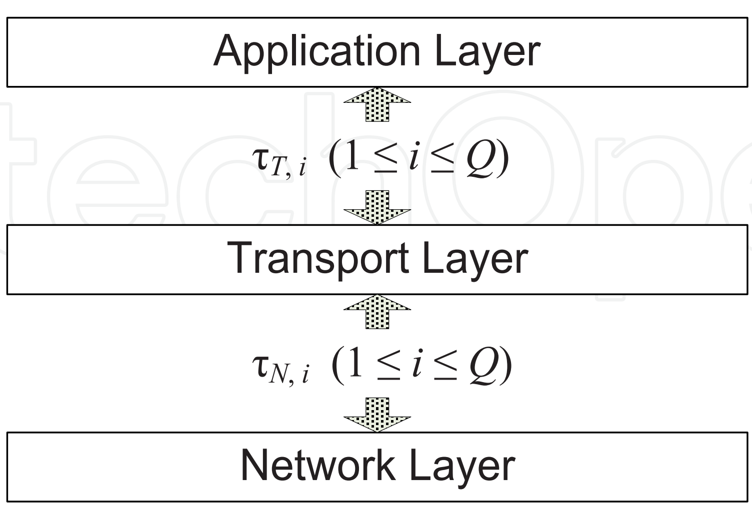

Fig. 2. The proposed retransmission scheme resides in the transport layer, relying on RT channels in the network layer and giving the service of reliable RT channels to the application layer.

The transport layer offers the service of reliable RT channels to the application layer. We define a reliable RT channel as a logical channel over which ordinary transmissions are always guaranteed to arrive in time, while retransmissions are made as long as no delay bounds of the ordinary transmissions are adventured. All parameters of a transport layer RT channel (reliable RT channel), $\tau_{T,i} = \{m_{s,i}, m_{d,i}, P_{T,i}, L_{T,i}, D_{T,i}\}$, where T stands for transport layer, can be mapped directly onto the corresponding parameters of a network layer RT channel, except for the delay bound parameter. We elaborate further on the delay bound parameter in the next section.

6. Our retransmission scheme

For the rest of the chapter, we will assume a point-to-point link over which we implement our retransmission scheme. Moreover, we assume that the packet queue in the transmitting node is an EDF queue, i.e. the packets are sorted according to their deadlines. To accomplish the possibility of having timing constraints on both ordinary transmissions and possible retransmissions, we divide the transport layer delay bound, $D_{T,i}$, for each RT channel into $T_{D\_ord,i}$ and $T_{D\_retr,i}$. The value of $T_{D\_retr,i}$ is set to the same value, D_{retr}, for all RT channels, where D_{retr} is a system parameter defining the time span allocated for retransmissions. An example is shown in Fig. 3, where a message consisting of three packets (to fit all $L_{T,i}$ bits) is transmitted. Packet three is not transferred correctly and is therefore retransmitted. Even the first retransmission attempt is incorrectly transferred and a second retransmission attempt is therefore initiated.

The second retransmission attempt is not acknowledged since no more retransmission attempts are allowed anyhow. This can of course be changed if the sending application really needs to know whether the transmission was successful or not. The ordinary packet transmissions follow the specification of the RT channel, but where the delay bound of the network layer RT channel is set to $D_{N,i} = T_{D\_ord,i}$. Network resources for ordinary transmissions are, in other words, allocated by the corresponding RT channel. Resources for retransmissions are, however, allocated through the use of special retransmission RT channels (using network layer RT channels) shared by all transport layer RT channels needing retransmissions. All retransmission RT channels will have their delay bounds set to $D_{retr,i} = D_{retr}$ (if several retransmission attempts are supported in the scope of D_{retr}, this is considered in the timing analysis), while $L_{retr,i}$ is set to the maximum packet length in the system. $P_{retr,i}$ is a system parameter indicating how often a retransmission RT channel can be used for retransmission of a packet belonging to an arbitrary ordinary RT channel.

The retransmission scheme can be seen as a truncated ARQ protocol, but where the timings of both ordinary transmissions and possible retransmissions are strictly regulated. Our retransmission scheme has not the drawbacks of Idle RQ, but allows for several consecutive transmissions without having received the ACK packets for the first transmitted packets yet. Normal flow control is not needed since RT channels are only admitted if all requirements can be guaranteed by the analysis discussed in the next section. Since the delays are bounded, it is possible to calculate the worst-case buffer population. Thereby, we can also ensure that we have enough buffer space and not admit RT channels otherwise.

