Ring Stability of the PROFIBUS Token Passing Protocol over Error Prone Links

Accepted for Publication in IEEE Transactions on Industrial Electronics

Andreas Willig, Adam Wolisz

Abstract— **The PROFIBUS is a well known and widely used fieldbus. On the MAC layer it employs a token passing protocol where all active stations form a logical ring on top of a broadcast medium. This protocol is designed to deliver real-time data transmission services in harsh, industrial environments. A necessary prerequisite for timeliness and Quality of Service (QoS) is the ring membership stability of the logical ring in the presence of transmission errors, since only ring members are allowed to transmit data. In this paper the ring membership stability under high error rates and using different error models is analyzed. The choice of the error behavior is in turn inspired by properties of possible future transmission technologies, e.g. wireless LANs. It is shown that the protocol has serious stability problems. To attack these problems, two changes to the protocol and its parameters are proposed, which can be implemented in a purely local manner, and show that they significantly improve ring stability.**

Keywords— **PROFIBUS, Link Errors, Ring Stability, QoS**

I. Introduction

THE PROFIBUS is a well known, standardized and widely used fieldbus ([1], with some corrections in [4], the European standards document is [8]). It is designed to deliver real-time services in harsh, industrial environments. The PROFIBUS addresses the real-time requirements on the MAC layer by using a token passing protocol similar to that used in IEEE 802.4 Token Bus. In these protocols a logical ring is built on top of a broadcast medium, using special control frames for ring maintenance, however, the maintenance mechanisms differ: IEEE 802.4 uses a contention-based mechanism for including stations into the ring, while PROFIBUS uses polling. In both protocols only members of the logical ring are allowed to transmit data. Thus, one important goal of the PROFIBUS protocol is that all stations, who wants to be, are member of the ring and remain so. The degree to which this is achieved is referred to as *ring stability* within this paper, and can be captured with different metrics. Since the ring membership is maintained by exchanging special control frames, the ring stability can be affected by loss of these frames due to transmission errors. Since data transmission is restricted to ring members it is clear that ring stability strongly impacts the achievable Quality of Service (QoS) and system reliability.

In this paper we study the ring stability of the PROFIBUS protocol in the presence of transmission errors and under two different error models. The choice of the error models and their parameters is motivated by properties of possible future transmission technologies, e.g. wireless LANs. We show that the protocol has serious stability problems under higher error rates and that ring stability is sensitive to the "burstiness" of errors. We propose two improvements of the protocol and its parameters, which require no modifications in frame formats and are interoperable with the unchanged protocol rules. For these improvements it is shown that they yield a significant increase in ring stability.

The behavior of PROFIBUS in the presence of transmission errors or its ring membership behavior / ring stability is to the author's best knowledge not covered in the literature. Most analyses of the PROFIBUS real-time capabilities (e.g. [6], [5, chap. 3, chap. 5]) allow for sporadic transmission errors by taking retransmissions into account, however, the influence of transient times where a station is involuntarily not a ring member is not considered. For the IEEE 802.4 Token Bus in [2] it is investigated using analytical techniques and measurements, how bursty errors affect the token passing process, and how this in turn affects the mean token passing time and, more important, the mean token rotation time. For the PROFIBUS some results on local stability metrics (see below) are available in [11], and an analytical model for the ring membership behavior in PROFIBUS is presented in [12]. This model validates our simulation results, but is not included here due to lack of space.

The paper is structured as follows: in Section II we first give a short overview of the PROFIBUS token passing and ring maintenance algorithm and then explain the major causes for ring instability in the presence of transmission errors. In Section III we define the metrics for ring stability used throughout this paper. In Section IV we present simulation results for the PROFIBUS protocol which indicate that under higher bit error rates there are serious stability problems, and furthermore that the protocol is sensitive to the error burstiness. In Section V we describe two protocol improvements and show with simulations that they significantly increase ring stability. Finally, in Section VI we give the conclusions.

II. Token Passing and Ring Maintenance

In this Section we give a brief description of the token passing and ring maintenance protocol, followed by defining some notions regarding ring membership and by an explanation of scenarios that may lead to station losses from the ring.

This work was partially sponsored by the Deutsche Forschungsgemeinschaft (DFG).

A. Willig and A. Wolisz are with the Telecommunication Networks Group, Department of Electrical Engineering, Technical University Berlin, Germany. E-mail: {willig,wolisz}@ee.tu-berlin.de .

A. Protocol

The PROFIBUS token passing protocol works on top of a broadcast medium. A logical ring is formed by ascending station addresses. The address space is small, a station address is in the range of 0 to 126. Every station (denoted as TS: This Station) knows by the ring maintenance mechanism the address of its logical successor (NS: Next Station) and its logical predecessor (PS: Previous Station). If TS receives a valid token frame with TS as destination address, it checks whether it is sent by its PS. If so, the token is accepted, otherwise the frame is discarded. In the latter case, if the same token frame is received again as the very next frame, the token is accepted, the token sender is registered as new PS and the *list of active stations* (LAS) is updated, see below. In any case, after accepting the token TS determines its *token holding time* THT (according to a simplified variant of the timed token protocol with target token rotation time TTRT) and is allowed to send some data during the THT. If there is no data anymore or THT expires, TS is required to pass the token to NS by sending a token frame. This must be done even if TS is the only ring member (NS = TS = PS), and TS must accept the token in the same way as if PS $\neq$ TS. After sending a token frame, TS listens on the medium for some activity. This can be the reception of a valid frame header (indicating that NS has accepted the token) or reception of some erroneous transmission. However, TS listens on the medium only for a short time (called *slot time*, T_{SL}) which is typically chosen very tight, e.g. in the range of 100 μs to 400 μs. If this time passes without any medium activity, the token frame is repeated. If there is again no activity, and a third trial is also unsuccessful, NS is assumed to be dead and TS determines the next station in the ring (i.e. the successor of NS), makes this the new NS and tries to pass the token to it, following the same rules. The new station can be determined from the LAS, which is gathered by the ring maintenance mechanism, as explained below. If TS finds no other station, it sends a token frame to itself.

A special protocol rule is the following: TS must read back from the medium bit by bit all token frames it transmits ("hearback"), in order to detect a defective transceiver and to resolve collisions (see below). If TS encounters a difference the first time, it waits for some response (which indeed may occur due to undetected errors in the token frame, see below). If there is no activity on the medium it repeats the token frame. If TS again encounters a difference, it discards the token immediately and removes itself from the ring, behaving as newly switched on and "forgetting" all knowledge previously obtained.

