Probabilistic Delay Time Analysis in Networked Automation Systems

Jürgen Greifeneder and Georg Frey

Juniorprofessorship Agentbased Automation
Erwin-Schrödinger-Str. 12, D-67653 Kaiserslautern, Germany
{greifeneder | frey}@eit.uni-kl.de

Abstract

The introduction of non-deterministic networks to the automation field leads to new questions in the analysis of the resulting systems often called networked control systems (NCS). To determine reaction times not only the cycle times of the controllers but also the network delays have to be taken into account. The problem further increases in reliability analysis. Here the failures of a variety of components have to be considered. For this purpose probabilistic model checking (PMC) is a promising formal approach. This paper investigates the application of existing techniques and tools from PMC to the analysis of delay times in NCS under the consideration of component failures. A key idea in the presented approach is that the nature of signals in automation systems is taken into account. A case study shows how properties, relevant for the reliable functioning of an automation system, can be checked.

1. Introduction

Modelling and analysis of automation systems requires not only detailed knowledge about their functional aspects, but also about their real time behavior. Especially it is necessary, to know about possible delays. Most control algorithms need to communicate with their process hardware (i.e., sensors and actuators often abbreviated I/O for inputs and outputs) within bounded time intervals as the control algorithm will fail, otherwise. This leads to the specification of properties like the following:

1. "A reaction to a change in a sensor value will be issued within 200 ms."
2. "A change in a sensor value that stays active only for a time interval (pulse) of 5 ms is detected by the controller."

In classical structures using a single controller and directly connected I/O the answer to these questions depends only on the cycle time of the controller. Considering distributed systems with controllers communicating with the I/O over networks the problem is much harder. Here the network delay (typically not given by a constant value but by a distribution) has to be taken into account. To avoid worst-case analysis that often leads to infeasible demands on the control systems hardware the properties could be relaxed by introducing probabilistic bounds. This leads to properties like:

1. "With a probability of at least 99.9% a reaction to a change in a sensor value will be issued within 200ms."
2. "A change in a sensor value that stays active only for a time interval (pulse) of 5 ms is detected by the controller in 85% of all cases."

To answer questions like this, simulation is infeasible for complex systems since a probabilistic solution will need a very long simulation time. The problem is even bigger if the analysis is extended from a simple performance check to a detailed reliability analysis where the possibilities of failures in the components are considered during the analysis. In this case the system under consideration contains very short cycle times of a controller together with very long mean times between failures (MTBFs) of the components.

The formal description of systems like this leads to models that contain time, stochastic distributions and probabilistic choice. A new formal technique for the description and analysis of systems and properties like the ones described above is Probabilistic Model Checking (PMC). PMC uses an extension of CTL (PCTL – Probabilistic Computation Tree Logic, [1]) to specify properties over systems described by Markov models. Model checking algorithms and tools are available [2, 3] that allow the proof of properties like the above mentioned ones, on a given system. Due to spacial limitations the reader is referred to [4, 5] for more information on PMC.

In the presented work PMC is applied to delay time analysis in Networked Control Systems (NCS) under consideration of component failures. A key idea in the presented approach is that the nature of signals in automation systems is taken into account. The properties to be checked are not directly related to system failures but to the loss or delay of process relevant information.

The rest of the paper is structured as follows. The next section describes the used modelling approach and the considered systems and properties in more detail. A small case study in Section 3 is introduced to explain the method and the derived results. Section 4 concludes the paper and gives an outlook on further work.

Greifeneder, J. and Frey, G.: *Probabilistic Delay Time Analysis in Networked Automation Systems.* Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2005, Catania, Italy, Vol. 1, pp.1065-1068

2 Problem description and modelling approach

Dealing with breakdowns in redundant data networks, it is not sufficient to know about the probability of a specific breakdown. One also must discuss the time necessary for the determination of the breakdown itself and additionally of the time needed to regenerate the information lost in this breakdown. Finally, the probability that information is lost at all, must be considered. That means: determining the probability of information not arriving within a given time frame.

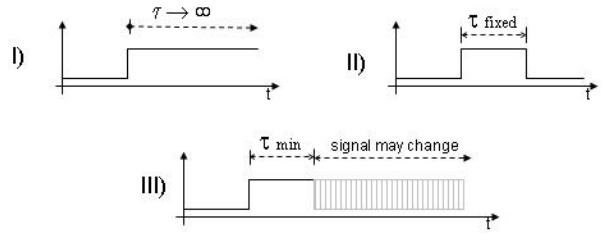

Figure 1. Different kinds of signals.