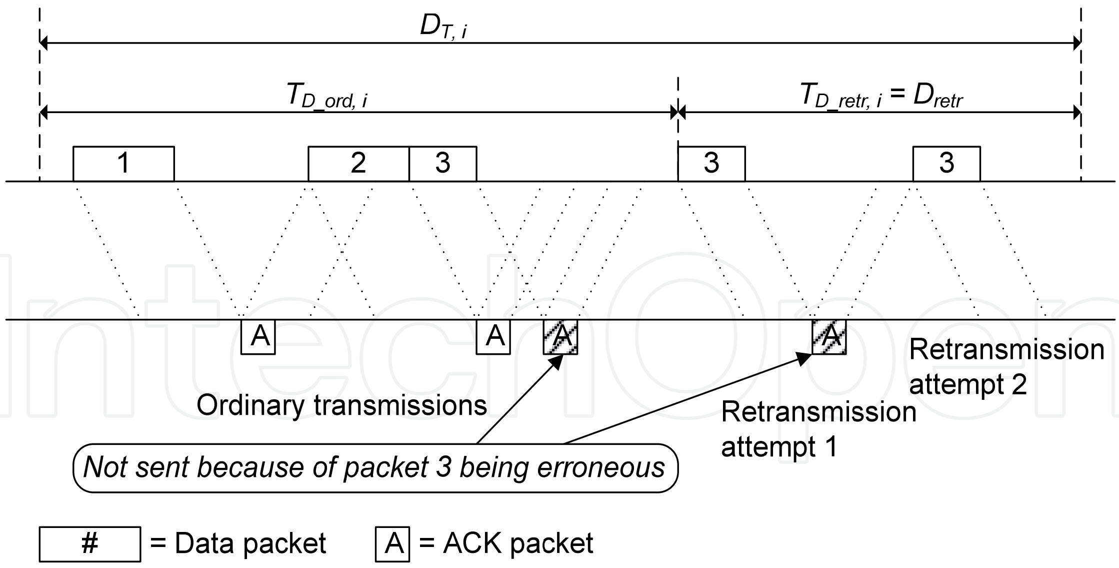

Fig. 3. The example shows how the end-to-end delay bound is divided into one bound for ordinary transmissions and one bound for a limited number of retransmission attempts.

7. Timing and scheduling analysis

The packets of a message belonging to a RT channel need to compete with packets belonging to other RT channels. As an example, the reason why packet 2 is not sent immediately after packet 1 in Fig. 3 might be that packets belonging to other RT channels, and with earlier deadlines, just arrived to the queue. To be able to analyse whether we can admit a new RT channel or not, we need to isolate the maximum message queuing delay (deadline) for each RT channel and perform a scheduling analysis for the EDF queue to see whether all such deadlines can be met. To do this we identify all other delays (constants or worst-case delays) and subtract them from the end-to-end delay bound. Examples of such delays are the propagation delay, blocking delay caused by a packet with a later deadline having to finish transmission, timing margins etc. For the retransmission RT channels, we need also to consider the maximum number of retransmission attempts per packet, $N_{attempt}$, to be supported. Detailed equations for those calculations are given in (Jonsson & Kunert, 2008b), while an extended analysis is given in (Jonsson & Kunert, 2009).
The scheduling analysis consists of two tests to be performed. The first test is to check that the aggregated utilization of all RT channels is not greater than 100%. The equation for calculating the utilization, quoted from (Jonsson & Kunert, 2008b), is:

$$U = \sum_{i=1}^{Q} \left(\frac{T_{x\_tot,i}}{P_{T,i}} \right) + \sum_{i=1}^{M} \left(\frac{T_{x\_retr,i}}{P_{retr,i}} \right) \tag{1}$$

where Q is the number of RT channels, M is the number of retransmission channels, $T_{x\_tot,i}$ is the total transmission time for all packets of a message including headers, while $T_{x\_retr,i}$ is the corresponding transmission time for a retransmission packet. In case of a successful first test, the second test is necessary. This second test comprises the following equation quoted from (Jonsson & Kunert, 2008b):

$$h(t) = \sum_{\substack{i\in[1,Q],\\ T_{d\_ord,i}\le t}} \left(1+\left\lfloor \frac{t-T_{d\_ord,i}}{P_{T,i}} \right\rfloor\right)\cdot T_{x\_tot,i} + \sum_{\substack{i\in[1,M],\\ T_{d\_retr,i}\le t}} \left(1+\left\lfloor \frac{t-T_{d\_retr,i}}{P_{retr,i}} \right\rfloor\right)\cdot T_{x\_retr,i} \quad (2)$$

where $T_{d\_ord,i}$ and $T_{d\_retr,i}$ are the scheduling (queuing) deadlines for ordinary RT channels and retransmission RT channels, respectively. The $h(t)$ function is a workload function adding together the transmission times of all message instances (from different periods) of all channels for those messages which have deadline instances less or equal to t. The constraint of the second test is that $h(t)$ must be less or equal to t for any value of t. The workload function has its origin from work done by Spuri (Spuri, 1996; Stankovic et al., 1998), later adapted for communication systems (Hoang & Jonsson, 2003; Hoang & Jonsson, 2003b). Details on how the test can be reduced to a discrete number of values of t to be checked are found in (Jonsson & Kunert, 2008b; Jonsson & Kunert, 2009; Stankovic et al., 1998). If not both the utilization test and the workload test are completed successfully, the new RT channel must be rejected.