The ring maintenance mechanism works by two different means. First, if a station is newly switched on, it is required to listen passively on the medium, until it has received two successive identical token cycles and thus has a valid view on the whole logical ring (referred to as "listen token" state). During this time it is not allowed to send or answer to data frames or to accept the token. Every station address found in a token frame belonging to this two cycles is included into the LAS. After building a valid view the station can enter the ring if another station passes the token to it. The second rule requires every station to inspect every correctly received token frame and to include the source and destination address into the LAS. An important rule here is the following: if TS feels itself as already included in the logical ring and reads a token frame, where TS is "skipped" (i.e. the address of TS lies truly within the address range spanned by sender and receiver of the token frame) it removes itself from the ring and behaves as newly switched on.

In order that another station can pass the token to a station newly switched on, every station a maintains a *gap list* (GAPL), containing all station addresses between a and its NS b. A station a is required to periodically poll all addresses in its GAPL by sending a "Request-FDL-Status" frame to a single address c and waiting one slot time for an answer, which indicates c's current status (ready / not ready for the ring). A station which tries to detect two identical token cycles will respond with a "not ready" status. Within every token cycle a polls at most one station address in its GAPL. If a station in the GAPL responds as "ready", a will change its NS, shorten its GAPL, update its LAS, and then send a token frame to the new station. The period for scanning the GAPL is created by a special timer ("gap timer"), which is set as an integral multiple ("gap factor", the standard requires values between 1 and 100) of the target token rotation time.

For leaving the ring it suffices to just stop all transmissions. In this case PS will detect the station loss when unsuccessfully trying to pass the token to TS.

A special mechanism is used for the very first ring initialization or to handle token loss due to system crash of the current token owner: every station listens permanently on the medium. Every time the medium goes idle, TS starts a special timer, the *timeout timer*, which is resetted when the medium goes busy. If the timer expires (no transmission on the medium for some time), TS "claims the token", i.e. it starts with behaving as the current token owner and performs some frame transmission: it sends data frames or passes the token to its current NS. If TS was not in the listen token state when the timeout timer expires, there is no change in its internal state, specifically in its LAS, NS and PS. In the other case, since the station has not yet a valid view on the ring, it assumes the ring to be empty and itself being the only member of LAS.

The timeout value linearly depends on the station's address. This can lead to collisions, and the hearback feature is necessary to resolve them. One situation where collisions can occur is the following: consider that in an empty ring two stations are newly switched on at different times, such that their timeout timers expire simultaneously. When both stations start transmitting token frames, the resulting collision induces hearback errors. Both stations retire from the ring and stop transmissions, while simultaneously starting their timeout timers. Because of the different station addresses the timers expire at different times, and now a valid ring can be built up without further collisions.

For data transmission a semi-reliable protocol is used, with a bounded number of retransmissions. For distinction between new frames and retransmitted frames the alternating bit protocol is used. The transmission of a data frame and its retransmissions is performed at once, i.e. it is not preempted by other data frames or token passing.

B. Ring-Membership related Definitions

We introduce two definitions: a *station loss event* (or simply *station loss*) denotes the single point in time where a station detects its loss from the ring and discards all of its knowledge previously obtained, especially the LAS. After a station loss a station behaves as newly switched on.

A *station outage time* denotes the time duration needed for a lost station to become a ring member again (by expiration of its timeout timer or by being reincluded).

C. Major Causes for Loosing Stations from the Ring

By analysis of the protocol specification and of our simulator traces, we have identified three different ways of how a station can get lost. The first scenario is due to the fact that the token frame has no checksum. It is protected only with a parity bit, startbit and stopbit for every single byte (every byte is transmitted serially with 11 bits). Thus there is a non-negligible probability that a token frame can be corrupted such that no station except the sender (by the hearback feature) will recognize an error[1]. Consider now the case of two stations with addresses a and b respectively, where $a < b$ holds to ease presentation. If a sends a token frame to b where the destination address is corrupted and equal to c with $a < b < c$, b considers itself being skipped and immediately removes itself from the ring, behaving as newly switched on. If a retransmits the token, b has not yet built a valid LAS and does not accept the token. After another token frame a considers b as lost from the ring, since again b is not allowed to answer. We refer to this as "error skipping".

The other scenarios are due to the presence of the hearback feature: when station a experiences hearback errors in two successive trials to send a token frame it gets lost from the ring (i.e. forgets its LAS). When the token frames are detected as faulty by all other stations, then the medium is idle until the timeout timer of the station with the lowest address expires. Within this scenario two cases can be distinguished: a has the lowest station address w.r.t. current ring members or not (we assume that a has negligible initialization delay). If a has the lowest address, it is the timeout timer of a that expires. Since there was no transmission during the idle time and a has forgotten its LAS, a now thinks it is alone in the ring and sends a token frame to itself. Then all other stations remove from the ring, feeling themselves skipped. We refer to this scenario as "ring jacking". If a has not the lowest address, the remaining ring keeps alive and a is reincluded later. We refer to this as "hearback removal".

To summarize, the mechanisms for loosing stations are as follows:

- Station a gets lost due to error skipping.
- Station a experiences a hearback removal.
- Station a gets lost because another station b with the lowest address performs ring jacking.

[1]This probability can be lower bounded by the probability P_R, that exactly two bit errors occur within the same byte, which cannot be detected by the parity scheme. The token frame is $3 \cdot 11 = 33$ bits long. Assuming that bit errors are independent and occur with fixed probability p, P_R is then given by $P_R = \frac{168}{1056} \cdot b(2; 33, p)$ where $b(k; n, p) = \binom{n}{k} p^k (1-p)^{n-k}$ is the distribution function of the binomial distribution. We have used the fact that from 1056 ways to distribute two errors over 33 bits only 168 of these lead to undetectable errors, all others are detected. With $p = 0.001$ we have $P_R \approx 0.00008$.

III. Ring Stability Metrics

In this Section the metrics for ring stability used throughout this paper are defined. They belong to the class of "global" stability metrics, which are focused to the whole logical ring, while for "local" metrics (not covered in this paper) the focus is on a single station. The global metrics can be computed also with the analytic model described in [12].

