Joachim Charzinski

Performance of the Error Detection Mechanisms in CAN

Abstract: **CAN systems are designed to be used in automotive and automation environments where it is likely to have a high degree of electromagnetic interference disturbing the data transported between the transmitter and receiver. CAN uses several error detection mechanisms to prevent receivers from accepting disturbed data. Assuming a two-state symmetric binary channel model for the physical transmission medium, this paper analyzes the probability for errors to be undetectable at receivers (residual error probability). The contributions from different error mechanisms to the residual error probability are identified and quantified.**

1 Introduction

The Controller Area Network (CAN) was developed for the automotive environment to allow for high speed data communication in cars with little wiring effort. The protocol and the hardware requirements have been standardized by the ISO [1]. Having been developed for an environment with a high degree of electromagnetic interference, CAN is also an ideal system to be used in the automation environment.

In both applications it is necessary to have high data security, i.e. no communications partner should accept any data from another partner when those data have been changed by transmission errors. In the following the communications partners will be regarded as *stations* exchanging information over a physical *transmission channel* using CAN controllers for medium access and conversion between user data and physical bits, see Fig. 1.

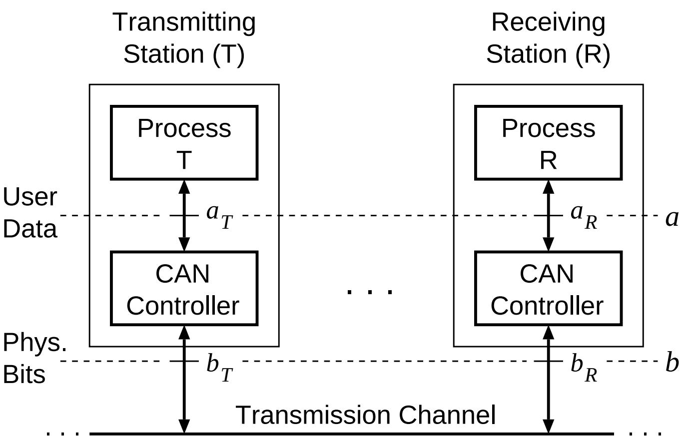


Figure 1: Data Communication over CAN

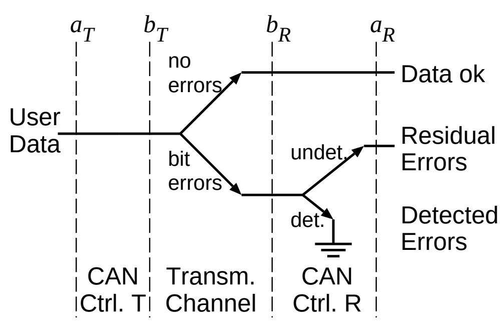


Figure 2: Residual Errors

Fig. 1 describes the communication between a transmitting station (T) and a receiving station (R). Due to the message filtering concept of CAN, there can be more than one receiving station for a message. The

[1]The author is a member of the scientific staff at University of Stuttgart, Institute of Communications Switching and Data Technics, Seidenstr. 36, D-70174 Stuttgart, Germany; e-mail: charzinski@ind.uni-stuttgart.d400.de

transmitting process T gives its user data to the CAN controller at the interface a_T. The CAN controller performs the medium access control and converts the user data into a stream of physical bits which are transmitted over the transmission channel (interface b_T) in the form of a *CAN frame*, also called a *message*. The receiving stations get a stream of bits from the transmission channel through interface b_R which is converted back into user data for the receiving process R and its interface a_R.

Bit errors on the transmission channel can cause the data seen at interfaces b_T and b_R to be different (see Fig. 2). The error detection mechanisms of CAN should delete the received data in this case and not give them to interface a_R. If there are errors in a received CAN frame which are not detected by the CAN controller, a *residual error* has occurred. The *residual error probability* p_{res} is the probability for a message to be accepted by a receiving controller although there are bit errors in it. Note that all CAN controllers check all messages seen on the bus for errors. p_{res} is therefore used to quantify the probability of an error being undetectable without regard to whether the affected station has a receiving process for this message or not. This probability also only shows how reliable the information *received* at an interface a_R is and it does not give any notion of the reliability of transfer, i.e. the probability for a message not to be affected by errors between b_T and b_R.