In an automated control system, there are three kinds of signals to discuss (cf. Figure 1):

I) A reaction is required within a specified time. An example for this kind is the emergency stop. After it occurs, it will not be reset until the message is received and an action could take place. Question: When will the signal be discovered?

II) A sensor, showing the signal only for a fixed time period τ_{fixed}. After that period the information is lost, if it has not been read out. Question: Will the signal be discovered?

III) The combination of the above mentioned ones. For example a multi valued signal, that changes its value with a probability p_{ch} and keeps its value until the next change (at least for τ_{min}). Therefore there are two questions:
   a) Will the signals change be discovered before it changes again and
   b) when will a signal change be discovered?

The networked automation systems under consideration consists of one or more controllers connected to the process by a network build from Switches and IO-modules. The components exchange signals and information using TCP/IP. Some components exist more than once and the model of the system has to be scalable in an easy way. Therefore it makes sense to model the different components separately and connect them to build the system discussed.

For the modelling of the components, in this work, probabilistic timed automata are used, including the method of digital clocks [6]. This involves the opportunity to eliminate the non-deterministic choice by reducing the time-space to a discrete set of steps. Doing so, all the significant choices are done synchronously. Consequently, the model will not represent the exact occurrence of an event but only the fact, that it occurred within the last time step. Yet, some systems will not allow to determine the optimal length of such a time step in the way, that each participating system will perform exactly one step in between. By using a time step frequency which is faster than the fastest system change, the size of the model has the tendency to increase exponentially. The models are implemented in Prism [2, 3] a probabilistic model checker from the University of Birmingham.

3 Case study

To illustrate the applicability of the method a case study is presented. The structure of the considered system is given in Figure 2. The model comprises a PLC, which represents the controller and asks for sensor data periodically (25 time steps). This inquiry is then passed over the network, represented by the Switch (Sw). From there it is again passed to the IO-Module (IO), which reads the actual data from the sensor and passes this information back through the network to the PLC.

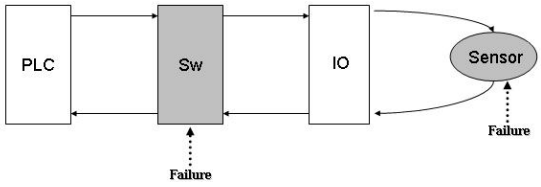

Figure 2. Network of the case study.

Each packet needs some time to pass through the network. In this case study an uniformly distributed time is assumed ranging from one to five time steps. To read a sensor value two time steps are required.

The network and the sensor are assumed to fail. For the network an expected mean time between failures of 10000 time steps is assumed, for the sensor 100 time steps. After such a breakdown, the network needs 50 time steps to recover, while the sensor only needs 35 time steps.

The controller is built robust enough, such that it holds the measured value unchanged unless it receives another value. Therefore, a failure in the network or the sensor has no consequences if the measured value does not changed during the respective downtime.

The first experiment deals with an emergency sensor (cf. Figure 1, case I). In this example the emergency stop has an occurrence probability of $p_{em} = 0.0001$. The question for this case is, how long it will take, until the controller recognizes this signal.

Figure 3 shows the state chart of the corresponding Prism module. Time, IOSw and SwPLC are synchronised automaton transitions, synchronising the switching operations together with these of other modules, e.g. the timer-module, the switch-module and the module describing the PLC. Time therefore refers to the time step module, IOSw indicates, that the package has arrived in the network and SwPLC means, that the package has been successfully transmitted.

Greifeneder, J. and Frey, G.: *Probabilistic Delay Time Analysis in Networked Automation Systems.* Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2005, Catania, Italy, Vol. 1, pp.1065-1068

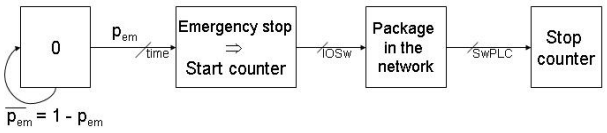

Figure 3. Emergency sensor.

Figure 4 depicts the probability of a delaytime introduced by the system between the occurrence and the recognition of an emergency signal in absence of failures. The value of 4% at the time step 20 means that with a probability of 4% the emergency signal will be detected by the controller in the 20^{th} time step after it occured. To come back to the typical property given in the introduction: Summing up the values in the curve it could be shown that, e.g., with a probability of 66% the emergency signal is detected in less than 20 time steps.