Let K be the number of stations and $\{N(t)\}_{t \in \mathbb{R}}$ a set of integer-valued random variables, denoting the number of stations that are members of the ring at time t (more precisely: which consider themselves being member). We have $0 \leq N(t) \leq K (t \in \mathbb{R})$, and $N(t)$ changes only at discrete points in time, by the operation of the protocol. We assume that all stations want to be member of the ring all the time. We introduce the following global metrics for ring stability:

- Consider at time t_0 we have $N(t_0) = K$ and $\lim_{\epsilon \to 0, \epsilon > 0} N(t_0 - \epsilon) < K$, i.e. the ring has just been completed at t_0. Furthermore let $t_1 = \inf\{t > t_0 : N(t) < K\}$ and $C = t_1 - t_0$. The random variable C denotes the time duration that the ring is complete, before the next time it looses a station. We are interested in its mean value $\bar{C}$ and distribution function $C(s) = \Pr[C \leq s]$. The "dual" of C, i.e. the time needed to re-enter the state of a full ring after the full ring breaks, is not covered here.
- Mean number of stations in the ring during interval $[0, t]$:

$$\bar{N}(t) = \frac{1}{t} \int_0^t N(s) ds,$$

additionally we are interested in the limiting mean value $\bar{N} = \lim_{t \to \infty} \bar{N}(t)$, which is assumed to exist and approximated by evaluating $\bar{N}(t)$ for some large t.

- Fraction of time where not all stations are member of the ring during time interval $[0, t]$:

$$\bar{M}(t) = \frac{1}{t} \int_0^t \mathbf{1}_{[0, K-1]}(N(s)) ds$$

where $\mathbf{1}_{[a,b]}(x)$ is the indicator function for the set $\{t \in \mathbb{R} : a \leq t \leq b\}$. Additionally we are interested in the limiting fraction $\bar{M} = \lim_{t \to \infty} \bar{M}(t)$.

Some important local metrics for a single station i are the following: the distribution of times between station loss events, the duration of station outages and the overall fraction of time that i is not member of the ring. Some simulation results for these metrics can be found in [11].

IV. Simulation Results

In this Section we present simulation results for the global stability metrics defined in Section III. We have built a detailed simulation model using the CSIM simulation library [3]. Information about the model can be found at *http://www-tkn.ee.tu-berlin.de/research/results.html.* This model includes parts of the PROFIBUS link layer, the PROFIBUS MAC protocol and a shared medium. In the shared medium all attached stations including the transmitter see the same signals and bits, thus the transmitter can perform proper hearback. All timing properties pertaining to the behavior of the medium (e.g. bit times, required idle times), and additionally a station's delay in processing received frames and generating answers are considered within the model. The simulator is validated by code inspection, comparison of generated frame sequences with expected frame sequences, and by the fact that for $\bar{N}(3600)$ and $\bar{M}(3600)$ the results are very close to those of an analytical model based on a markov chain description of the ring behavior presented in [12].

In the first set of simulations there are 10 stations without any external load, thus only token frames and Request-FDL-Status frames occur. This restriction was introduced to highlight the stability problems, simulations with load are discussed in Section V. Every station always wants to be a member of the ring and there are no failures except transmission errors. All simulations run for 3600 simulated seconds. The gap factor was chosen to be 6, the slot time T_{SL} is 400 μs, the station delay is 100 μs and the bitrate is 500 kBit/sec (these settings are typical real-world values). We have used two different error models: in the "independent" error model bit errors occur independently from each other with fixed rate. The second model is the "Gilbert-Elliot" error model (Gilbert model for short) [10], where the channel is always in one of two states: *Good* or *Bad*. Within each state, bit errors are assumed to be independent with a fixed rate. The channel state is modulated according to a two-state continuous time markov chain. For parametrization of the Gilbert model four values suffice: bit error rate (BER) in good state e_{g}, BER in bad state e_{b} ($e_{\text{g}} \ll e_{\text{b}}$), mean duration of good state λ in seconds, and mean duration of bad state μ in seconds. With $p_{\text{g}} = \frac{\lambda}{\lambda+\mu}$ and $p_{\text{b}} = \frac{\mu}{\lambda+\mu}$ being the steady state probabilities for being in state good or bad, respectively, the mean BER m is given by

$$m = p_{\text{g}} \cdot e_{\text{g}} + p_{\text{b}} \cdot e_{\text{b}}. \quad (1)$$

The Gilbert model is very popular for modeling wireless channels due to its simplicity and its ability to capture bursty error behaviour with short term correlation. The (mean) bit error rates chosen in this paper for both error models are in the range $10^{-4} \ldots 10^{-3}$. These values are realistic for wireless transmission in an industrial environment, since corresponding measurements taken in an industrial environment establish high bit error rates (up to 10^{-2}) and nonstationary error behavior [13].

In Figure 1 we show $\bar{M}(3600)$, and in Figure 2 we show

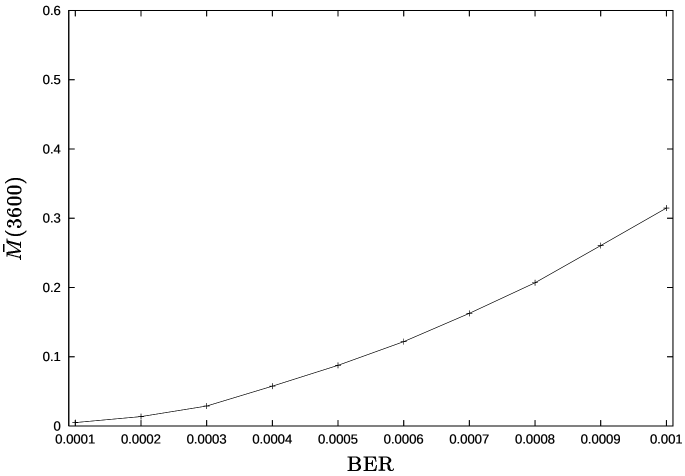

Fig. 1. $\bar{M}(3600)$ vs. BER (independent errors)

$\bar{N}(3600)$, both for varying BER using the independent error model. Furthermore, in Figure 3 we show the distribution functions $C(s)$ for the different bit error rates assuming independent errors. The nearly vertical line on the left side comes from the time resolution used (5 ms) and the fact that all distributions have a share between 5% and 21% of their mass within the first 5 ms. The confidence intervals for $\bar{N}(3600)$ are very tight and thus not shown[2]. In Figure 1, a nearly linear relationship between the bit error rate and the fraction of time during which the ring is not complete can be observed. For the highest bit error rate this fraction is approximately 1/3. Even more frustrating is the result that for the lowest investigated bit error rate of 10^{-4} a full ring is stable for less than 15 seconds in more than 40 % of all cases, even if $\bar{N}(3600)$ and $\bar{M}(3600)$ look good. This is a serious problem for real-time applications over error prone links, since for reincluding of a lost station some time is needed.