This paper considers only the performance of the error detection mechanisms in the *CAN protocol*, assuming a channel model for the errors occurring during the transmission between b_T and b_R presented in section 2. To be able to model a variety of physical media, the most important parameters of the channel are left open in the results. Also the performance of additional error detection mechanisms in the receiving process above interface a_R cannot be considered here. Some systems will perform format and consistency checks of the data received and may even incorporate additional error detection mechanisms in the receiving process. On the other hand, especially in automotive applications there are systems which have to be constructed with minimum memory and CPU power requirements, so that there are no further checks done once the user data appears at a_R. Especially in these systems, the residual error probability analyzed in this paper is of great significance for the overall system safety.

The error detection mechanisms of CAN and the different ways in which errors can be undetectable in a CAN system are discussed in section 3. The analytical model for the mechanisms is presented in section 4, leading to the results shown in section 5.

2 Channel Model

In order to evaluate the residual error probability at interface a_R, a model for the bit error properties of the transmission channel between b_T and b_R is needed. The CAN protocol uses an asymmetric channel (bus) for physical bit transmission with dominant and recessive states (**d** and **r**). If more than one station sends a signal, the whole bus takes the dominant state if at least one station sends a dominant signal. The bus takes the recessive state only when all stations send recessive signals.

If parts of the bus wires are exposed to an interfering electromagnetic field, bit errors can occur at some or at all stations. So far there have been no tractable models published for dominant/recessive channels. For this reason the channel is approximated as a binary symmetric channel (Fig. 3) where the probability for a bit to change from **d** to **r** during transmission is the same as for a change from **r** to **d**, denoted by the bit error probability p_E.

Further modelling of the channel is done in a *temporal* as well as in a *spatial* dimension. The temporal sequence of bit errors is described by a model with the two states *good* and *bad*. In both states, the channel can be described by the binary symmetric model from Fig. 3 with bit error probabilities $p_{E_{good}}$ and $p_{E_{bad}}$, respectively, see Fig. 4. It is assumed that the channel is in the *bad* state with probability q_{bad} and in the *good* state with probability $1 - q_{bad}$.

This temporal channel model allows for a decomposition of the residual error probability into a contribution from the *good* and the *bad* phases (1).

with $p_r(p_E)$ denoting the residual error probability for a symmetric channel which always exhibits the bit error probability p_E.

$$p_{res} = q_{bad}\, p_r(p_{E_{bad}}) + (1 - q_{bad}) p_r(p_{E_{good}}) \quad (1)$$

The second dimension of the model is the distribution of bit error occurrences at the different stations, including the transmitting station. A spatially varying distribution of bit errors is the consequence of the

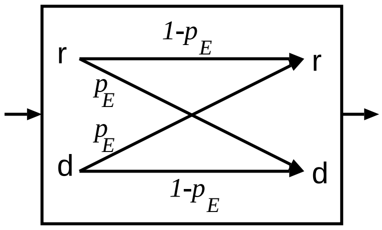


Figure 3: Transfer Model for the Binary Symmetric Channel

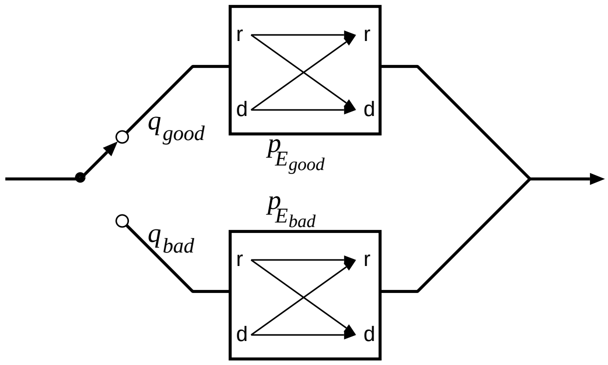


Figure 4: Two-State Binary Channel Model

assumption that different parts of the network can be exposed to different sources of interference and that interfering noise can modify the transmitted signal to be around the decision limit between the **d** and **r** states, so that the resulting output decision is random. Therefore the spatial distribution of errors in the network is modelled by p_{eff}, the probability for a bit error occurring somewhere in the network to appear at a certain station. For each time slot (bit time) equation (2) holds:

$$p_{\mathit{eff}} = \mathrm{P}\{\text{error affects considered station} \mid \text{error somewhere in the network}\} \tag{2}$$

3 Error Mechanisms

3.1 Frame Format

Fig. 5 shows the communication on a CAN bus observed on different logical and temporal scales.