8. Simulation analysis

With the aim of demonstrating the potential of our ARQ scheme we have implemented a MATLAB simulator and conducted two types of simulations in order to evaluate two different performance parameters of the system which are influenced by our retransmission scheme. The first type of simulation studies the utilization of the link by ordinary transmissions of hard real-time traffic (without including any retransmission packets). In order to calculate the utilization parameter, the simulation implements a schedulability analysis on the RT channel level, where an admission decision for generated RT channel requests is made. If a channel's delay bound can be guaranteed, while not violating already stated guarantees, the channel will be admitted; otherwise it will be denied access to the bandwidth. We will show and evaluate both the results for the case when using no retransmissions and for the case when our retransmission scheme is used. In the second type of simulation we observe the improvement of the message error rate which can be obtained by the usage of our retransmission scheme. These observations are made on the packet level, and in order to be able to relate these results to the utilization measurements from the first simulation type, the packets studied are those transmitted on the RT channels generated in this previous simulation.

We simulated both parameters typical for wired networks and those typical for wireless ones. Assumptions in common for both networks were full-duplex links and the same bit rate in each direction for reasons of simplicity. The maximum packet length was assumed to be 1000 bits in all cases, and the propagation delay in one direction experienced by the packets is assumed to be 1 μs. Each acknowledgement packet has a length of 100 bits, including a sufficient amount of redundant bits for error correction, resulting in a negligible error rate experienced by those packets. In order to better see the potential of our retransmission scheme, an implementation of piggyback acknowledgements was chosen as this generally leads to more efficient bandwidth utilization. The bit rates were set to be 100 Mbit/s and 10 Mbit/s for the wired and wireless links, respectively. The simulation study comprises three different cases which are presented in the following, and the traffic scenarios belonging to them are summarized in Tables 1, 2, and 3. Each traffic scenario defines four traffic classes, from which each RT channel is randomized (even distribution), and the parameters of the retransmission channel. Additionally, the number of retransmission attempts and the number of retransmission channels are indicated. In all figures the results obtained for a link without retransmissions are shown as a dashed line, while the results when using our retransmission scheme are shown by an unbroken line. In order to arrive at statistically reliable numbers, each point on the curves is, for each x-value, the average of 1000-1100 simulation runs over the length of 25-120 hyperperiods. (One hyperperiod starts at the point in time when all periods start simultaneously until they do so again.)

Case 1 - Wireless network			
Bit rate = 10 Mbit/s BER = 10^{-4} Number of retransmission attempts: 3 Number of retransmission channels: 10			
Traffic classes	*Period*	*Deadline*	*Message length*
1	20 ms	20 ms	4000 bits
2	40 ms	40 ms	4000 bits
3	80 ms	80 ms	4000 bits
4	160 ms	160 ms	4000 bits
Retransmission channel	*Period*	*Deadline*	*Packet length*
	10 ms	6 ms	1000 bits

Table 1. The specification parameters for the first simulated case of a wireless network.

The first case (see Table 1) evaluates the scenario of a wireless network (10 Mbit/s) with a bit error rate of 10^{-4}. All traffic classes have relatively long periods and deadlines, and the message length was set to 4000 bits, corresponding to four maximum sized packets. The ten retransmission channels have a period of 10 ms, a deadline of 6 ms and for each erroneous packet a maximum of three retransmissions was attempted. The simulation results illustrating the link utilization and the experienced message error rate are shown in Fig. 4 and Fig. 5, respectively. The results show a clear reduction of the message error rate by three orders of magnitude, while only reducing the utilization of the link by ordinary RT channels by about 12 percentage units at saturation.