In order to show that the protocol is not only sensible to the overall bit error rate but also to the characteristics of the error process (specifically: its "burstiness"), we have performed simulations with the Gilbert error model. Specifically, we have chosen to keep $m = 0.001$, $e_{\text{g}} = 0.0000820$ and $\lambda = 0.061736$ fixed and to vary μ using values of 5, 10, 20, 30, 40, 50 and 60 ms[3], then determining e_{b} from equation 1. We define the burstiness index (BI) to be λ/μ. The question, whether the ring stability metrics are invariant of the scale of λ and μ is not further inves-

[2]The maximum relative error for all simulations in this paper for the $\bar{N}(\cdot)$ value is with 98 percent confidence not larger than one percent of the absolute value. Most relative errors are smaller than 0.1 percent. For actually calculating these values within our simulation we approximate $N(t)$ by a sampled version $N_k = N(k \cdot T)$ with $T = 100\mu s$ fixed and $k \in \mathbb{N}$. Accordingly we calculate $\bar{N}(t)$ with $k_t = \max\{k \in \mathbb{N} : k \cdot T < t\}$ as the sample mean: $\bar{N}(t) = \frac{1}{k_t} \sum_{i=0}^{k_t} N_i$ and the variance $\bar{N}^2(t)$ as the sample variance. Furthermore, in the simulator we apply transient removal techniques for achieving steady-state results.

[3]The values for λ and e_g are calculated directly from [10], while the values of μ chosen in this paper have the same order of magnitude as those from [10].

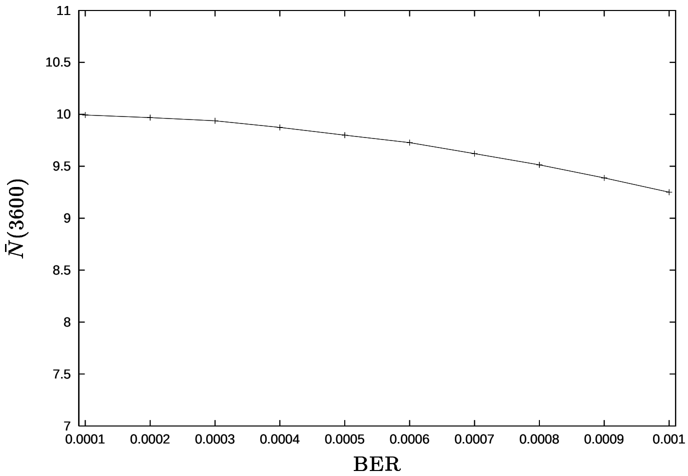

Fig. 2. $\bar{N}(3600)$ vs. BER (independent errors)

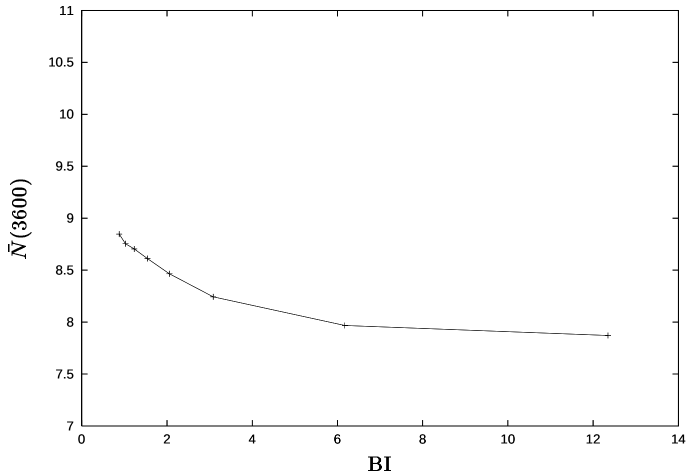

Fig. 4. $\bar{N}(3600)$ vs. BI for m = 0.001 (Gilbert errors)

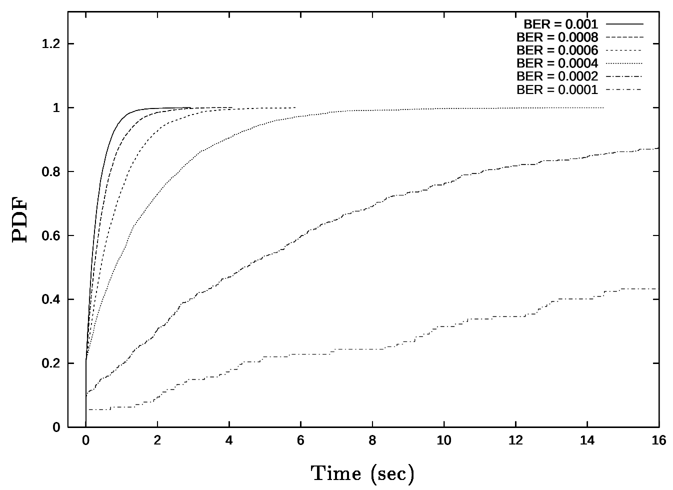

Fig. 3. Distribution of C(s) for different BER's

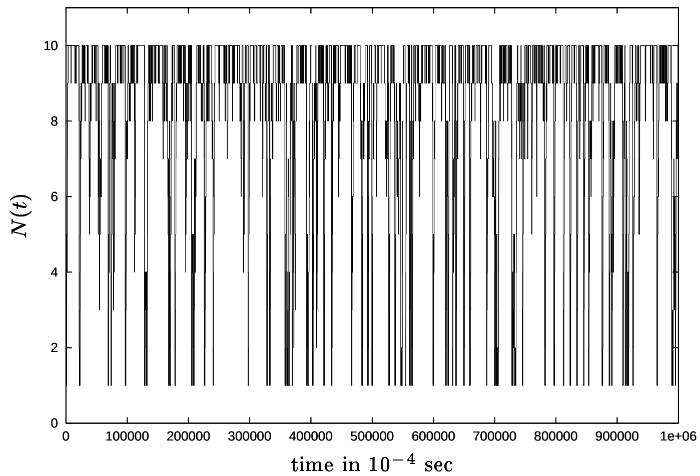

Fig. 5. $N(t)$ vs. time (Gilbert errors, normal protocol)

tigated. In Figure 4 we show $\bar{N}(3600)$ vs BI. Apparently for more bursty errors (larger BI) this metric decreases. This can be explained as follows: since for constant m the value e_b increases when BI increases, it is more likely that a station experiences a hearback error.