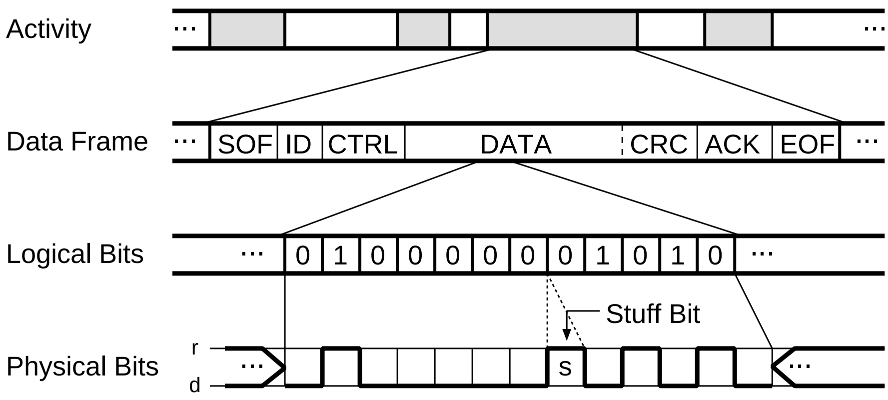
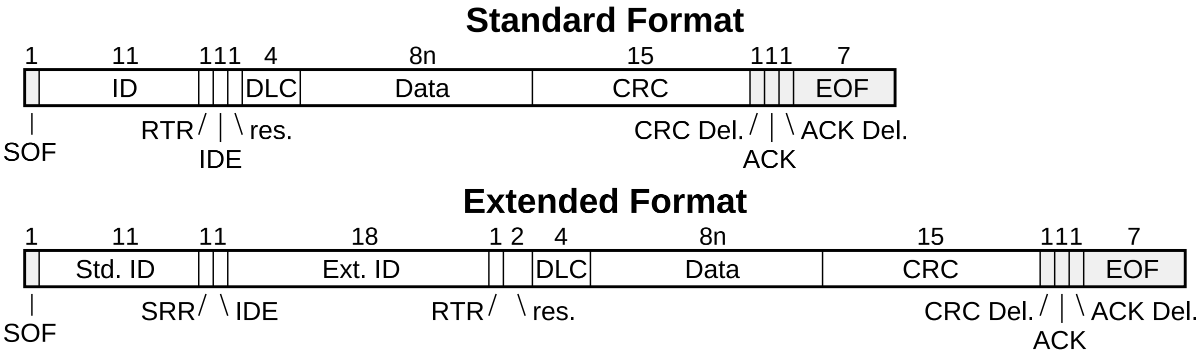


Figure 5: Observation and Time Scales

On a coarse time scale, activity and idle periods can be distinguished on the bus. During an activity period, frames are transmitted on the bus and during the idle period the bus is in its idle state: all stations generate **r** bits (interframe space). Each frame consists of different fields which are labelled according to their function, see also Fig. 6. There are two frame formats defined, one with a short message identifier (ID) field and one with an extension for the ID field.

The ID, Data and CRC fields contain data with logical interpretation. These *logical bits* are converted into *physical bits* according to a bit coding scheme. In CAN, any sequence of more than five consecutive physical bits of equal value within a frame is interpreted as an error signalling frame. In order to prevent these *error frames* from occurring within a data frame and to maintain receiver synchronization even through longer periods of consecutive equal data bits, a bit stuffing rule is applied as bit coding scheme.



SOF	Start of Frame bit (fixed)	DLC	Data Length Code
ID	Message Identifier	Data	n data bytes
RTR	Remote Transmit Request bit	CRC	Cyclic Redundancy Check
SRR	Substitute Remote Request bit	CRC Del	CRC Field Delimiter bit (fixed)
IDE	ID Extension Switch	ACK	Acknowledgement bit for receivers
Ext. ID	ID Extension Field	ACK Del	ACK Field Delimiter bit (fixed)
res	reserved bit(s)	EOF	End of Frame (fixed)

Figure 6: Frame Formats

Whenever there are five consecutive data bits of the same value somewhere between SOF and the end of CRC, the transmitter *inserts* a physical bit of opposite polarity into the bit stream. This bit is taken out again by the receiver when it converts the physical bits back to logical bits.

3.2 Error Detection Mechanisms

The CAN protocol uses the following error detection mechanisms:

1. *Monitoring*: The transmitter of a bit compares the signal sent with the signal seen on the bus line (transmission channel). Except for the arbitration phase during the transmission of the identifier of a message and in the ACK slot, a transmitter starts sending an error frame if the signal on the bus line is different from the signal sent. In this way bit errors affecting all stations on the bus cannot lead to non-detectable errors because they will be detected by the transmitter of the frame.

