Research Article

Quality and Reliability Engineering International

(wileyonlinelibrary.com) DOI: 10.1002/qre.1131

Published online 26 August 2010 in Wiley Online Library

DeviceNet Reliability Assessment Using Physical and Data Link Layer Parameters

Yong Lei,[a]*† Dragan Djurdjanovic[b] and Jun Ni[c]

Since the 1990s, the increasing deployments of networked automation systems led to increased manufacturing productivity, improved interchangeability of devices from different vendors, facilitated flexibility and reconfigurability for various applications and improved reliability, while reducing installation and maintenance costs. However, the reliability of a network has great impact on the reliability of a networked automation system. This paper presents a novel network reliability assessment method that provides diagnostic and prognostic information for DeviceNet. This work proposes a hybrid network error analysis method using combined physical and datalink layer features to provide complete communication log information. Furthermore, a network/node time to failure (bus-off) prediction algorithm was developed based on the analysis of the patterns of the interrupted packets on the network. The method developed in this study can be used for network reliability evaluation and diagnosis, facilitating better network maintenance decision making. A laboratory testbed was constructed and the experiments on network and node time to failure were conducted to demonstrate the concept. Experimental results show that the proposed method can fully reconstruct the communication log, and predict the network/node bus-off time successfully. Copyright © 2010 John Wiley & Sons, Ltd.

Keywords: DeviceNet; reliability assessment; physical and data link

1. Introduction

Industrial networks have been widely used in distributed automation systems since the 1990s to improve system flexibility and scalability, reduce cable and maintenance costs. In most systems, an industrial network is often considered as a transparent layer of the overall system with bounded data communication delay. In reality, as indicated in Reference[1], the reliability of a network has a great and profound impact on the performance of the networked automation system. The reliability of an industrial network can be affected by several factors including the electromagnetic interferences (EMI), cable defects, network bandwidth overuse as well as other factors[2, 3]. The growing complexity of distributed automation systems has led to the need to develop network health monitoring tools for network reliability assessment.

The literature shows that most of the research studies on networked automation systems are focused on the effects of network delays on the control system performance. The bounded transmission delay and worst-case deadline analysis on controller area network (CAN)-based networks have been comprehensively addressed in References[4–6]. Hasson *et al.*[7] analyzed the CAN node timing responses in a noisy setup, and used the probability of missing the transmission deadline as a reliability indicator. The packet transmission delay analysis on other network structures was also studied in the literature. For example, Lian *et al.*[8] analyzed the performance of ControlNet and Ethernet; Georges *et al.*[9] studied the packet transmission characteristics of switched Ethernets. Moon *et al.*[10, 11] studied the performance of IEEE 802.4 token ring bus in noisy environments. Research studies were also conducted on networked automation system dependability issues. Cauffriez *et al.*[12] and Jumel *et al.*[13] analyzed network dependability of distributed systems, especially at the controller level. Corno *et al.*[14] analyzed dependability of the CAN-based networked systems using simulation models. Recently, the performance analysis of CAN networks under different fault conditions had been studied. For example, Rodriguez-Navas *et al.*[15] developed a fault injection system to analyze the effect of network faults on the control system performance. Similar concept is also developed in Reference[16]. Tran[17] analyzed the multiple-bit error vulnerabilities in different bit error rate (BER) environments in a CAN network. However, little attention has been paid to develop methods to evaluate the network reliability, specifically, prediction of the time to shutdown the network.

[a] *The State Key Laboratory of Fluid Power Transmission and Control, Zhejiang University, Hangzhou, People's Republic of China*
[b] *Department of Mechanical Engineering, University of Texas, Austin, U.S.A.*
[c] *Department of Mechanical Engineering, University of Michigan, Ann Arbor, U.S.A.*
* *Correspondence to: Yong Lei, The State Key Laboratory of Fluid Power Transmission and Control, Zhejiang University, Hangzhou, People's Republic of China.*
† *E-mail: ylei@zju.edu.cn*

In a normal plant operation, network maintenance is not scheduled as high priority, unless the network performance is severely degraded and affects the system operations. In this case, maintenance engineers need to evaluate the network performance once the network is degraded. BER is a commonly used performance measure in practice. For example, Gaujal and Navet[18] used BER to calculate the rate of successful and unsuccessful transmissions, and a Markov model was established to estimate the bus-off hitting time. However, the dynamics of the transmit error counter (TEC) are not fully considered.

BER is an indicator of the network errors that are captured by the network interface. It usually does not provide fully the information on individual node performance and how a node affects the overall performance. For example, when the network has an intermittent connection problem, which is one of the most frequent and impacting failure modes observed in industrial networks, the network errors are induced by unreliable connections between the field device and the network backbone[3]. In this case, using BER could only indicate that the network experiences errors, but could not indicate which node causes the problems and how long the network nodes can sustain. For example, the intermittent connections may cause perfectly connected nodes to turn into the bus-off state and cause a system-wide shutdown, while the use of the BER could not identify the source of the problem. Moreover, BER is difficult to measure accurately in practice[19]. Therefore, it is desirable to provide the network reliability information, specifically the network time to failure prediction since it is a good measure for maintenance decision making purposes.

The goal of this study is to develop a systematic methodology that is able to provide network and node reliability assessment using online information, without interrupting the normal network operation. We focus our study on the DeviceNet network and use the network error information to conduct network time to failure prediction. We first introduce the DeviceNet (CAN) error confinement mechanism and construct a Discrete Time Markov Chain (DTMC) to describe the confinement rules of the network node. Then we introduce a network error analysis method to fully restore the network information, which will be used to assess the reliability of the network.

The remainder of the paper is organized as follows. We first briefly introduce DeviceNet (CAN) in Section 2, followed by the problem definition in Section 3. In Section 4, the proposed method is introduced in detail. The experiment testbed is described in Section 5, followed by the results and discussion in Section 6. The summary and future work is provided in Section 7.

2. Introduction to DeviceNet (CAN)

Fieldbus network architectures are developed to use networks as digital communication interfaces between controllers, sensors and actuators, which changes the system design from a point-to-point architecture to a fully distributed networked architecture. DeviceNet is a commonly used fieldbus protocol. It is an application layer protocol based on the standard CAN specifications as its physical and data link layer protocols. CAN is a serial communication protocol based on Carrier Sense Multiple Access/Arbitration on Message Priority (CSMA/AMP) media access method. The physical layer electrical connection, as defined in Reference[20], can be seen in Figure 1. This standard contemplates a bus with 2 V differential electrical signals, where the bit-stream of a transmission is synchronized at the physical layer. The logic states on the bus are defined as recessive ('1' logic) and dominant ('0' logic), where the terms recessive and dominant indicate that a dominant state will always cancel a recessive state. In DeviceNet (CAN), the protocol is message oriented and each message contains a specific priority defined by the message identifier. Below is a brief description of the DeviceNet communication and error handling protocol. More information about the DeviceNet can be found in its specification[21].