As a visual impression that frequently the number of ring members reduces from five or more to one within a very short time, the evolution of $N(t)$ for the first 100 seconds (Gilbert errors, $m = 0.001$, $\mu = 20$ ms) is displayed in Figure 5. A careful analysis of the corresponding simulator traces shows that often multiple stations are lost simultaneously, and that these breakdowns are indeed caused by the ring jacking scenario. Furthermore, it shows up that the frequent transitions from ten members to nine members are caused by hearback removals. The error skipping scenario is much more rare: for the worst parameter setting (Gilbert errors, $m = 0.001$, $\mu = 5$ ms, $e_b \approx 0.012$) a token frame with undetectable errors is observed once every minute in the mean. Therefore it is not considered furthermore in this paper.

V. Improvements

In this Section we propose a new method for setting timeout timers and an additional protocol feature. The new timer setting tries to prevent the breakdowns of the ring by letting expire the timeout timer for current ring members first, while the additional protocol feature aims at reincluding lost stations as fast as possible. Since both of them require no modification of frame formats or protocol operation, they are interoperable with the unchanged protocol. Thus, in principle, stations with the modified and the unchanged protocol stack can be operated in the same PROFIBUS LAN. However, we need the ability to dynamically influence the timeout timer setting, which may require an upgrade of today's ASIC-based protocol implementations. Both methods are limited to combatting the ring jacking and hearback removal scenarios, avoiding the error skipping scenario requires a better protection of the token frame and thus a change in frame formats.

The effect of the proposed methods is investigated with simulations, using the same scenarios and stability metrics as in Section IV, and with additional simulations taking the effects of system load and different numbers of active stations into account.

A. Timeout Calculation

From our simulations and from analysis we have observed that the ring jacking scenario (described in Section II-C), where the station with the lowest address can destroy the whole ring, occurs frequently. The calculation of the timeout value is for station n as follows [1, sec. 4.1.7]:

$$T_{TO}(n) = (6 + 2 \cdot n) \cdot T_{SL}$$

where T_{SL} is the slot time. The basic problem of this scenario is that the timeout timer may expire for a station which is in the listen token state and has no valid LAS. If the timer of a station in the ring (not in the listen token state) expires, the ring keeps alive. Thus we propose to make the timeout calculation state-dependent:

$$T_{TO}(n) = \begin{cases} (6 + 2 \cdot n) \cdot T_{SL} & : \text{state} \neq \text{listen token} \\ (254 + 6 + 2 \cdot n) \cdot T_{SL} & : \text{state} = \text{listen token} \end{cases}$$

in order to make sure that the timeout timer expires first for stations in the ring and as a result to avoid ring jacking. We show the effects of this improvement in the next Section.

B. Fast Reinclusion of Lost Stations

When a station is lost from the ring, it does take some time before it is reincluded. First, the station is required to observe the same sequence of token frames twice, second, it will not be reincluded before it is pinged by its predecessor using the Request-FDL-Status frame. We propose to add the following extra feature to the protocol: after station a has lost its successor b (i.e. there is no reaction of b to three consecutive token frames), a waits for two token cycles and then pings b with the Request-FDL-Status frame as soon as there is token holding time available. This is the earliest moment where b can be reincluded, due to b's need for reading two identical token cycles. This procedure should be carried out independently of the normal ring-inclusion algorithm. Thus it can happen, that a includes another station c during the two token cycles it waits for reincluding b. In this case b should only be reincluded if its address lies in the range between a and c, otherwise c will remove itself from the ring, being skipped by the first token frame a sends to b. However, when the ring jacking scenario occurs more frequently, this protocol extension should be used in conjunction with the new timeout calculation method, since otherwise fast reinclusion will not happen.

C. Performance Evaluation

We compare three different versions of the protocol: the normal protocol without any improvements, the protocol with the new timeout calculation method and the protocol with both improvements. The simulation setup is the same as described in Section IV. The results for $\bar{M}(3600)$ are shown in Figure 6, the results for $\bar{N}(3600)$ are shown in Figure 7, both for independent errors and varying BER. These figures show that the new timeout computation significantly improves stability, the protocol with both improvements performs best. In Figure 8 the sample coefficient of variation for N is shown. It can be seen that the improvements reduce the variability of N. In Figure 9 we compare the three protocol versions for the case of Gilbert errors and varying burstiness index (BI) for fixed mean BER $m = 0.001$. The stability gain of the improvements as compared to the normal protocol is larger for more bursty errors than for the "smooth" independent errors. As a visual impression we show in Figure 14 the evolution of $N(t)$ for the same system as for Figure 5 (ten masters, no load, Gilbert errors with $m = 0.001$ and $\mu = 20$ ms), however, with both protocol improvements enabled. It can be seen that most of the breakdowns visible in Figure 5 are removed.

We additionally mention here that the ring jacking scenario also influences the local stability metrics mentioned in Section III. One example is the fraction of time that station i is not in the ring. For the station with the lowest address this fraction is small and nearly independent of the gap factor or the TTRT for a fixed bit error rate, while for all other stations this metric depends almost linearly on the gap factor, and furthermore increases with increasing station address (see ref. [11] for examples).

In order to show that ring stability problems occur also when there is load in the system (and thus a smaller number of vulnerable token frames per fixed unit of time), we have investigated two more scenarios. In the first scenario there are four active stations, two passive stations[4], and four traffic sources, each attached to a different active station. The traffic sources generate requests, the attached station puts them in a queue of infinite size. Two traffic sources generate requests with a fixed interarrival time of ten ms. The corresponding requests lead to frames of 25 bytes (carrying 16 bytes of user data), which are acknowledged by the passive station with frames of 25 bytes (including 16 bytes user data). The other sources generate sporadic requests with exponentially distributed interarrival times (ten ms mean value), destined for the second passive station and with data sizes uniformly distributed between 8 and 30 bytes (leading to frame sizes between 17 and 39 bytes), however, the acknowledgement carries no data. Thus, we have a mixture of synchronous and asynchronous traffic.