2. *Cyclic Redundancy Check* (CRC): The 15 CRC bits are computed from every bit from SOF to the last data bit. The BCH code used for generating the CRC leads to a hamming distance of six, including a parity check, in the unstuffed bit sequence.

3. *Message Frame Check*: The SOF, RTR, IDE, DLC, delimiter and EOF fields must be consistent with the CAN specification. If a fixed format field in a received frame (except for the last EOF bit) does not conform to the standard, the receiver sends an error frame and does not accept the received frame.

4. *Bit Stuffing*: Any violation of the stuff rule between SOF and CRC is regarded as an error.

5. *Acknowledgement*: The transmitter of a data or remote frame treats a missing acknowledgement as an error and destroys the EOF field by sending an error frame.

6. *Error Signalling*: Each station that detects an error starts sending an error frame, so that other stations are notified of this condition by seeing a violation of the stuff rule or the fixed format delimiter or EOF fields. Note that due to the Bit Stuffing mechanism all stations answer a received error frame with error frames of their own.

Each station that detects an error starts sending an error frame, so that other stations are notified of this condition by seeing a violation of the stuff rule or the fixed format delimiter or EOF fields. Therefore a non-detectable error must be non-detectable at every station that is disturbed and the transmitter of the frame must not be disturbed.

3.3 Error Classes

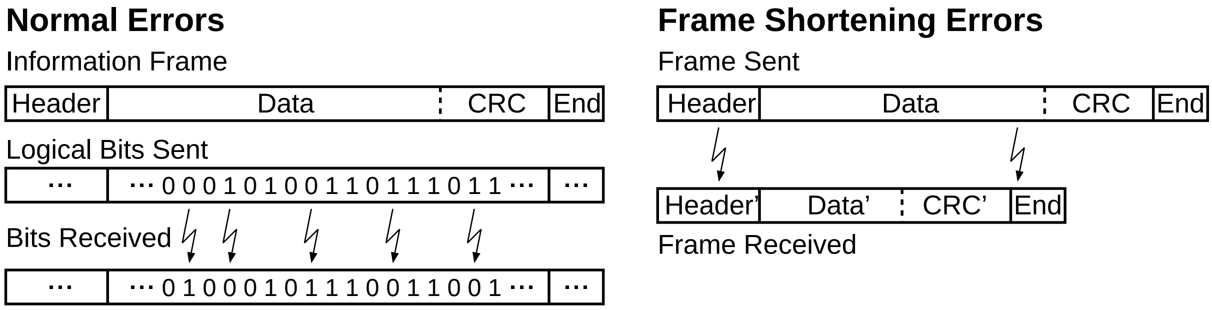


Figure 7: Types of Errors

Two main classes of residual errors can be distinguished, see Fig. 7:

a) Errors that do not affect the frame length: In this class, all fields of the transmitted frame are interpreted with their correct function by the receiver. The fields affected by bit errors must be restricted to the Data, DLC and ID fields.

b) If the SOF, RTR, IDE or DLC fields are changed by bit errors, the receiver expects the frame to have a length different from the original length and thus interprets frame fields differently from their original meaning, see Fig. 7. Depending on the changes produced by the errors, the receiver can expect a frame with a greater or smaller length than the original frame. However, the residual error probability is much smaller for increasing than for decreasing the frame length by errors: The ACK bit sent by the affected receiver will disturb the fixed format ACK Del-EOF sequence or be interpreted as SOF of a new frame if it is transmitted after the original ACK position. The error frame produced upon this event by other stations (at least the sender of the disturbed message) will destroy the EOF' field seen by the disturbed receiver. On the other hand, there is a high probability for a disturbed receiver accepting a shortened frame because there is only one bit error needed e.g. to change DLC from 7 to 3 and one more bit error to create the ACK Del-EOF sequence from a random stuffed bit sequence.

3.4 Error Transformation

Apart from bit errors affecting only normal data bits and not violating the stuffing rule, there is also a chance of a pair of bit errors deleting a stuff bit and creating another, see Fig. 8.