The data packet format of DeviceNet is shown in Figure 2. The total data frame includes Start of Frame (SOF), Arbitration Field (11-bit Identifier), Control Field, Data Field, Cyclic Redundancy Check (CRC) Field, Acknowledgment (ACK) Field, End of Frame (EOF) and Intermission (INT) Field. The size of the data field varies between 0 and 8 bytes. The arbitration field provides message prioritization as well as source and destination identification.

The network errors, which are usually caused by EMI, grounding problems and connection problems, will interrupt the normal communication and result in repeated packets or lost packets. According to the CAN error confinement mechanism, an error packet will be sent to the bus immediately once the node detects an error. After all the nodes finish sending the error packets, the bus will turn idle, and then the normal network traffic will resume.

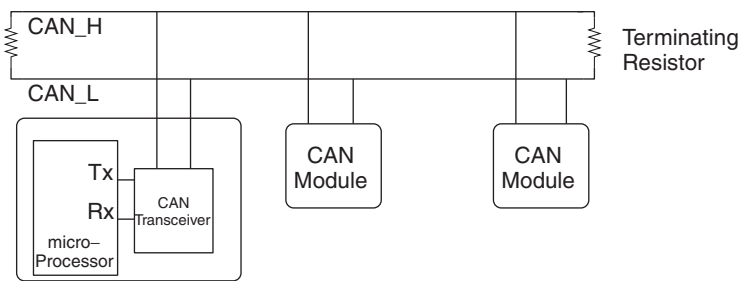

Figure 1. Physical layer connection recommended by ISO 11898

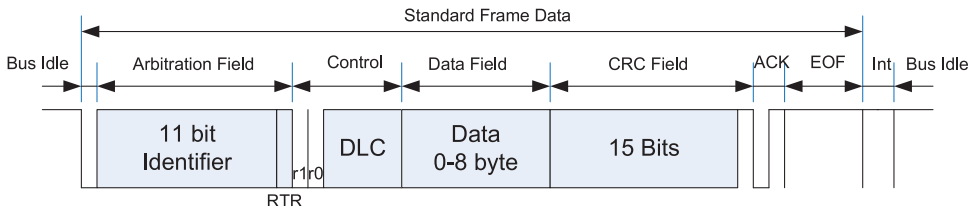

Figure 2. DeviceNet Data packet format (Standard CAN data frame)

3. Problem definition

The design philosophy of the error confinement mechanism of the CAN protocol is to reduce the disturbances of a problematic node gradually, so that the network can be protected from flooding by error packets. Each node is equipped with a TEC. According to the CAN specification, if the TEC value of a node crosses the threshold, it will turn to bus-off state, in which the node will not communicate with the rest of the system. However, from the industrial application point of view, the loss of a network node is intolerable due to safety and product quality concerns. Therefore, in this study, we concentrate on the network reliability in terms of the time that a network goes offline (that is, a node goes to a bus-off state). It is important that this measure be predicted before it occurs, without interrupting the normal network operation.

This problem becomes much simpler if the TEC counter of all the network nodes can be accessible, since the bus-off hitting time of a node depends directly on its TEC value. However, the error counters in most industrial products cannot be accessed online. Therefore, it is needed to develop a methodology to infer the node error status and estimate the bus-off hitting time from the observed error patterns in the existing network communication logs. Unfortunately, according to the CAN protocol design, the CAN interface hardware will discard the incomplete packet upon each error. As a result, the digital communication log is incomplete in the sense that it is unknown which node was transmitting when an error occurred.

Therefore, in order to evaluate the reliability of the network and its nodes in terms of bus-off hitting time, the following challenges must be addressed:

- How to develop a systematic method to automatically restore the communication log? As described previously, the available information does not provide complete communication information when an error occurs.
- How to estimate the bus-off hitting time of a node given the current error pattern? Since it is impossible to obtain the true value of the TEC in each node, a model needs to be developed to estimate and predict the trend of the TEC value.

This study is based on two assumptions: (1) there is only one master device (PLC) on the network, and (2) the communication setup is polling method only. These two configuration options are very common in automotive manufacturing plant networks.

4. Proposed methodology

As described in the previous section, our goal is to evaluate the reliability of the network and its nodes without interrupting the normal network communication. To do so, one should be able to estimate and predict the trend of the TEC values of the network nodes. The principal idea of the proposed method consists of the following steps. First, we model the error confinement principle of the TEC counter using a DTMC. Since it is difficult to obtain the historical and the current TEC values, we developed a new approach to predict the TEC values by collecting cycle-based communication information. Second, we developed a novel procedure to fully restore the detailed communication information by combining the unreliable long-term digital records with the short-term network error analysis records. Finally, based on the constructed discrete time Markov model, we predict the bus-off hitting time of the network using the restored communication log. Details of the proposed method will be introduced below.

4.1. Node bus-off hitting time model

According to DeviceNet (CAN) specification, the error counters of each node, TEC and Receiver error counter (REC), determine the node error status. Normally a node stays in the *error active* state to participate in the bus communication, and sends active error packets when errors occur. If multiple errors ($REC > 127$ and $127 < TEC < 255$) are experienced by the node, it will turn to the *error passive* state. If the node TEC is greater than 255, it will turn to bus-off state, in which no packet can be received or transmitted.

Figure 3 shows the CAN error state machine with the following three states:

- Error Active. The node can normally participate in network communication.
- Error Passive. The node can participate in network communication; however, it will send error passive flags instead of error active flags. Moreover, eight recessive bits will be sent before starting a new packet. If during sending these 8 bits, another node begins to transmit a packet, this node will turn to receiving mode.
- Bus off. The node cannot participate in the network communication unless reset by hardware or software after successfully observing 128 occurrences of 11 consecutive recessive bits.

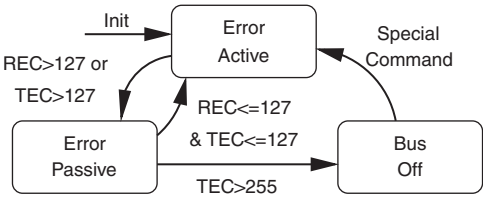

Figure 3. CAN error state machine

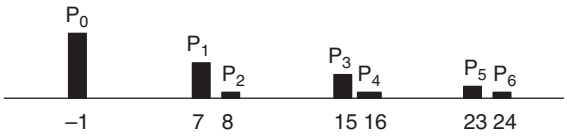

Figure 4. Histogram plot for TEC increments per polling cycle

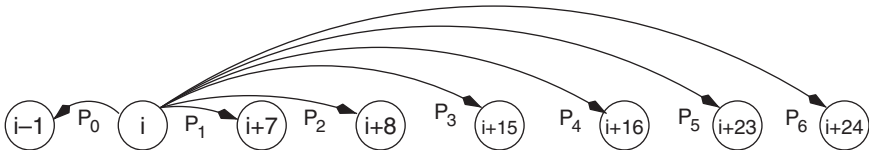

Figure 5. TEC state transition diagram using empirical probabilities calculated from observation data

According to the CAN specification, each time a packet is received or transmitted successfully, the value of the error counter is decreased by 1, and if the counter is zero, it will remain zero. Similarly, the value of the error counters will increase if error occurs. The simplified rules for modifying the error counters of the nodes are as follows:

- If a node successfully received a packet, REC is decreased by 1 if it is between 1 and 127. REC remains 0 if it was 0; if REC was greater than 127, it will choose a value between 119 and 127. If error occurs during receiving, TEC is increased by 8 and REC is increased by 1 or 9. (When error occurs in receiving mode, REC is increased by 1 first, then the TEC is increased by 8 when the error flag is sent. If a node detects a dominant bit at the first bit after its error flag, REC will be increased by 8).
- If a node successfully transmitted a packet, TEC is decreased by 1. Otherwise TEC is increased by 8, and REC is increased by 1 or 9 (using the same rule as above).