In the second scenario we have ten active stations and ten traffic sources. The first five sources are periodic (with 25 ms period), the other sources are sporadic (with 25 ms mean value). Thus in both scenarios a minimum bandwidth of $\approx 35\%$ of the medium bandwidth is devoted to exchange of data frames including the acknowledgements, but not including retransmissions. The need for retransmissions at error rates of $\approx 10^{-3}$ saturates the system, higher loads lead to growing request queues. This is true espe-

[4] Active stations are those who can participate in the token passing process. They are simply referred to as "stations" within this paper. Passive stations only transmit data when they are polled. In the simulations they are used as a mere data sink.

cially for independent errors, for Gilbert errors the queues can be emptied during good channel periods. The simulations run for 10000 simulated seconds, the other parameters (gap factor, TTRT, bit rate, slot time) are kept fixed. The $\bar{N}(10000)$ results for the scenario with ten stations are shown in Figure 10 (independent errors) and Figure 11 (Gilbert errors). It can be seen that for all three protocol versions and both error models this value is better than in the corresponding simulations without any load. However, for high bit error rates the stability problems and their dependence on the type of channel errors are still visible, but the proposed improvements again yield a significant gain.

The $\bar{M}(10000)$ values for both station numbers are shown in Figures 12 (independent errors) and 13 (Gilbert errors) for the normal protocol and the protocol with both improvements. Again, in the presence of load this metric is better (lower) than for the corresponding simulations without load (not shown here for Gilbert errors), and the improved protocol version yields the best results. Interestingly, in both figures the numbers are smaller for fewer stations. While for four stations and ten stations the times for breaking a full ring are comparable (four stations: mean value $\bar{C} \approx 1.12$ sec, stddev ≈ 1.38; ten stations: mean $\bar{C} \approx 1.17$ sec, stddev ≈ 1.27) with ten stations it takes much longer to complete the ring. Likely the difference stems from the time needed to complete the ring after multiple stations have been lost at once, as in the ring jacking scenario. If only a single station gets lost, it is reasonable to expect that reinclusion is slightly faster in the ten station case, since the gap lists typically are shorter than with fewer stations. Furthermore, for a newly reincluded station there might be some delay between its reinclusion and the time it starts to poll its gap list, since in the simulation the gap timer is independent from the stations state of ring membership. As a result, if more stations need to be reincluded, a higher delay for ring completion can be expected.

All these findings together confirm our belief that ring instability is an issue for higher bit error rates, and furthermore that two important sources for instability are the ring jacking and hearback removal scenario, while the error skipping scenario seems to play a much smaller role. The ring jacking and hearback removal scenarios can be combatted with the two improvements proposed in this paper. Since for lower bit error rates station losses occur rarely and the improvements are not invoked, they impose no additional cost in terms of bandwidth or delay.

VI. Conclusions

In this paper we have identified ring instability as an issue to be considered for Quality of Service and timing behavior of the PROFIBUS. Especially when carrying out schedulability analysis for PROFIBUS traffic streams, it does not suffice to take only retransmissions into account and to assume a stable logical ring. We have shown that station losses and longer station outage times occur frequently, when an error prone medium is used, and furthermore that station losses are sensitive to the type of errors (bursty vs. non-bursty). We have identified three different scenarios, which may cause station loss. We have found that especially the hearback removal scenario and the ring jacking scenario (both relying on the hearback protocol mechanism) can lead to an unacceptable degree of instability, while the error skipping scenario occurs only rarely. We have proposed two improvements in the protocol and parameter settings, which, when operated jointly, significantly increase ring stability. Furthermore, these improvements require no changes in frame formats or the basic protocol. It is possible to implement them only in a subset of stations without affecting the behavior of the remaining stations or the ring.

We are convinced that the behavior of fieldbus protocols over error prone and time varying links is an extremely interesting topic, especially with regard to future transmission technologies, e.g. wireless LANs. The behavior of protocols like P-NET [7] or WorldFIP [9] is an important issue of future research. For the PROFIBUS it is worthwhile to look for further improvements (e.g. find other means for transceiver self checks and collision detection, eliminating

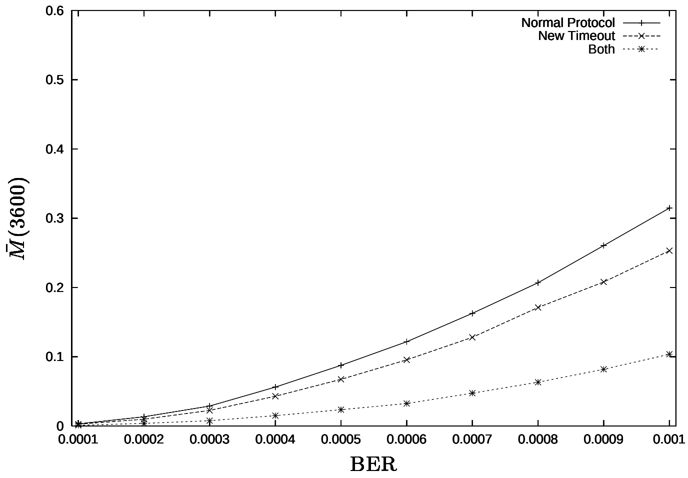

Fig. 6. $\bar{M}(3600)$ vs. BER (independent errors)

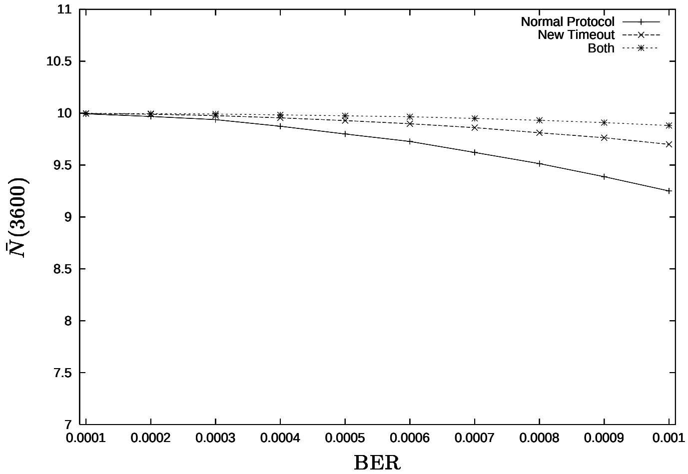

Fig. 7. $\bar{N}(3600)$ vs. BER (independent Errors)