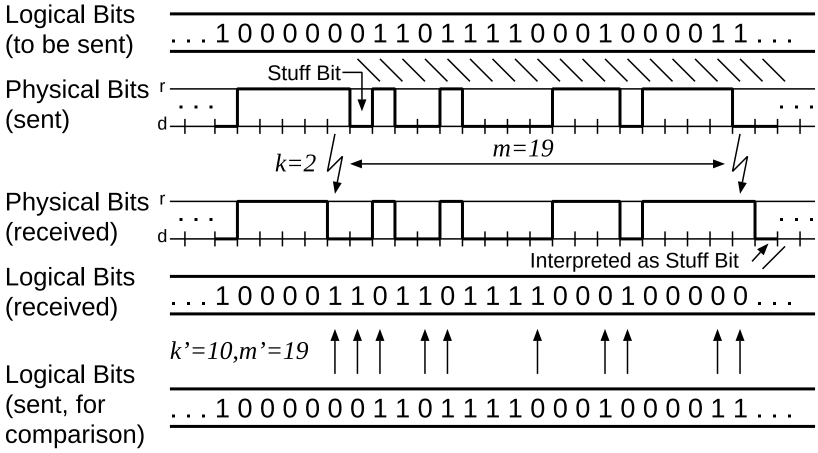


Figure 8: Stuff Bit Errors

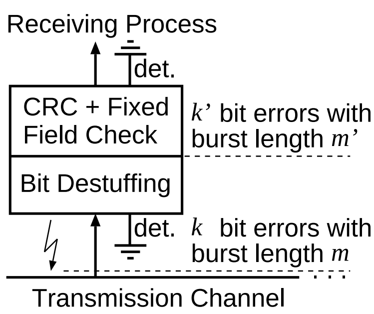


Figure 9: Error Transformation

This property of the bit stuffing and destuffing rules can be tracted by analyzing a replacement channel consisting of the transmission channel and the destuffing unit in the receiving station. This replacement channel shows an error burst of length m' with k' bit errors where the transmission channel introduced k bit errors with a burst length m. The destuffing unit is thus described by the transformation probabilities $p_T(k, m, k', m')$. Depending on the data bits between the stuff bit deleting and creating errors, the number k' of destuffed bit errors can be much greater than the number k of bit errors on the transmission channel.

The effect of error multiplication due to bit stuffing has first been stated in [4] for the HDLC protocol and in [3] for CAN. Due to the bit stuffing effects, the error detection capabilities of the CRC are reduced from safely detecting $h - 1 = 5$ bit errors to 2 errors.

4 Analysis

Only the main line of the analysis can be outlined here. The formulae derived in [2] for the analysis of the case of bit errors in the Data and CRC fields of a CAN frame are explained. The results are given for all error cases and it can be seen that over a wide range of bit error probabilities this case is dominant.

4.1 Summary of Assumptions and Parameters

Only the error detection performance of the CAN protocol between physical bit transmission (interfaces b in Fig. 1) and user data transmission (interfaces a) is analyzed. The bit transmission between b_T and b_R is modelled by a two-state channel with probability q_{bad} for transmitting a frame during a *bad* phase of the channel. The channel is assumed to stay in the same state during the transmission of one frame.

For the results shown in section 5, the effectivity of bit errors is assumed to be randomly distributed over the stations with $p_{eff} = 1/N$ and the distribution is different for each bit error.

The results are displayed as a function of the two main channel parameters, q_{bad} and p_E.

The user data are assumed to be a random bit sequence (for each user bit: $P\{0\} = P\{1\}$).

All receiving stations are assumed to be in the *error active* state, i.e. they can generate dominant error frames.

4.2 Channel Model, Distributed Error Detection and Error Burst Transformation

In each of its states, the transmission channel is described by the binomial distribution of the number of bit errors (3)

$$p_c(k) = \binom{L}{k} p_E^k (1 - p_E)^{L-k} \tag{3}$$

to have k bit errors in L bits at a bit error probability p_E. The frame length can be computed to be

$$L_s = 8\text{DLC} + 44 \tag{4}$$

for standard data frames and

$$L_e = 8\text{DLC} + 64 \tag{5}$$

for extended data frames, using the number of user data bytes (DLC) and neglecting the number of stuff bits in a frame.