The complete error confinement rules are explained in Reference[22, pp. 24–25].

A network node can be forced to turn offline when its TEC value is greater than 255. As can be seen from Figure 3, the criterion for a node to reach bus-off state is determined only by the TEC. In this paper, we model the behavior pattern of the TECs using a DTMC in a polling setup, since the changing of TEC is only determined by the current value of TEC, and the time spent in the current state is irrelevant in determining the next state. We describe the statistics of the TEC increments per polling cycle of a node using a histogram. Figure 4 illustrates the concept of a histogram plot of TEC increments per polling cycle of a node.

If the histogram of TEC increments per polling cycle of a node can be obtained as well as the associated empirical probabilities, we could then construct the dynamics of the TEC values using DTMC. Figure 5 illustrates the TEC state transition diagram using empirical probabilities calculated from the example of the histogram of the TEC increments per polling cycle of a node. Therefore, the state transition matrix of a DTMC with 257 states, as shown in Equation (1), can be obtained from Figure 5. The state number 256 in Equation (1) is an absorbing state, which corresponds to the bus-off state of this node.

$$\mathbf{P}=\begin{array}{c|cccccccccccc} & 0 & 1 & 2 & \dots & 7 & 8 & 9 & \dots & 253 & 254 & 255 & 256 \\ \hline 0 & P_0 & 0 & 0 & \dots & P_1 & P_2 & 0 & \dots & 0 & 0 & 0 & 0 \\ 1 & P_0 & 0 & 0 & \dots & 0 & P_1 & P_2 & \dots & 0 & 0 & 0 & 0 \\ 2 & 0 & P_0 & 0 & \dots & 0 & 0 & P_1 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 254 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & P_0 & 0 & 0 & 1-P_0 \\ 255 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & P_0 & 0 & 1-P_0 \\ 256 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 \end{array} \quad (1)$$

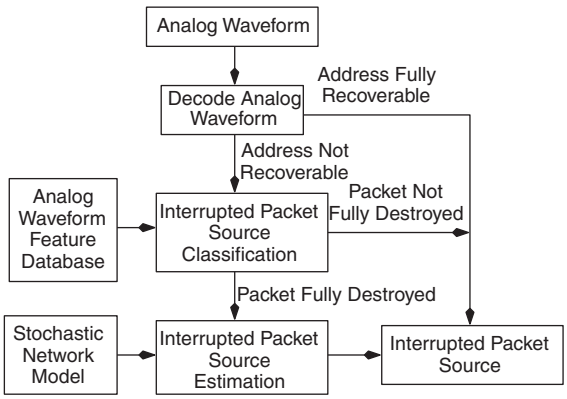

Figure 6. Flowchart of the interrupted packet analysis

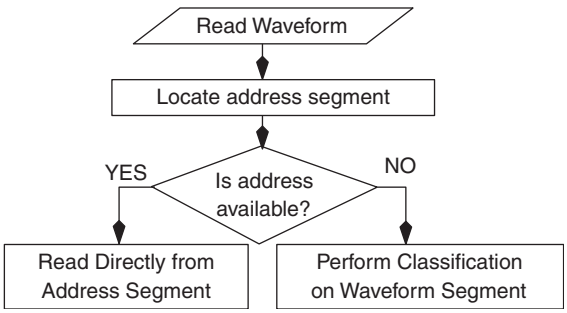

Figure 7. Flowchart of interrupted packet source identification

4.2. The interrupted network packet analysis

Instead of assuming a BER and inferring the probabilities needed to predict the bus-off time, we developed a network error analysis system to accurately evaluate how the TEC value of a node may change upon each error. As we introduced in the previous section, when an error occurs, the TEC values of each node will change depending on how the node involved in the error, or more specifically, which node's transmission is interrupted by the error. The error handling procedure of CAN protocol requires a node to discard the packet currently being received once an error is detected. Therefore, it is impossible to obtain the address of the interrupted packet from the digital interface, and analog waveforms must be used to extract this addresses. Using our data acquisition (DAQ) equipment developed for this study, we record every interrupted packet upon each network error, as well as the timestamp information. Therefore, one of the key components in the network error analysis is to determine the source address of the interrupted packet when an error occurs. Figure 6 illustrates the procedures for the interrupted packet analysis. When an error occurs, the recorded waveform will be decoded first to analyze the header address of the waveform segment. If it is available, then the source address of this waveform segment can be determined. Otherwise, classification is needed to identify the source of the waveform using the physical layer features extracted from the waveform segments from all the nodes. In addition, a stochastic network model is constructed to estimate the source of the interrupted packet in case the remnant information is not sufficient to conduct source identification. In this section, the detailed procedures of each function block are discussed.

4.2.1. The interrupted packet source identification using physical layer information. At the beginning of each normal packet, there exists an address segment. However the address segment of the packet can be damaged by an error packet, and in that case an address identification procedure is needed to recover the address. Therefore as illustrated in Figure 7, if the address segment remains intact, the address can be read directly from the analog waveform. Otherwise, a pattern classification method is employed to infer the address based on the analog waveform features.

When designing a classifier, the choice of features considerably affects the performance of the classifier. Although each feature represents certain physical or statistical meaning, putting all the features into the classification might not be efficient, and in some cases may result in worse instead of better classification performance. Hence, it is essential to find an appropriate set of features so that the separability can be preserved with the reduced feature set dimension.

The features extracted from the physical layer signals in[3] can be grouped into three categories.

- Dominant state features. Bit static features represent the static voltage profiles at each individual bit. Features included are Signal-to-Noise ratios (SNR), common mode voltage features, DC voltage.