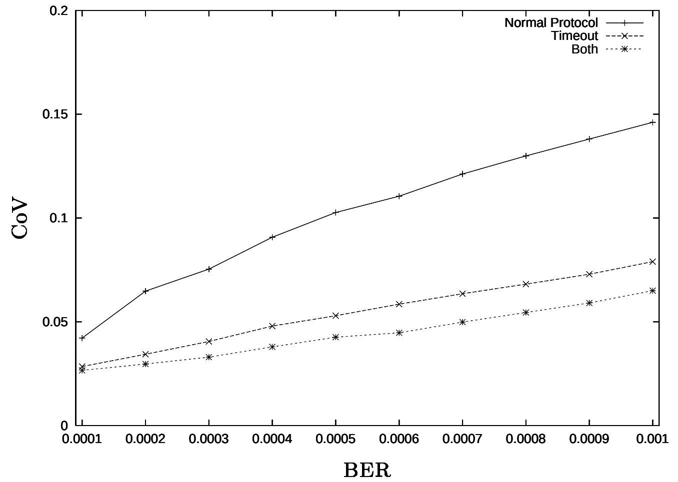

Fig. 8. Sample Coefficients of Variation for N vs. BER (independent errors)

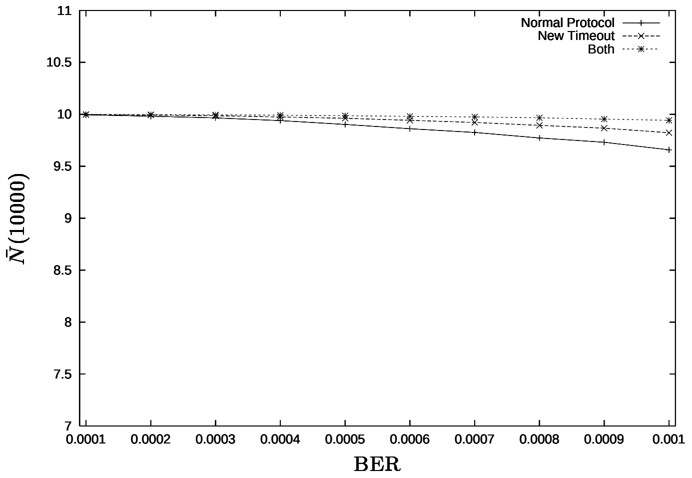

Fig. 10. $\bar{N}(10000)$ vs. BER (independent errors) with 10 masters and $\approx$ 36% load

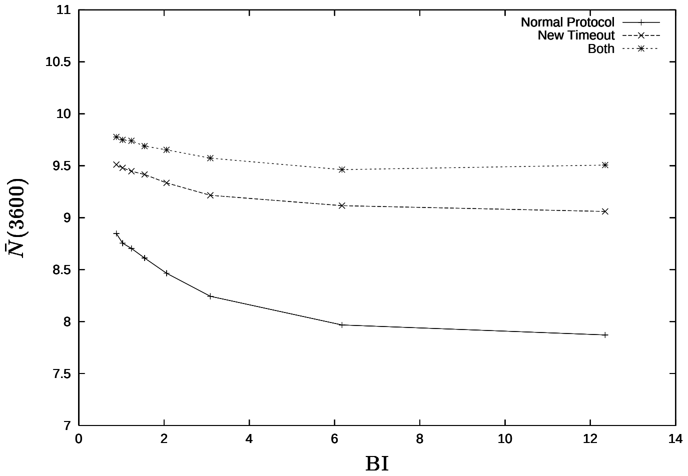

Fig. 9. $\bar{N}(3600)$ vs. BI for m = 0.001 (Gilbert errors)

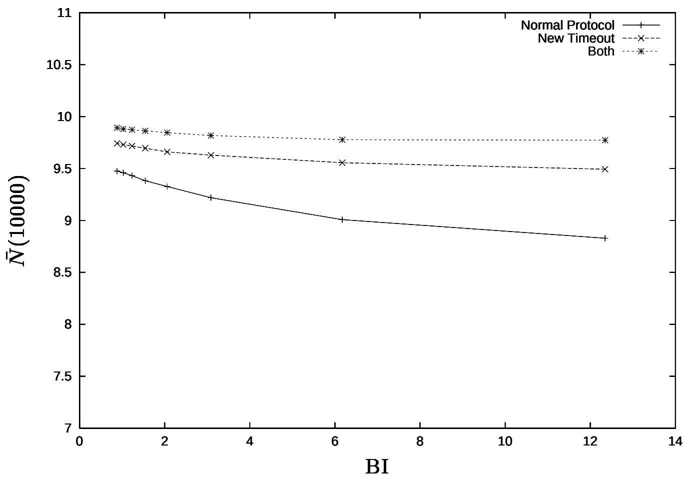

Fig. 11. $\bar{N}(10000)$ vs. BI for m = 0.001 (Gilbert errors) with 10 masters and $\approx$ 36% load

the need for hearback) and to find out how the protocol behaves, if the hearback feature is not available.

Acknowledgments

The authors wish to thank Morten Schläger, Holger Karl, and the anonymous reviewers for their helpful comments.

References

[1] German Institute of Standardization (DIN). *PROFIBUS Standard Part 1 and 2*, 1991.

[2] Hong ju Moon, Hong Seong Park, Sang Chul Ahn, and Wook Hyun Kwon. Performance Degradation of the IEEE 802.4 Token Bus Network in a Noisy Environment. *Computer Communications*, 21:547–557, 1998.

[3] Mesquite Software, Inc., T. Braker Lane, Austin, Texas. *CSIM18 Simulation Engine – Users Guide*, 1997.

[4] PROFIBUS Nutzerorganisation e.V., PROFIBUS Nutzerorganisation e.V., Haid-und-Neu-Str. 7, Karlsruhe, Germany. *Implementation Guide to DIN 19245 Part 1*, August 1994.

[5] Eduardo Tovar. *Supporting Real-Time Communications with Standard Factory-Floor Networks*. PhD dissertation, Dept. of Electrical Engineering, Univ. of Porto, Portugal, 1999.

[6] Eduardo Tovar and Francisco Vasques. Real-Time Fieldbus Communications Using Profibus Networks. *IEEE Transactions on Industrial Electronics*, 46(6):1241–1251, December 1999.

[7] Union Technique de l'Electricité. *General Purpose Field Communication System, EN 50170, Volume 1: P-NET*, 1996.

[8] Union Technique de l'Electricité. *General Purpose Field Communication System, EN 50170, Volume 2: PROFIBUS*, 1996.

[9] Union Technique de l'Electricité. *General Purpose Field Communication System, EN 50170, Volume 3: WorldFIP*, 1996.