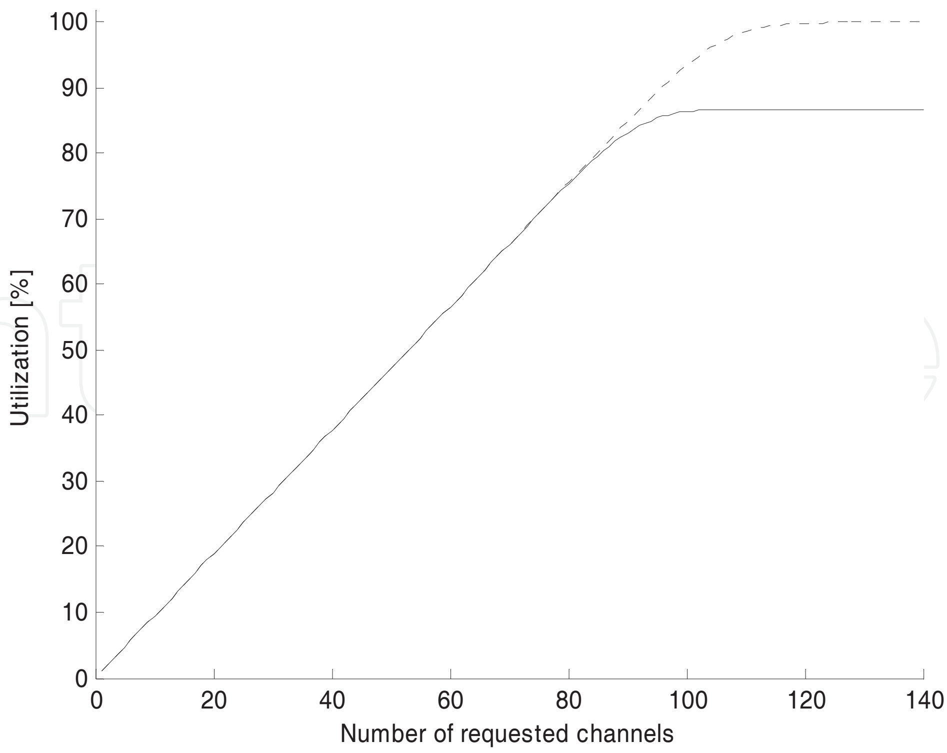

Fig. 4. Simulation results showing the link utilization both using no retransmission scheme (dashed line) and using our retransmission scheme (unbroken line).

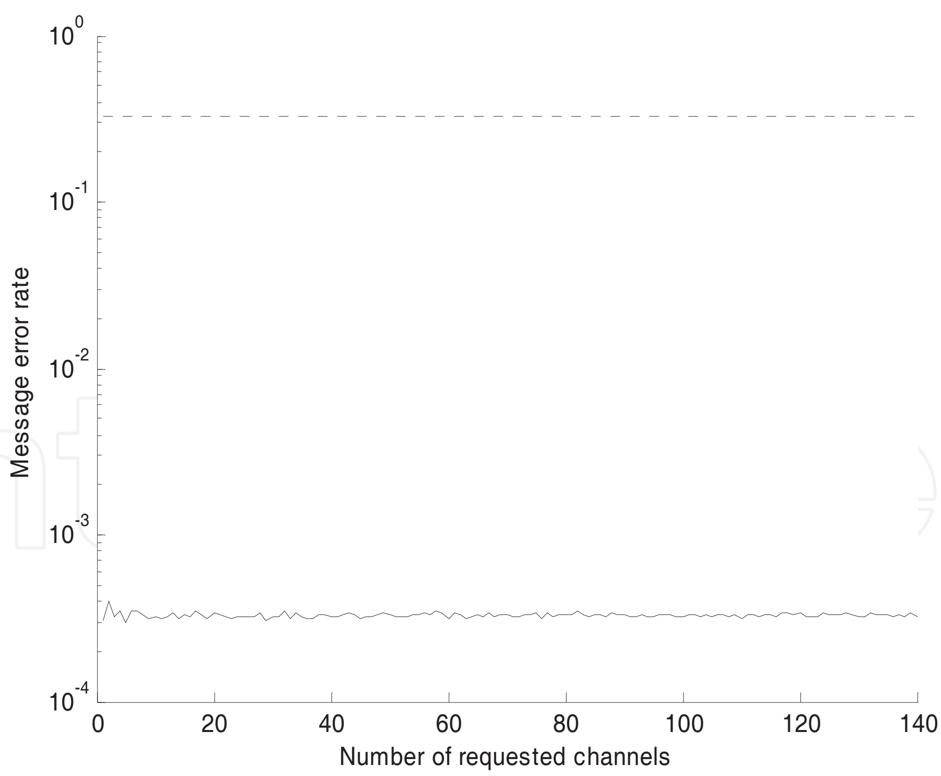

Fig. 5. Simulation results showing the experienced message error rate both using no retransmission scheme (dashed line) and using our retransmission scheme (unbroken line).

In case 2 (see Table 2) a second wireless scenario was simulated. While still keeping the bit rate at 10 Mbit/s, the bit error rate was lowered to 10^{-5} in order to see if we could improve this value even further by the help of our retransmission scheme. The periods and deadlines of the four different traffic classes were shortened, increasing the demand on the link, and additionally the deadline of the retransmission channel was shortened, making the scheduling of the retransmission channels more difficult. For a system with four retransmission channels and two retransmission attempts per packet, the results are shown in Fig. 6 and Fig. 7. The message error rate was improved by about four orders of magnitude, while the utilization penalty was no higher than approximately 5 percentage units. This combination of system and simulation parameters outperformed the ones from case 1 easily.

Case 2 - Wireless network			
Bit rate = 10 Mbit/s BER = 10^{-5} Number of retransmission attempts: 2 Number of retransmission channels: 4			
Traffic classes	*Period*	*Deadline*	*Message length*
1	10 ms	10 ms	4000 bits
2	20 ms	20 ms	4000 bits
3	40 ms	40 ms	4000 bits
4	80 ms	80 ms	4000 bits
Retransmission channel	*Period*	*Deadline*	*Packet length*
	10 ms	2 ms	1000 bits

Table 2. The specification parameters for the second simulated case of a wireless network.