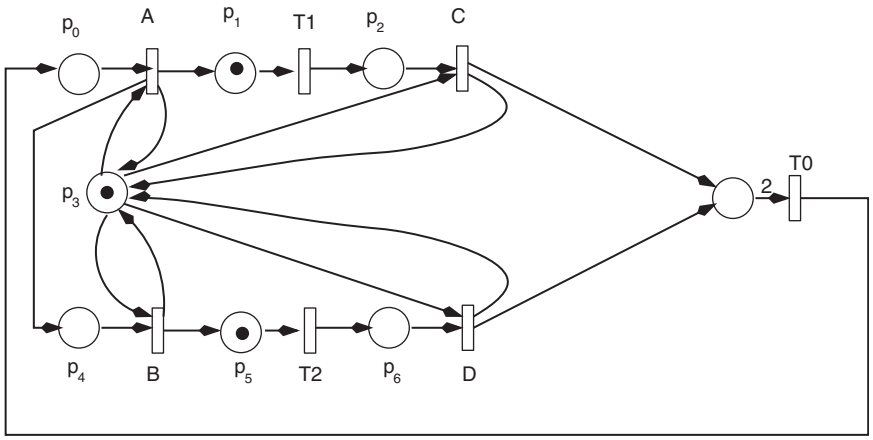

Figure 8. SPN model of a 3-node DeviceNet system (including PLC)

Table I. Specification of the transitions of the SPN of Figure 8

Transition	Rate	Semantics
T_0	λ_0	Single-server
A	λ_A	—
T_1	λ_{T1}	—
C	λ_C	—
B	μ_B	—
T_2	μ_{T2}	—
D	μ_D	—

- Recessive state features. Features are extracted using similar definitions as dominant state features.
- Dominant/Recessive state transition features. Bit transient features represent the voltage profiles between different logic states. Features included are voltage overshoot and voltage rise/fall time features.

Statistical analysis shows that a linear classifier using the dominant state features is sufficient to identify waveforms sent by different nodes, whereas features from other categories do not provide significant contributions. Therefore, the dominant state features are used in this study for the interrupted packet source identification.

4.2.2. Interrupted packet source identification using data link layer information. In most cases, the source address information about the interrupted packet can be successfully recovered using the techniques described previously. However, not all the interrupted packets can be recovered since some packets could be completely damaged or the available waveform segments do not provide sufficient information. In this scenario, we only have the timestamp information of the interrupted packet available. Therefore, the interrupted packet needs to be estimated from the packet trace information.

The traces of the network behavior can be summarized through the prefix-closed language L. L is a subset of E^*, the Kleene-closure of the event set E[23]. The post language $L+s$ is the set of possible continuations of a string s[24], i.e.

$$L+s=\{t\in E^*|st\in L\}.$$

The problem of estimating the source of interrupted packet now can be formulated as follows:
Given $L_a\subseteq L$, $\exists\mu\in E^*$ at given t satisfy

$$P_r\{L_a+\mu|L_a,t\}\geqslant P_r\{L_a+\alpha|L_a,t\}\quad\forall\alpha\in E^*, \tag{2}$$

where t denotes the inter-event time between L_a and the next event, and μ denotes the estimated interrupted packet.

To illustrate the concept, we constructed the model of a simplified 3-node (including PLC) DeviceNet system using a stochastic Petri Net (SPN), in which the concurrency and stochastic nature of DeviceNet can be described. Figure 8 illustrates this simple SPN. We modeled the system based on the PLC's batch polling setup since it is commonly used in practice. In this setup, all the polling commands are sent together to the sending queue of the interface chip. Another common setup is polling one node at a time, which significantly reduces the complexity of the problem. The specification of the SPN is shown in Tables I and II, and the values of the parameters are shown in Table AI, in the appendix.

In Figure 8, the transitions A and B represent the event when the PLC polling commands are sent by PLC transceiver queue to node 1 and node 2, respectively, while the transitions T_1 and T_2 represent the command receiving and data preparing operation

Table II. Markings of the SPN of Figure 8

M_0	=	$2P$									
M_1	=		P_0								
M_2	=			P_1	+		P_4				
M_3	=					P_2	+	P_4			
M_4	=	P	+					P_4			
M_5	=			P_1	+				P_5		
M_6	=					P_2	+		P_5		
M_7	=	P							P_5		
M_8	=			P_1	+						P_6
M_9	=					P_2	+				P_6
M_{10}	=	P	+								P_6
M_{11}	=	P	+				P_2				
M_{12}	=	P	+		P_1						

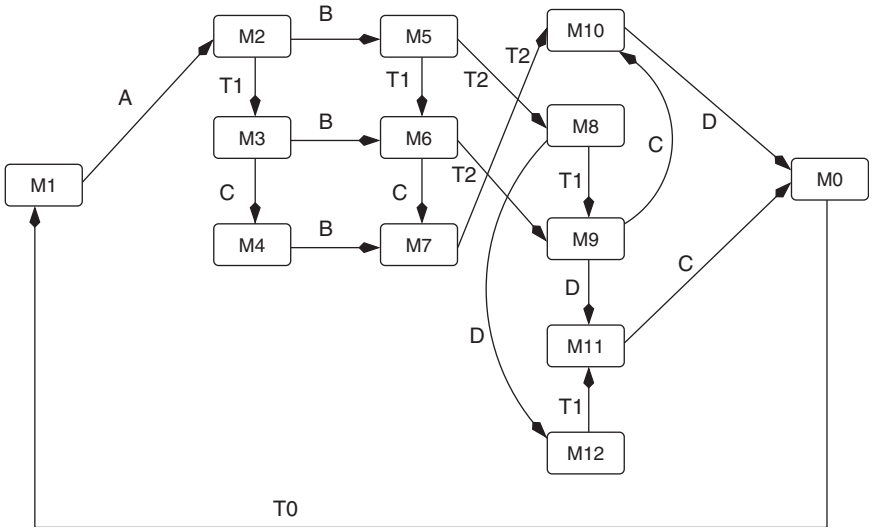

Figure 9. Reachability graph obtained from the SPN model in Figure 8

on the node 1 and node 2, respectively. Transitions C and D represent the events of the node polling responses sent by node 1 and node 2, respectively. Place p_3 denotes the bus availability, and transition T_0 denotes the PLC polling cycle.

As can be seen from Figure 8, the order of the events is determined by the firing sequence of the transitions. Therefore, given an event trace, the probabilities of the next possible events can be estimated by the firing rate of the transitions in SPN. Since polling commands (A, B) are sent sequentially from the transceiver queue, it is relatively easy to estimate the concurrency of one polling command with the node responses. However, in practice, it is needed to estimate the next possible node response when an intermittent connection error occurs. For example in Figure 8, given transitions A and B fired at t_A and t_B, respectively, the probabilities of transitions C and D being triggered at a given time t are given by the following equations:

$$P_r\{C|AB, t_A, t_B, t\} = P_r\{M_7 \cup M_{10}|M_5, t_A, t_B, t\}, \tag{3}$$

$$P_r\{D|AB, t_A, t_B, t\} = P_r\{M_{11} \cup M_{12}|M_5, t_A, t_B, t\}, \tag{4}$$

where M_5, M_7, M_{10}, M_{11} and M_{12} are the markings of the SPN shown in Table II and the reachability graph in Figure 9.