The distributed error detection capabilities due to error signalling demand that every station affected by errors must be affected by undetectable errors in order to get a globally undetectable residual error. Under the assumption that undtetctable error bursts occur seldomly, the probability for several different undetectable error bursts to affect different stations is even lower. Therefore it is assumed as an approximation for small p_E that there is only one undetectable error burst in a system at one time. This leads to the modified demand of all stations affected by an error burst being disturbed by the *same* error burst. Several models considering different degrees of correlation between the bit errors at different stations have been investigated and the result was that all models could be very well described by the term

$$p_{dist}(k) = \alpha \cdot e^{-\beta(k-1)} \tag{6}$$

giving the probability for all stations that are affected by an error burst to be affected by the same error burst with k bit errors and for the transmitter to be unaffected [2]. Under the assumption of an equal spatial distribution of bit errors, the parameters α and β can be shown to have the values

$$\alpha \approx \frac{N-1}{N} \qquad \text{and} \qquad \beta \approx \begin{cases} 1.7 & \text{for} \quad N = 5 \\ 2.4 & \text{for} \quad N = 10 \end{cases} \tag{7}$$

where N is the number of stations in the network (including the transmitter). Depending on the degree of correlation of bit errors and the error probabilities at different stations, α and β can be smaller or greater than the values given in (7).

A combined formula for the error burst transformation probability matrix $p_T(k, m, k', m')$ and the failure to detect the transformed error coming from k phyisical bit errors was derived as $p_{TE}(k)$ in [2] for $k \geq 2$. This is a good approximation for the more exact analysis derived in [2], which needs five sums.

$$p_{TE}(k) \approx 2^{-15} \cdot \sum_{k_p=1}^{[k/2]} (p_{SD} p_{SC})^{k_p} (p_N + p_{DE})^{k-2k_p} + \begin{cases} 2^{-14} p_N^k & \text{for } k \geq 6 \text{ and } k \text{ even} \\ 0 & \text{for } 2 \leq k \leq 5 \text{ or } (k > 6 \text{ and } k \text{ odd}) \end{cases} \tag{8}$$

Equation (8) considers k bit errors, $2k_p$ of which create and delete stuff bits. Neglecting short range correlation effects, the bit error effects after destuffing can be described by the set of probabilities [2] given in (9) through (13):

Normal Error (transparent bit error)	p_N	$=$	$39/61$	(9)
Stuff Error ($\rightarrow$ error frame)	p_{SE}	$=$	$223/1952$	(10)
Stuffbit Deletion (stuff bit converted into data bit)	p_{SD}	$=$	$79/488$	(11)
Stiffbit Creation (data bit converted into stuff bit)	p_{SC}	$=$	$157/1952$	(12)
Double Error (SD and SE within 11 bits)	p_{DE}	$=$	$1/244$	(13)

4.3 Residual Error Probability Formulae

There are residual error probability formulae for standard and extended ID format data frames covering the disjunct cases of errors affecting

- only the Data and CRC fields,
- Data, CRC and ID fields,
- the DLC, Data, CRC and ID fields,
- changing the IDE bit and
- errors deleting a SOF bit.

The formulae for the Data and CRC fields are given here, the others are too complex to be included in this paper.

The single channel state residual error probability p_r, see (1), for errors only in the Data and CRC fields (with a total length of 8DLC $+$ 15 $= L - 29$ bits) in standard CAN data frames is given by

$$p_{r,Data} = \sum_{k \geq 2} \left[\binom{L-29}{k} - \sum_{m=k}^{15} (L - 28 - m) \binom{m-2}{k-2} \right] p_E^k (1 - p_E)^{L-2-k} p_{dist}(k) p_{TE}(k) \tag{14}$$

with $p_{TE}(k)$ from (8) and $p_{dist}(k)$ from (6). The sum for $m \leq 15$ reflects the fact that error bursts with a length $m \leq 15$ can be safely detected by the 15 bit CAN CRC.

The expression is only slightly changed if extended format data frames are used:

$$p_{r,data} = \sum_{k \geq 2} \left[\binom{L-49}{k} - \sum_{m=k}^{15} (L - 48 - m) \binom{m-2}{k-2} \right] p_E^k (1 - p_E)^{L-2-k} p_{dist}(k) p_{TE}(k) \tag{15}$$

The other contributions to the residual error probability are shown in section 5, but the formulae are not given here in order to limit the length and complexity of this paper to a suitable amount.

5 Results

5.1 Normalized Residual Error Probabilities

Figures 10 through 13 give the one-state residual error probability as a function of the bit error probability of the channel. With (1), this can be used as an approximation for the normalized bit error probability

$$\frac{p_{res}}{q_{bad}} \approx p_r(p_{E_{bad}}) \tag{16}$$

if

$$(1 - q_{bad})\, p_r(p_{E_{good}}) \ll q_{bad} p_r(p_{E_{bad}})$$

or

$$\left(\frac{p_{E_{good}}}{p_{E_{bad}}}\right)^2 \ll \frac{q_{bad}}{1 - q_{bad}} \tag{17}$$

because the main contribution to the residual error probability in CAN comes from error cases with two bit errors per frame which makes $p_r(p_E)$ proportional to p_E for small p_E. The bad state probability q_{bad} can in this case be regarded as the probability of a message being transmitted during a bad state period of the channel.