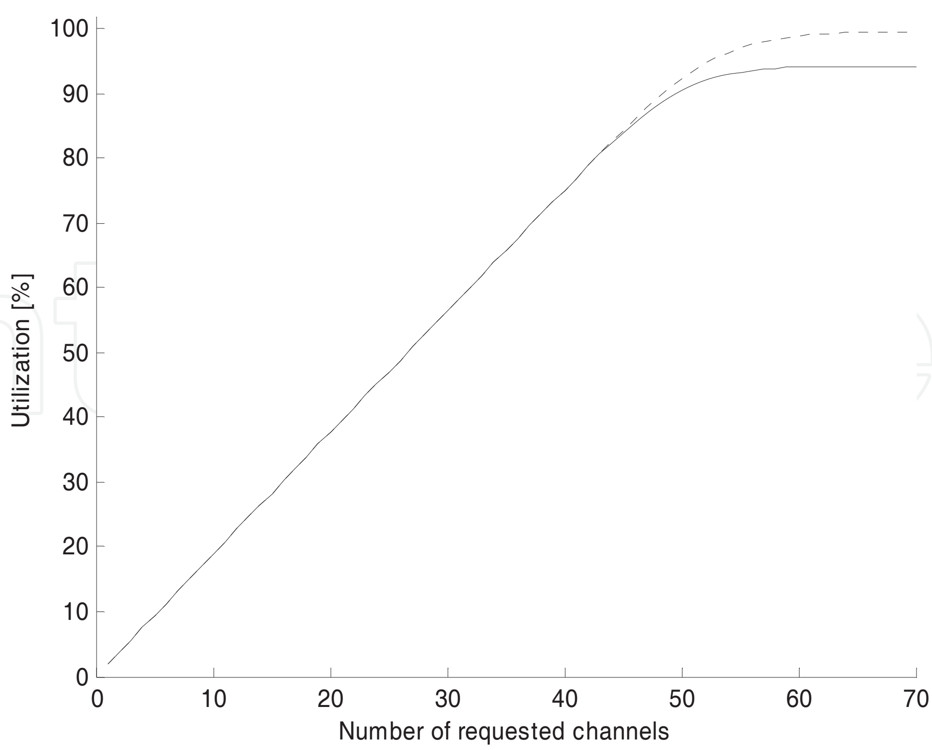

Fig. 6. Simulation results showing the link utilization both using no retransmission scheme (dashed line) and using our retransmission scheme (unbroken line).

Case 3 - Wired network			
Bit rate = 100 Mbit/s BER = 10^{-6} Number of retransmission attempts: 2 Number of retransmission channels: 3			
Traffic classes	*Period*	*Deadline*	*Message length*
1	1 ms	1 ms	20000 bits
2	2 ms	2 ms	20000 bits
3	4 ms	4 ms	20000 bits
4	8 ms	8 ms	20000 bits
Retransmission channel	*Period*	*Deadline*	*Packet length*
	1 ms	200 µs	1000 bits

Table 3. The specification parameters for the simulated case of a wired network.

In the last simulation setup, case 3 (see Table 3), the scenario of a wired network was implemented. While the bit rate was increased to 100 Mbit/s, the bit error rate was lowered to 10^{-6}. Both period and deadline of the traffic classes were shortened to values between 1 ms and 8 ms, and the packets grew to a size of 20000 bits.