In the example described previously, Equations (3) and (4) can be solved by using Equations (5) and (6), respectively:

$$P_r\{M_7 \cup M_{10}|M_5, t_A, t_B, t\} = \left(1 + \frac{\lambda_{T1}\lambda_C}{(\lambda_{T1} - \lambda_C)\lambda_{T1}} \exp^{\lambda_{T1}(t-t_A)} + \frac{\lambda_{T1}\lambda_C}{(\lambda_C - \lambda_{T1})\lambda_C} \exp^{\lambda_C(t-t_A)}\right) \times \left(\frac{\mu_{T2}\mu_D}{(\mu_D - \mu_{T2})\mu_{T2}} \exp^{\mu_{T2}(t-t_B)} + \frac{\mu_{T2}\mu_D}{(\mu_{T2} - \mu_D)\mu_D} \exp^{\mu_D(t-t_B)}\right), \tag{5}$$

$$P_r\{M_{11}\cup M_{12}|M_5,t_A,t_B,t\}=\left(1+\frac{\mu_{T2}\mu_D}{(\mu_{T2}-\mu_D)\mu_{T2}}\exp^{\mu_{T2}(t-t_B)}+\frac{\mu_{T2}\mu_D}{(\mu_D-\mu_{T2})\mu_D}\exp^{\mu_D(t-t_B)}\right)\times\left(\frac{\lambda_{T1}\lambda_C}{(\lambda_C-\lambda_{T1})\lambda_{T1}}\exp^{\lambda_{T1}(t-t_A)}+\frac{\lambda_{T1}\lambda_C}{(\lambda_{T1}-\lambda_C)\lambda_C}\exp^{\lambda_C(t-t_A)}\right). \quad (6)$$

The decision of the source of the interrupted packet is made by using the modified *maximum a posteriori* (MAP) decision rule[25]:

$$\text{If } \frac{P_r\{C|AB,t_A,t_B,t\}}{P_r\{D|AB,t_A,t_B,t\}} > \frac{P(ABD(t,t_A))}{P(ABC(t,t_A))} \quad \text{decide D}$$
$$< \cdots \quad \text{decide C}$$
$$= \cdots \quad \text{decide the higher priority packet,} \quad (7)$$

where $P(ABC(t,t_A))$ and $P(ABD(t,t_A))$ denote the prior probabilities of the possible events after trace AB at time t, given t_A.

The method illustrated here can be extended to multiple node responses. In practice, due to the limited size of the PLC transceiver poll, the PLC will send the polling commands in small batches during one polling cycle. As a result, only limited number of network nodes (usually three or four nodes) will respond to the polling commands at each batch. Therefore, the complexity of the network model used in this section will not increase significantly.

4.3. Communication log registration

In the previous section, we introduced a method to identify the source of the interrupted packet upon each network error. We still need the complete communication log of each polling cycle to obtain the histogram needed to predict the time to bus-off. However, it is impractical to record all the analog signal waveforms of each polling cycle using high-speed acquisition method. Therefore, in order to fully reconstruct a complete communication data log, we propose an integrated approach that combines the physical and data link layer information, which synchronizes the analog-interrupted packet source identification results with the digital packet log from the DeviceNet interface.

The digital packet log is recorded using a DeviceNet (CAN) interface card, which has a separate clock that is different from the analog DAQ hardware. The reason behind this, as described previously, is that the DAQ hardware is designed to capture the network errors, which is not suitable for long-term data recording at high-speed sampling rate. Therefore, the differences between two clock systems need to be compensated. In addition, the timestamps of the errors from the interface card may not be accurate, since the time stamp of an error may have been delayed due to a higher processing priority for the next successful packet, and sometimes only one error is recorded in case of multiple casted errors. Therefore, a log registration algorithm is developed to handle the uncertainties in analog and digital logs.

Let $\mathbf{\Theta}=[C_0,C_s]'$ denote the registration parameter, where C_0 and C_s denote the location and scale factors of the clock mapping from DAQ clock system to digital interface clock system, respectively. Let $t_a(i)$ denote the timestamp of the ith error recorded by the DAQ hardware. Then we have the following clock mapping relationship:

$$t_a^d(i,\mathbf{\Theta})=\frac{t_a(i)-C_0}{C_s}, \quad (8)$$

where $t_a^d(i)$ denotes the mapped timestamp of the ith analog error. We define the fitness measure of the registration for the ith analog error using a radial basis function

$$\mathscr{L}(\mathbf{\Theta},i)=\sum_{j=1}^{N_d}\exp\left(-\frac{||x_d(j)-t_a^d(i,\mathbf{\Theta})||^2}{r^2}\right), \quad (9)$$

where N_d and $x_d(j)$ denote the number of errors and the timestamp of the jth error recorded by the digital interface card, respectively. r denotes the radius.

The optimal registration parameter $\mathbf{\Theta}^*$ can be determined by maximizing the fitness measure for all analog errors:

$$\mathbf{\Theta}^*=\underset{\mathbf{\Theta}}{\text{argmax}}\prod_{i=1}^{N_a}\mathscr{L}(\mathbf{\Theta},i), \quad (10)$$

where N_a denotes the number of errors recorded by the DAQ hardware. The solution of the optimization problem in Equation (10) can be obtained numerically using the well-known Newton–Raphson method.

4.4. Estimation of node and network bus-off hitting time

The state transition matrix in Equation (1) can be rearranged as the canonical form for an absorbing Markov chain:

$$\mathbf{P}=\left(\begin{array}{c|c}1 & \mathbf{0}\\ \hline \mathbf{R} & \mathbf{Q}\end{array}\right), \quad (11)$$

where $\mathbf{R}\in\Re^{1\times 256}$, $\mathbf{Q}\in\Re^{256\times 256}$.

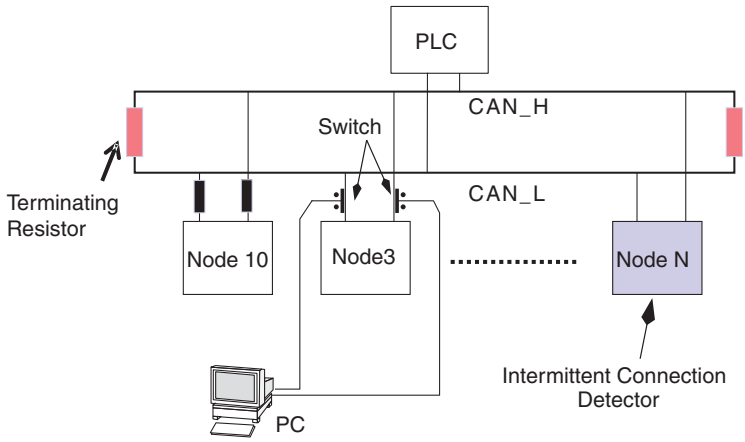

Figure 10. Schematic illustration for experiment design of intermittent connection emulation

Let **N** denote the fundamental matrix of an absorbing Markov chain. It can be defined as:

$$\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}. \tag{12}$$

Let **t** be the function giving the total number of steps needed to reach an ergodic set (including the original position). The mean and variance of **t** are:

$$\begin{aligned} E[\mathbf{t}] &= \tau, \\ Var[\mathbf{t}] &= (2\mathbf{N} - \mathbf{I})\tau - \tau_{\mathbf{sq}}, \end{aligned} \tag{13}$$

where **I** denotes the identity matrix, τ denotes the row sums of **N** and $\tau_{\mathbf{sq}}$ denotes the squaring of each entry in matrix[26].