[10] H.S. Wang and N. Moayeri. Finite State Markov Channel - A Useful Model for Radio Communication Channels. *IEEE Transactions on Vehicular Technology*, 44(1):163–171, February 1995.

[11] Andreas Willig. Analysis of the PROFIBUS Token Passing Protocol over Error Prone Links. In *Proc. 25th Annual Conference of the IEEE Industrial Electronics Society (IECON'99)*, pages 1246 – 1252. IEEE, November 1999.

[12] Andreas Willig. Markov Modeling of PROFIBUS Ring Membership over Error Prone Links. TKN Technical Report Series TKN-99-004, Telecommunication Networks Group, Technical University Berlin, May 1999. http://www-tkn.ee.tu-berlin.de/publications/tknrreports.html.

[13] Andreas Willig, Martin Kubisch, and Adam Wolisz. Results of Bit Error Rate Measurements with an IEEE 802.11 compliant PHY. TKN Technical Report Series TKN-00-008, Telecommunication Networks Group, Technical University Berlin, November 2000. http://www-tkn.ee.tu-berlin.de/publications/tknrreports.html.

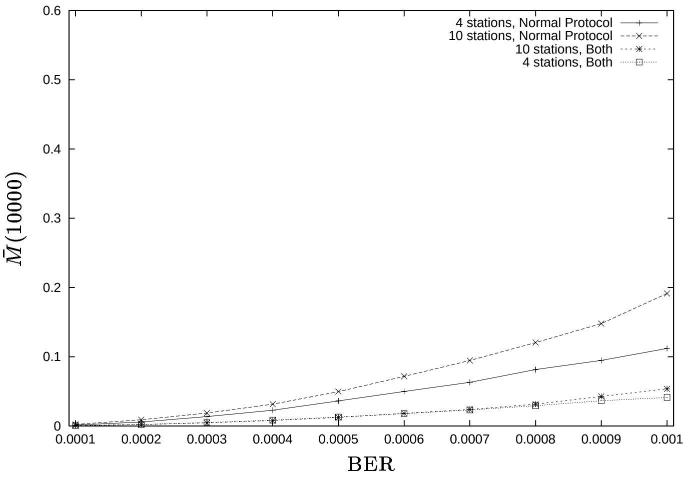

Fig. 12. $\bar{M}(10000)$ vs. BER (independent errors) with 4 and 10 masters and $\approx 36\%$ load

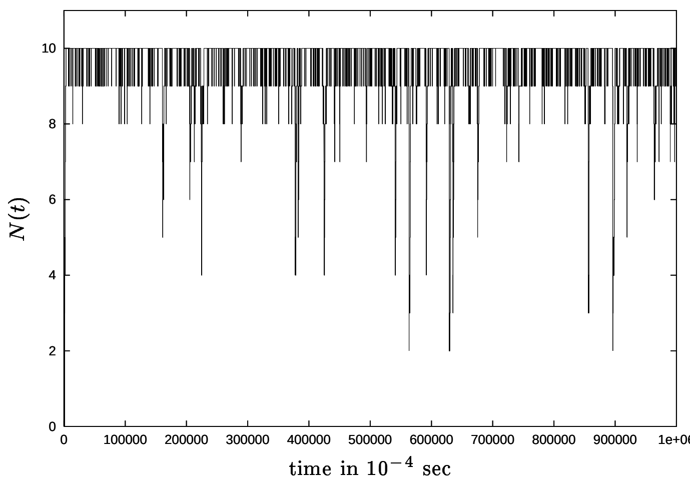

Fig. 14. $N(t)$ vs. time (Gilbert errors, both protocol improvements)

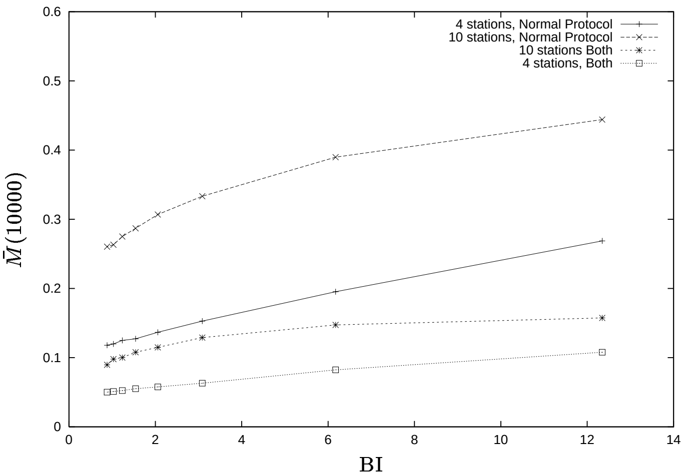

Fig. 13. $\bar{M}(10000)$ vs. BI $m = 0.001$ (Gilbert errors) with 4 and 10 masters and $\approx 36\%$ load

Andreas Willig (M '97) is currently a research assistant at the Telecommunication Networks Group at Technical University Berlin, working toward a Dr.-Ing. degree on wireless PROFIBUS systems. He holds a diploma degree in computer science from University of Bremen, Germany. His research interests include wireless and cellular networks, fieldbus systems, real-time systems and active networks.

Adam Wolisz obtained his degrees: Dipl.- Ing in Control Engineering, Dr.-Ing. and Habilitation (both in Computer Engineering) respectively in 1972, 1976 and 1983, all at the Silesian Technical University in Gliwice, Poland. In the period 1972-1989 he was with the Institute of Complex Control Systems of the Polish Academy of Sciences working initially on real-time operating systems and computerised industrial control systems, and since 1980 on computer networks and distributed systems. From 1990-1993 he was with the Research Institute for Open Communication Systems of the German National Research Center for Computer Science (GMD-Fokus) in Berlin heading activities on quantitative aspects of high-speed networks and multimedia systems.

Since 1993 he is Professor of Electrical Engineering and Computer Science at the Technical University Berlin, where he is leading the Telecommunication Networks Group (TKN) Parallelly he is also a member of the senior board of GMD Fokus.

His research interests are in architectures and protocols of communication networks as well as protocol engineering with impact on performance and Quality of Service aspects. Recently he is working mainly on wireless LANs and their integration into both Internet and ATM environment.

He is author of 2 books and over 60 papers in technical journals and Conference Proceedings. He is Member of IEEE, IEEE Communication Society and the GI/ITG Technical Commitee on Communication and Distributed Systems.