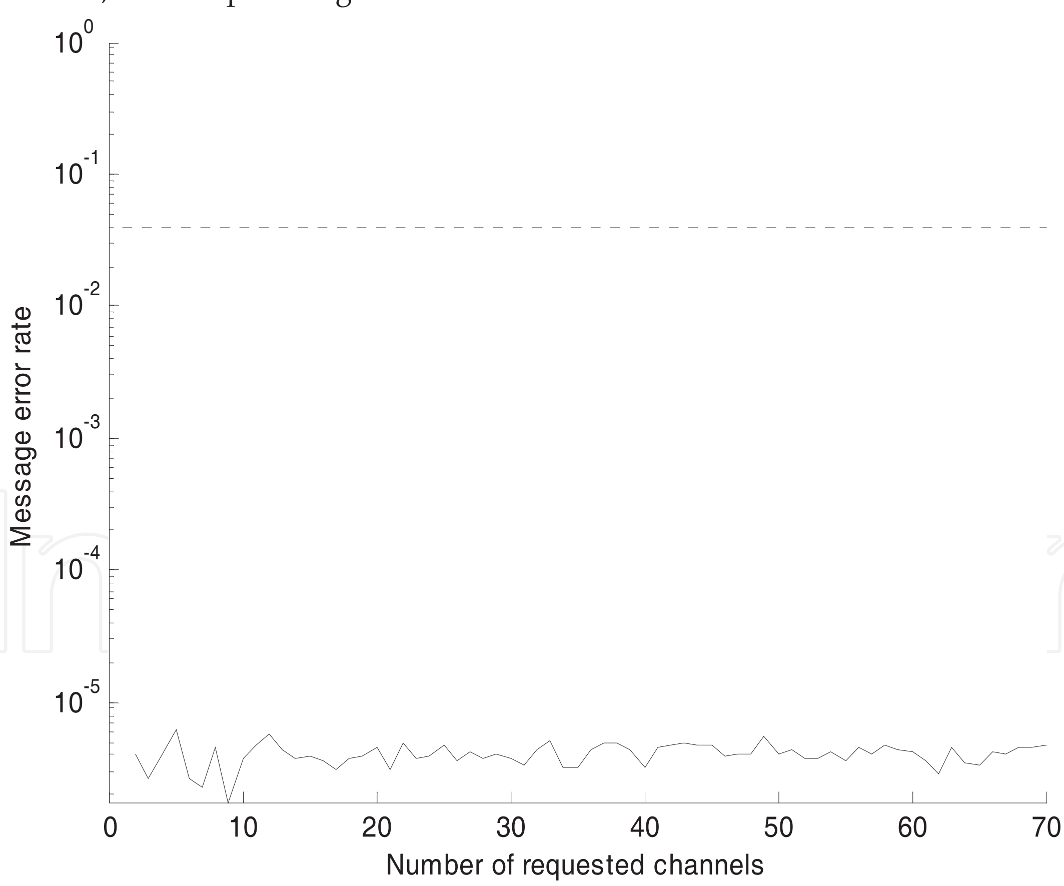

Fig. 7. Simulation results showing the experienced message error rate both using no retransmission scheme (dashed line) and using our retransmission scheme (unbroken line).

Period and deadline of the three retransmission channels, supporting two retransmission attempts per packet, were also shortened substantially. The results obtained from this simulation setup were as positive as can be seen in the figures. Here, the utilization penalty of about 5 percentage units was able to improve the message error rate experienced by the packets by approximately four orders of magnitude (see Fig. 8 and Fig. 9).

9. Conclusions

Industrial communication systems would benefit immensely if both their demands on reliability, in terms of correctly transferred information, and real-time constraints could be considered, especially thinking about the emerging use of wireless communication technology. In this chapter, we have shown how this could be made possible by using a retransmission scheme where retransmissions are only allowed if they do not put at risk any delay bounds of ordinary transmissions by making sure they are not violated. Timing and scheduling analyses ensure the timely delivery of both ordinary transmissions and retransmissions. By the help of simulations, we have shown that a reduction of the message error rate with several orders of magnitude is possible, while just sacrificing a small fraction of the bandwidth. Future work includes investigating further how to enhance the framework for complex networks instead of a point-to-point link.

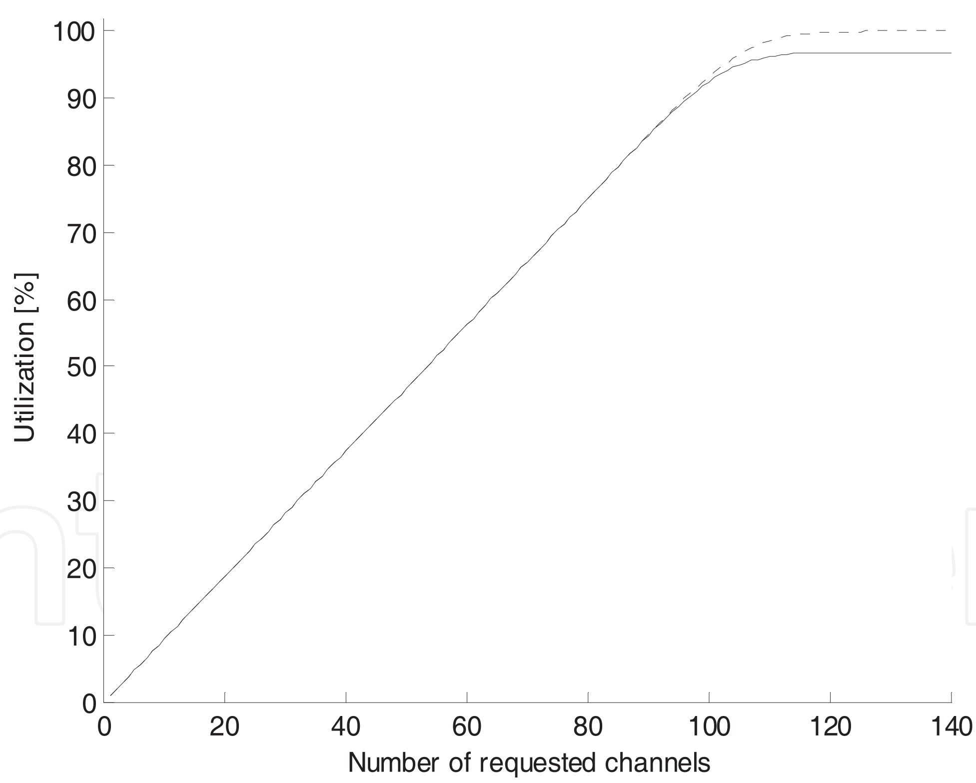

Fig. 8. Simulation results showing the link utilization both using no retransmission scheme (dashed line) and using our retransmission scheme (unbroken line).