Since it is impractical to obtain the node TEC value through the network online, we use **t**(0) as the optimistic node bus-off hitting time which represents the time needed from the initial state (TEC value 0) to the bus-off state (TEC value 256). The time to shut down the whole network is the minimal bus-off hitting time of all the nodes.

5. Experimental setup

5.1. Testbed

The schematic of the experimental setup is illustrated in Figure 10. The DeviceNet scanner is set to communicate using the polling method with a 10 ms polling interval. The network errors are induced by an intermittent connection, which is generated using a digital on–off switch controlled by a computer. The intermittent inter-event time follows a uniform distribution and the duration of the disconnection follows a Poisson distribution, with a mean width of 1 bit. Figure 11 shows the networked system, as well as the DAQ hardware used in this study.

5.2. DAQ and error capture

We developed a DAQ system to concurrently record the analog and digital packet information of the network. The analog waveforms are acquired at 100 MHz sampling rate, and the acquisition is triggered by an error packet detector that we developed for this study. The online error packet detector is developed to generate a trigger signal to the DAQ system once an error packet is captured. The generated triggers determine when the DAQ system should record the analog waveforms of error packets and the interrupted packets.

The time-stamped packet sequence is logged using a DeviceNet interface card[27]. The error packets are time stamped as error marks in the digital packet trace log. Figure 12 shows one segment of the digital log.

6. Experimental results and discussion

6.1. The interrupted packet identification

Figure 13 shows one data segment obtained using the DAQ system developed in this study. CAN_H and CAN_L of the DeviceNet voltage waveforms and error trigger signal are recorded. The falling edges of the error trigger signal mark the positions of the error packets.

Table III shows the source identification results of the interrupted packets, where Packet_3 denotes a packet sent by node 3. The first row represents the data log obtained using DeviceNet/CAN logging systems alone. The second row shows the identification

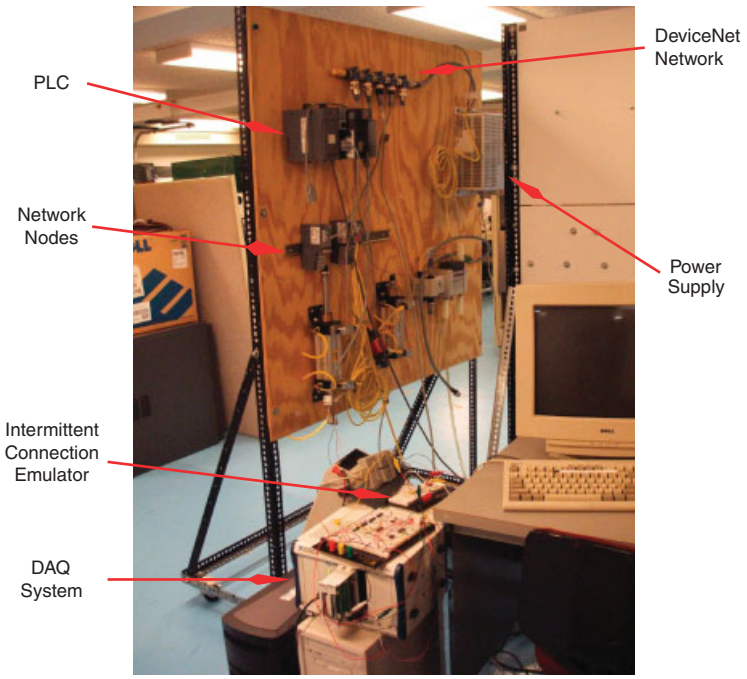

Figure 11. Data acquisition system developed for intermittent connection detection

```
;    Columns description:
;    +-Number of message
;    |       +Time offset of message (ms)
;    |       |         +ID of message (hex)
;    |       |         |     +Data length code
;    |       |         |     |   +Data bytes (hex) ...
;----+-   ---+---  ----+---  +  -+ -- -- ...
     1)          0       041D  0  -- -- -- -- -- -- -- --
     2)          0       0455  0  -- -- -- -- -- -- -- --
     3)          1       03C3  3  0C 00 00 -- -- -- -- --
     4)          1       03CA  1  00 -- -- -- -- -- -- --
     5)          3       0002  4  ERROR
     6)          5       03CA  1  00 -- -- -- -- -- -- --
     ....
```

Figure 12. One segment of packet sequence log contains error marks

result using the method presented in this paper. In this data segment, a packet sent by node 3 is interrupted by an error. After normal network communication is resumed, the packet sent by node 10 is interrupted by another error. The last packet is sent by node 3. It can be seen from the table that our method can fully recover the sequence of the events, while the digital log alone can only indicate the presence of the errors.

6.2. Node bus-off hitting time estimation

Figure 14 shows the histogram plots of TEC increments per polling cycle of node 3 under different frame error rate (FER) settings. As can be seen with the increasing communication interruptions by the intermittent connection errors, the probability of successful transmission of a packet drops. The increase in the TEC histogram shows an increasing trend toward the bus-off state.

Figure 15 shows the predicted mean and variance of the node bus-off hitting time under different FERs using the method proposed in this study. As can be seen from the figure, the node bus-off hitting time decreases significantly when more errors appear on the network. In this study, we compared the observed node bus-hitting times (10 experiments) with the predicted values at FER = 0.056 (error rate/frame). At this FER setup, the bus-off hitting time can be observed in a limited time, without violating other assumptions made in this study. We found that the true node bus-off hitting times fell in the confidence region of the estimated values and close to the mean estimates, which indicates that the proposed method is effective in the bus-off time prediction.