In this way, the parameters q_{bad} and $p_{E_{bad}}$ of the channel are left open to insert theoretical or measured values for a specific transmission medium.

5.2 Contributions of Error Classes

Fig. 10 gives the contributions of the different error cases to the residual error probability for standard CAN frames with 8 data bytes per frame in a system with 10 stations.

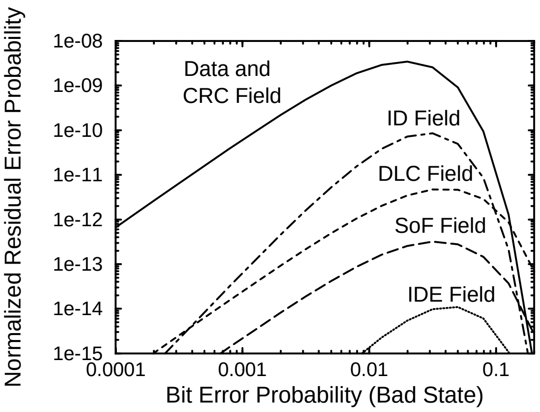


Figure 10: Contributions to the Residual Error Probability

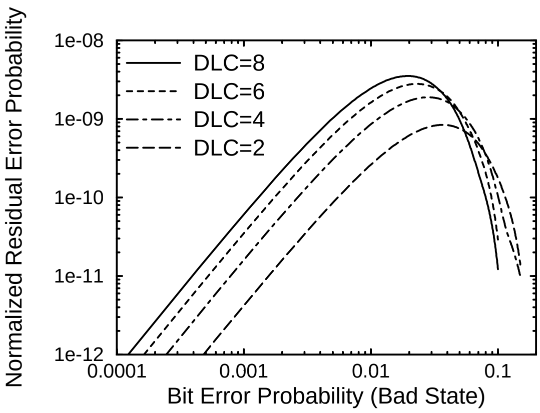


Figure 11: Influence of the Data Field Length

The errors affecting only the Data and CRC fields are the dominant contribution to p_r. For low bit error probabilities the residual error probability is dominated by error bursts with only 2 bit errors ($\Rightarrow p_r \sim p_E^2$). In the two-state channel model the worst case situation is $p_{res} = 3.5 \cdot 10^{-9} q_{bad}$, which occurs for $p_{E_{bad}} \approx 0.02$. Note that p_r decreases for high p_E, which comes from a high probability for error detection due to fixed field and stuff rule violations at high bit error probabilities.

5.3 Data Field Length

In the same system with 10 stations, the data field length of the CAN frames is now varied. Fig. 11 shows how the residual error probability decreases with decreasing frame frame length and the maximum points move towards greater bit error probabilities because the mean number of bit errors in one frame decreases with decreasing frame length.

5.4 Frame Format and Number of Stations

Fig. 12 shows the influence of the number of stations and the choice of standard or extended format on the residual error probability for 8 data bytes per frame.