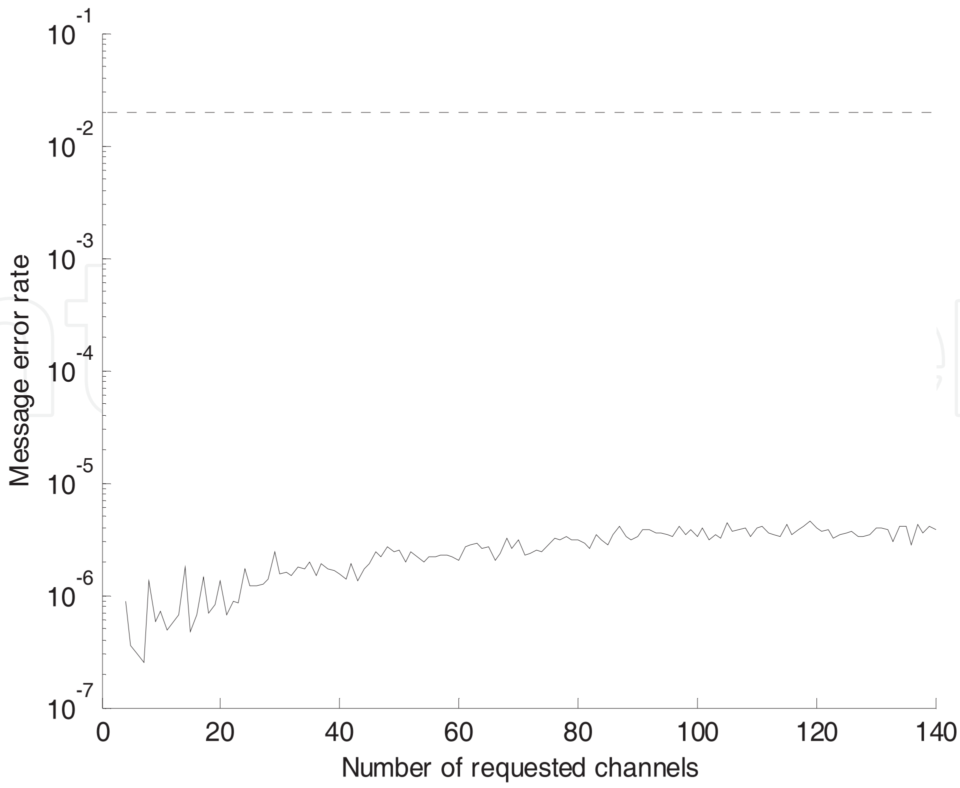

Fig. 9. Simulation results showing the experienced message error rate both using no retransmission scheme (dashed line) and using our retransmission scheme (unbroken line).

10. References

Bilstrup, U.; Sjöberg, K.; Svensson, B. & Wiberg, P.-A. (2004). A fault tolerance test enabling QoS in a Bluetooth piconet, *Proceedings of the 3rd International Workshop on Real-Time Networks (RTN 2004)*, June 2004, pp. 33-36.

Butt, M. M. (2006). Provision of guaranteed QoS with hybrid automatic repeat request in interleave division multiple access systems, *Proceedings of the 10th IEEE Singapore International Conference on Communication Systems (ICCS 2006)*, Oct. 2006, pp. 1-5.

Fan, X. & Jonsson, M. (2005). Guaranteed real-time services over standard switched Ethernet, *Proceedings of the 30th Annual IEEE Conference on Local Computer Networks (LCN 2005)*, Sydney, Australia, Nov. 2005.

Fan, X.; Jonsson, M. & Jonsson, J. (2009). Guaranteed real-time communication in packet-switched networks with FCFS queuing, *Computer Networks*, Vol. 53, No. 3, Feb. 2009, pp. 400-417.

Ferrari, D. & Verma, D. C. (1990). A scheme for real-time channel establishment in wide-area networks, *IEEE Journal of Selected Areas in Communications*, Vol. 8, No. 3, Apr. 1990, pp. 368-379.

Giancola, G.; Falco, S. & Di Benedetto, M. G. (2002). A novel approach to error protection in medium access control design, *Proceedings of the 4th International Workshop on Mobile and Wireless Communications Networks (MWCN 2002)*, Sept. 2002, pp. 337-341.

Hoang, H.; Jonsson, M.; Hagström, U. & Kallerdahl, A. (2002). Switched real-time Ethernet with earliest deadline first scheduling - protocols and traffic handling, *Proceedings of the Workshop on Parallel and Distributed Real-Time Systems (WPDRTS 2002) in conjunction with the International Parallel and Distributed Processing Symposium (IPDPS 2002)*, Fort Lauderdale, FL, USA, April 2002.