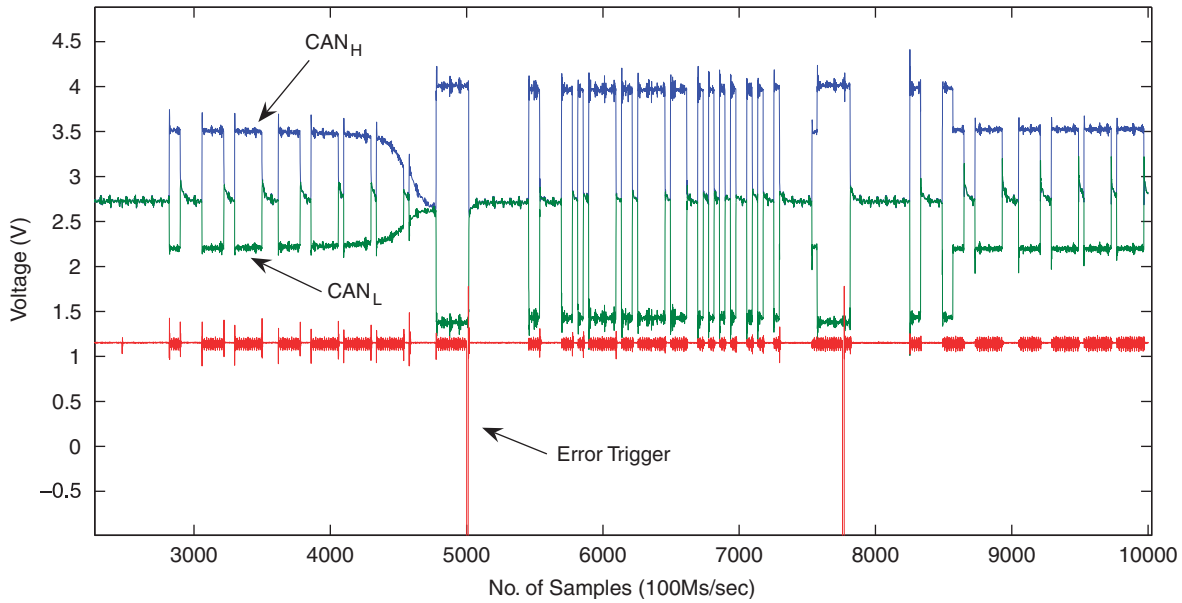

Figure 13. Example of the interrupted packets source identification using analog waveforms acquired upon each network error

Table III. Source identification results of the data segment in Figure 13

Commercial System:	ERROR		ERROR		Packet_3
System developed in this study:	Packet_3	Error	Packet_10	Error	Packet_3

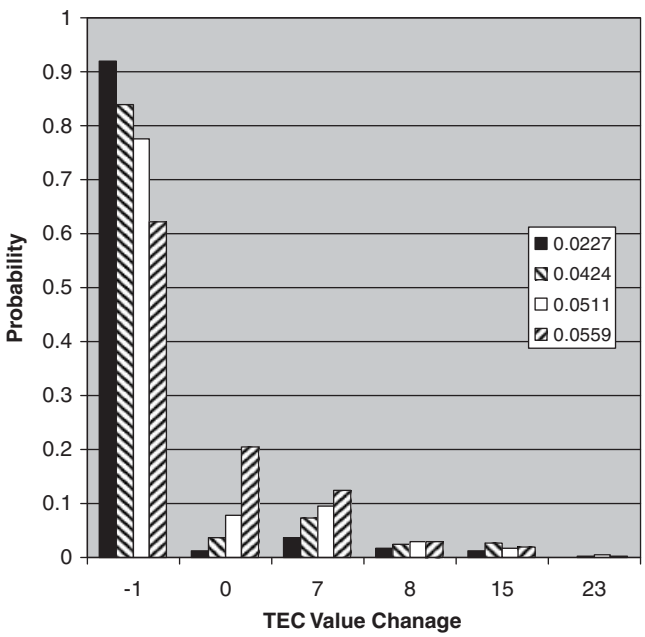

Figure 14. Histogram plots of a node TEC increments per polling cycle under different test frame error rates (FER)

7. Summary

In this study, a novel network reliability assessment method is proposed. It is based on passively observed network information obtained from the physical and data-link layer data. A hybrid analysis method is developed to identify the addresses of the nodes that originated the interrupted packets so that the complete information about each network error can be restored using the collected data. A method to predict the node bus-off hitting time using a discrete time Markov model is developed. A testbed is constructed to generate the intermittent connection-induced errors, and experiments are conducted to prove the concept. The experimental results show that the observed node bus-off hitting time measurements agree with the values predicted using

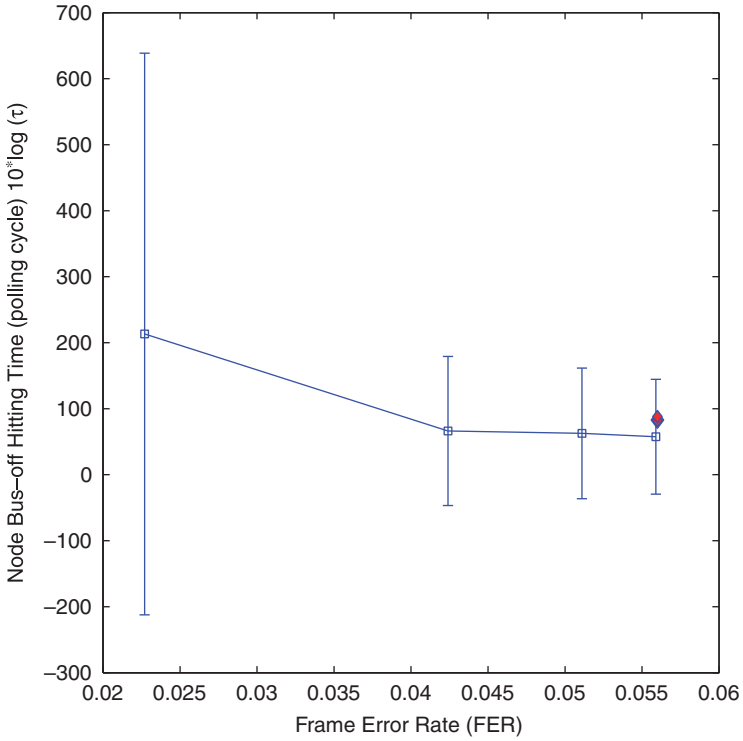

Figure 15. Observed and predicted node bus-off hitting time under different frame error rates. The filled squares on the right denote the observed bus-off hitting times at FER = 0.056

the proposed method. The future work includes improving of the proposed method by considering more complex network communication setups coexisting with the polling method, implementation of an active node TEC value query method to reduce the estimation variance and field testing in industrial environments.