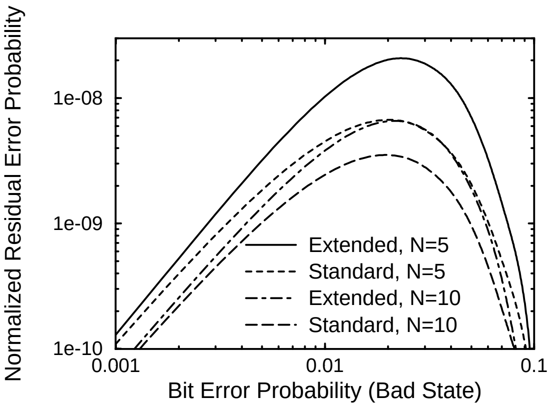


Figure 12: Influence of Format and Number of Stations

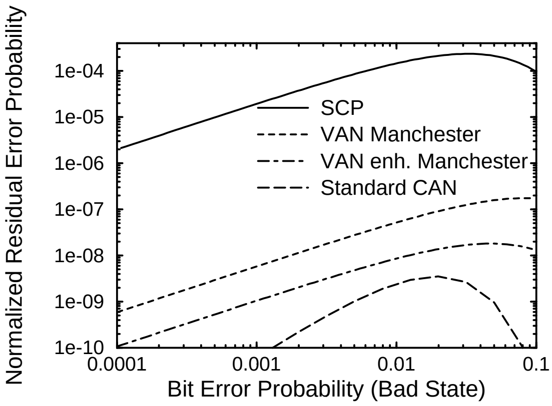


Figure 13: Comparison with SCP and VAN

The extended format frames show an increased residual error probability, mainly due to the increased frame length. These frames can also be changed into shorter frames more easily, as also the reception of a standard frame stemming from a disturbed extended frame is counted as a residual error. The number of stations on the bus (determining the effectivity of distributed error detection and the probability to have an undisturbed transmitter) also affects the residual error probability, varying by about a factor of 2 between $N = 5$ and $N = 10$ stations.

5.5 Comparison with other Protocols

The results of a different study, partially published in [5], are shown in Fig. 13, comparing CAN with the two other serial bus protocols for automotive applications standardized by the ISO [6]. The results have been derived with the same models and methods and under the same assumptions as in this paper. CAN and both VAN versions are shown with N=10 stations and 8 data bytes per frame. The SCP data field length was set to its maximum of 7. Both VAN (pure and enhanced Manchester bit coding) versions and SCP exhibit a significant amount of residual errors due to frame shortening by one bit error, which makes $p_r \sim p_E$.

5.6 Application for Traffic Mixes

In a real CAN system, the traffic on the bus will consist of frames of different data field lengths, frame formats and types. The residual error probability for extended data frames with DLC $= 8$, which is $p_{res} < 7.2 \cdot 10^{-9} q_{bad}$ for the ten station distributed channel model (or $p_{res} < 2.3 \cdot 10^{-8} q_{bad}$ for five stations), can be used as an upper bound for the residual error probability of any frame type and format sent over a channel of any bit error probability.

If a more precise evaluation is needed, in order to predict the residual error probability for a traffic mix, the total traffic can be decomposed into M traffic classes. Each traffic class C_i is described by the channel load ρ_i caused by this traffic class, the frame format f_i and type t_i as well as the length $L_i(\text{DLC}_i)$ of frames in this traffic class.

If $p_{t_i,f_i}(\text{DLC}_i)$ gives the residual error probability for class C_i, the combined residual error probability for the traffic mix can computed from (18).

$$p_{\textit{res,mix}} = \frac{\sum_{i=1}^{M} \rho_i \frac{1}{L_i} p_{t_i,f_i}(\mathrm{DLC}_i)}{\sum_{i=1}^{M} \rho_i \frac{1}{L_i}} \tag{18}$$

If T and v_T denote the total time that a CAN system is in use and the bus transmission speed, respectively, the total number n_R of messages that are undetectably disturbed during system usage is given by (19):

$$n_R = T \cdot v_T \cdot \sum_{i=1}^{M} \rho_i \frac{1}{L_i} p_{t_i,f_i}(\mathrm{DLC}_i) \tag{19}$$

6 Conclusions

Starting from a channel model allowing to be adapted to the properties of a transmission medium used in real systems, an analytical model was derived for error transformation due to bit stuffing and for several different error cases modifying different fields of a CAN frame. The results were displayed as a function of the bit error probability of the transmission channel and two decomposition formulae allow for computing the residual error probability for multi-state channel models and heterogeneous traffic mixes.

References

[1] Draft International Standard ISO-DIS 11519-2, ISO-DIS 11898

[2] J. Charzinski, “Fehlersicherungsverfahren im CAN-Protokoll.” *Institute of Communications Switching and Data Techniques*, University of Stuttgart, report No. 1067

[3] J. Unruh, H.-J. Mathony, K.-H. Kaiser, “Error Detection Capabilities of the CAN Protocol.” Robert Bosch GmbH, Stuttgart, Germany, Dec. 1989

[4] G. Funk, “Message Error Detecting Properties of HDLC Protocols.” *IEEE Trans. Communications* Vol. COM-30, pp.252–257, Jan. 1982

[5] J. Charzinski, O. Friedrichsohn, “The Security of Data Transport in Serial Bus Vehicular Networks.” *ATZ Automobiltechnische Zeitschrift* Vol. 96, May 1994, pp.324–330 (in German, with English abstract)

[6] Draft International Standard ISO-DIS 11519-3 and ISO-DIS 11519-4