Hoang, H.; Jonsson, M.; Kallerdahl, A. & Hagström, U. (2002b). Switched real-time Ethernet with earliest deadline first scheduling - protocols and traffic handling, *Parallel and Distributed Computing Practices*, Vol. 5, No. 1, 2002.

Hoang, H. & Jonsson, M. (2003). Switched real-time Ethernet in industrial applications - asymmetric deadline partitioning scheme," *Proceedings of the 2nd International Workshop on Real-Time LANs in the Internet Age (RTLIA 2003) in conjunction with the 15th Euromicro Conference on Real-Time Systems*, Porto, Portugal, July 2003.

Hoang, H. & Jonsson, M. (2003b). Switched real-time Ethernet in industrial applications - deadline partitioning, *Proceedings of the Asia-Pacific Conference on Communications (APCC 2003)*, Penang, Malaysia, Sept. 2003, pp. 76-81.

Jonsson, M. & Kunert, K. (2008). Reliable hard real-time communication in industrial and embedded systems, *Proceedings of the 3rd IEEE International Symposium on Industrial Embedded Systems (SIES)*, Montpellier, France, June 2008.

Jonsson, M. & Kunert, K. (2008b). Meeting reliability and real-time demands in wireless industrial communication, *Proceedings of the 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008)*, Hamburg, Germany, Sept. 2008.

Jonsson, M. & Kunert, K. (2009). Towards reliable wireless industrial communication with real-time guarantees, *IEEE Transactions on Industrial Informatics*, Vol. 5, No. 4, Nov. 2009, pp. 429-442.

Lin, S.; Costello, D. & Miller, M. (1984). Automatic-repeat-request error-control schemes, *IEEE Communications Magazine*, Vol. 22, No. 12, Dec. 1984, pp. 5-17.

Liu, C. L. & Layland, J. W. (1973). Scheduling algorithms for multiprogramming in a hard-real-time environment, *Journal of the ACM*, Vol. 20, No. 1, pp. 46-61, Jan. 1973.

Malkamäki, E. & Leib, H. (2000). Performance of truncated type-II hybrid ARQ schemes with noisy feedback over block fading channels, *IEEE Transactions on Communications*, Vol. 48, No. 9, Sept. 2000, pp. 1477-1487.

Malcolm, N. & Zhao, W. (1995). Hard real-time communication in multiple-access networks, *Real-Time Systems*, Vol. 8, No. 1, 1995, pp. 35-77.

Moyne, J. R. & Tilbury, D. M. (2007). The emergence of industrial control networks for manufacturing control, diagnostics, and safety data, *Proceedings of the IEEE*, Vol. 95, No. 1, Jan. 2007, pp. 29-47.

Pejhan, S.; Schwartz, M. & Anastassiou, D. (1996). Error control using retransmission schemes in multicast transport protocols for real-time media, *IEEE/ACM Transactions on Networking*, Vol. 4, No. 3, June 1996, pp. 413-427.

Postel, J. (1981). Transmission control protocol - darpa internet program protocol specification, *IETF RFC 793*, Sept. 1981.

Spuri, M. (1996). Analysis of Deadline Scheduled Real-Time Systems, *Technical Report RR No. 2772, INRIA*, France, 1996.

Stankovic, J. A.; Spuri, M.; Ramamritham, K. & Buttazzo, G. C. (1998). *Deadline scheduling for real-time systems - EDF and related algorithms*, Kluwer Academic Publishers, Boston, MA, USA, 1998.

Tanenbaum, A. S. (2003). *Computer Networks*. Fourth edition, Prentice Hall, Upper Saddle River, NJ, USA, 2003, ISBN 0-13-066102-3.

Uhlemann, E.; Wiberg, P.-A.; Aulin, T. M. & Rasmussen, L. R. (2000). Deadline dependent coding - a framework for wireless real-time communication, *Proceedings of the 7th International Conference on Real-Time Computing Systems and Applications (RTCSA 2000)*, Cheju Island, South Korea, Dec. 2000, pp. 135-142.

Uhlemann E. and Rasmussen, L. K. (2005). Incremental redundancy deadline dependent coding for efficient wireless real-time communications, *Proceedings of the 10th IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2005)*, Catania, Italy, Sept. 2005, Vol. 2, pp. 417-424.

Verma, D ; Zhang, H. & Ferrari, D. (1991). Guaranteeing delay jitter bounds in packet switching networks, *Proceedings of TriComm'91*, Chapel Hill, NC, USA, Apr. 1991, pp. 35-46.

Willig, A.; Matheus, K. & Wolisz, A. (2005). Wireless technology in industrial networks, *Proceedings of the IEEE*, Vol. 93, No. 6, June 2005, pp. 1130-1151.

Willig, A. (2008). Recent and emerging topics in wireless industrial communications: a selection, *IEEE Transactions on Industrial Informatics*, Vol. 4, No. 2, May 2008, pp. 102-124.

Zhang, H. (1995). Service disciplines for guaranteed performance in packet-switching networks, *Proceedings of the IEEE*, Vol. 83, No. 10, Oct. 1995, pp. 1374-1396.