References

1. Ye N. Secure, reliabile computer and network systems. *Quality and Reliability Engineering International* 2002; **18**(3):iii.
2. Prodanov W, Valle M, Buzas R. A controller area network bus transceiver behavioral model for network designs and simulation. *IEEE Transactions on Industrial Electronics* 2009; **56**(9):3762–3771.
3. Lei Y, Djurdjanovic D, Barajas L, Workman G, Biller S, Ni J. DeviceNet network health monitoring using physical layer parameters. *Journal of Intelligent Manufacturing* 2010; DOI: 10.1007/s10845-009-0291-9.
4. Tindell K, Burns A, Wellings AJ. Calculating controller area network (can) message response times. *Control Engineering Practice* 1995; **3**(8):1163–1169.
5. Navet N, Song Y-Q, Simonot F. Worst-case deadline failure probability in real-time applications distributed over controller area network. *Journal of Systems Architecture* 2000; **46**(7):607–617.
6. Rufino J, Veríssimo P. A study on the inaccessibility characteristics of the Controller Area Network. *Proceedings of the Second International CAN Conference*, CiA, London, England, October 1995; 7.12–7.21.
7. Hasson HA, Nolte T, Norstrom C, Punnekkat S. Integrating reliability and timing analysis of CAN-based systems. *IEEE Transactions on Industrial Electronics* 2002; **49**(6):1240–1250.
8. Lian F-L, Moyne JR, Tilbury DM. Performance evaluation of control networks: Ethernet, Controlnet, and DeviceNet. *Control Systems Magazine, IEEE* 2001; **21**(1):66–83.
9. Georges J-P, Rondeau E, Divoux T. Evaluation of switched ethernet in an industrial context by using the network calculus. *Proceedings of Fourth IEEE International Workshop on Factory Communication Systems*, Västerås, Sweden, 2002; 19–26.
10. Moon H-J, Park H-S, Kim D-W, Kwon W-H. Analysis of the IEEE 802.4 token-passing mechanism with noise. *Proceedings of the 19th Annual International Conference on Industrial Electronics, Control, and Instrumentation*, Maui, Hawaii, U.S.A., vol. 1, 15–18 November 1993. IEEE: New York, 1993; 541–546.
11. Moon H-J, Park HS, Ahn SC, Kwon WH. Performance degradation of the IEEE 802.4 token bus network in a noisy environment. *Computer Communications* 1998; **21**(6):547–557.
12. Cauffriez L, Conrard B, Thiriet J, Bayart M. Fieldbuses and their influence on dependability. *Instrumentation and Measurement Technology Conference, 2003. IMTC '03. Proceedings of the 20th IEEE*, vol. 1, 2003; 83–88.
13. Jumel F, Thiriet J-M, Aubry J-F, Malasse O. Towards an information-based approach for the dependability evaluation of distributed control systems. *Proceedings of the 20th IEEE Information and Measurement Technology Conference*, Vail, CO, U.S.A., vol. 1, 20–22 May 2003. IEEE: New York, 2003; 270–275.
14. Corno F, Perez J, Ramasso M, Sonza Reorda M, Violante M. Validation of the dependability of can-based networked systems. *Proceedings—IEEE International High-level Design Validation and Test Workshop, HLDVT*, Sonoma Valley, CA, U.S.A., 2004; 161–164. ISSN 1552-6674.
15. Rodriguez-Navas G, Jimenez J, Proenza J. An architecture for physical injection of complex fault scenarios in can networks. *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*, Lisbon, Portugal, vol. 2, 2003; 125–128.
16. Reorda MS, Violante M. On-line analysis and perturbation of CAN networks. *Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, Cannes, France, 2004.

17. Tran E. Multi-bit error vulnerabilities in the controller area network protocol. *Master's Thesis*, Carnegie Mellon University, 1999.
18. Gaujal B, Navet N. Fault confinement mechanisms on can: Analysis and improvements. *IEEE Transactions on Vehicular Technology* 2005; **54**(3):1103–1113.
19. Ferreira J, Oliveira A, Fonseca P, Fonseca J. An experiment to assess bit error rate in CAN. *Proceedings of the Third International Workshop on Real-time Networks*, Catania, Italy, 2004.
20. ISO11898-1. Road vehicles—controller area network (CAN) part 1: Data link layer and physical signalling, *ISO11898-1*, 2003.
21. Open DeviceNet Venders Association. *DeviceNet Specifications* (2.0 edn), 1997.
22. Robert Bosch GmbH. *CAN Specification*, Version 2.0, 1991.
23. Cassandras CG, Lafortune S. *Introduction to Discrete Event Systems*. Springer: Berlin, 1999.
24. Thorsley D, Teneketzis D. Diagnosability of stochastic automata. *Proceedings of the IEEE Conference on Decision and Control*, Hawaii, U.S.A., vol. 6, 2003; 6289–6294.
25. Ludeman LC. *Random Processes*: *Filtering, Estimation, and Detection*. Wiley: New York, 2003.
26. Kemeny JG, Snell JL. *Finite Markov Chains*. Springer: New York, 1976.
27. Molex Incorporated, U.S.A. Available at: http://www.molex.com [2004].

Appendix A: Parameters used in the SPN model

Table AI shows the parameters used in the SPN model. The parameters are obtained from the specifications of the products connected to the network used in our experiments, and verified through network operation using one PLC and each individual node.

Table AI. Identification result of unknown parameters in SPN model in Figure 8

Parameter	Estimated value (s^{-1})
λ_A	2×10^4
λ_B	2×10^4
λ_C	2×10^4
λ_D	2×10^4
λ_{T1}	7.593×10^4
λ_{T2}	7.123×10^4

Authors' biographies

Dr Yong Lei is an assistant professor in the Department of Mechanical Engineering at Zhejiang University. He received his BS in Control Science and Engineering from Huazhong University of Science and Technology, China, his MS in Machine Building and Automation from Tsinghua University, and PhD in Mechanical Engineering from the University of Michigan, Ann Arbor. His research interests include intelligent maintenance systems, monitoring and fault diagnosis of networked automation systems, and statistical quality control.

Prof. Dragan Djurdjanovic is an assistant professor in the Department of Mechanical Engineering at the University of Texas, Austin. He obtained his BS in Mechanical Engineering and in Applied Mathematics in 1997 from the Univ. of Nis, Serbia, his MEng in Mechanical Engineering from the Nanyang Technological Univ., Singapore in 1999, and his MS in Electrical Eng. (Systems) and PhD in Mechanical Eng. in 2002 from the Univ. of Michigan, Ann Arbor. He co-authored 36 published or accepted journal publications, 2 book chapters, and 25 conference publications. He is the recipient of several prizes and awards, including the 2006 Outstanding Young Manufacturing Engineer Award from the Society of Manufacturing Engineers (SME), 2005 Teaching Incentive Award from the Dept. of Mechanical Eng. of the University of Michigan, Nomination for the Distinguished PhD Thesis from the Dept. of Mechanical Eng., University of Michigan in 2003, and The Outstanding Paper Award at 2001 SME North American Manufacturing Research Conference. His research interests include advanced quality control in multistage manufacturing systems, intelligent proactive maintenance techniques and applications of advanced signal processing in biomedical engineering.

Prof. Jun Ni is the Shien-Ming (Sam) Wu Collegiate Professor of Manufacturing Science and Professor of Mechanical Engineering at the University of Michigan, U.S.A. He is the founding Dean of the University of Michigan—Shanghai Jiao Tong University Joint Institute located in Shanghai, China (2006–2011). He serves as the director of the S. M. Wu Manufacturing Research Center and the co-director of a National Science Foundation sponsored Industry/University Cooperative Research Center for Intelligent Maintenance Systems at the University of Michigan. Among the many honors and awards that Professor Ni received are the 2009 William T. Ennor Manufacturing Technology Award from American Society of Mechanical Engineering (ASME), the 2007 Education Excellence Award from the University of Michigan (UM), the elected Fellow of ASME in 2004, the 2003 Magnolia Silver Medal Award from Shanghai Municipal Government, the elected Fellow of Society of Manufacturing Engineers (SME) in 2002, the 1999 Cheung-Keung Endowed Professorship Award from the Ministry of Education of China's, the 1994 Presidential Faculty Fellows Award from the National Science Foundation. He also received many Best Paper Awards from SME/NAMRI, ASME/MSEC, ICFDM, SRC/TechCon, etc. His research interests are in manufacturing process modeling, analysis and prediction; precision engineering and metrology; cutting tool development, quality control methods, intelligent maintenance systems, monitoring, and fault diagnosis.