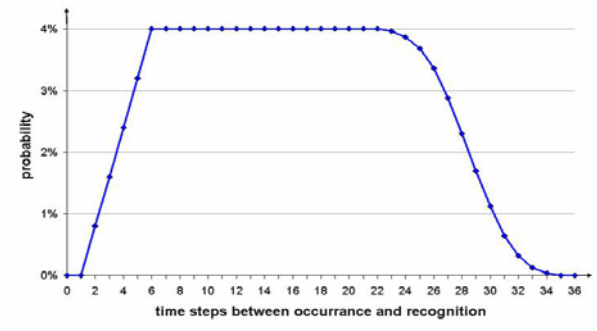

Figure 4. Time steps emergency signal needs to be recognized (no failures).

The graph can be split into three regions of interest. The first region (from 2 to 6 time steps) shows the case, where the request arrives to the sensor, in the moment when the value changes or shortly afterwards. In this case, the recognition of the signal is only delayed by the network (1 to 5 time steps). This explains the linear rise of the probability. From 6 to 22 the probability values remain constant. In the third region from 23 to 34 the graph sharply declines. The high delays occur, if the emergency stop is activated shortly after the IOSw was triggered. The quadratic shape of the curve can be explained as follows. In contrary to the increasing side, the delay caused by the switch (1 to 5 time steps) is considered twice (one way and back) in the decreasing region. Furthermore, the switching delays of two sequent request packets are not related (i.e., linearly independent).

In the case of a network failure probability of $p_{nf} = 0.0001$, which is equal to 10000 time steps, the probability distribution up to 34 time steps is nearly the same as without the failure. However, there exist packages arriving after more than 34 time steps (cf. Figure 5).

This is caused by the network downtime of 50 time steps and the cycle time of 25. The network may fail again directly after it has restarted. Then of course, it is possible that the emergency signal can not be passed for a very long period. This can happen with a probability of less than 10^{-6}. To conclude, a signal will not take longer than 110 time steps with a probability of 99.98858 %.

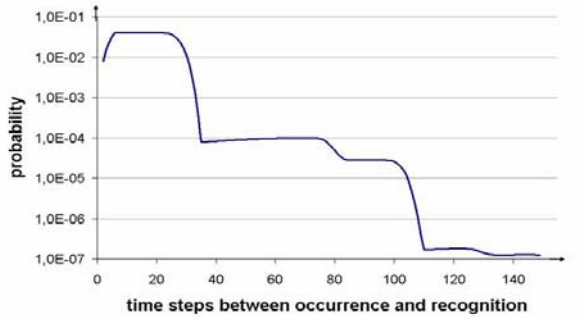

Figure 5. Time steps emergency signal needs to be recognized under consideration of network failures (logarithmic scale).

The second experiment deals with a sensor holding the signal for a fixed time (cf. Figure 1, case II). The question to answer in this case is: will the signal be discovered while it is visible? The corresponding automaton of this module can be found in Figure 6. For reasons of readability the state "package in the network" contains several substates which are not displayed in the figure. As this can be interpreted as a special case of signal type III, the results are not presented in detail here.

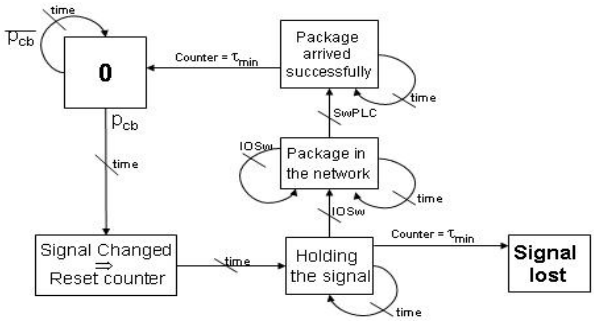

Figure 6. State chart of sensor type II.

The third experiment, the structure of which is shown in Figure 7, describes the probability of a change not being recognized over the time axis. The signal used is of type III (cf. Figure 1, case III). A differential quotient, using the last five time steps, was used to determine the probability that a change of the value would not be discovered.

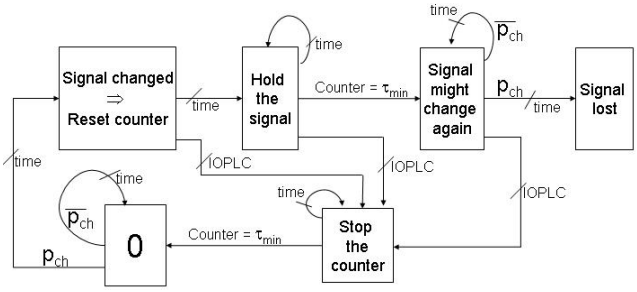

Figure 7. State chart of sensor type III.

Greifeneder, J. and Frey, G.: *Probabilistic Delay Time Analysis in Networked Automation Systems.* Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2005, Catania, Italy, Vol. 1, pp.1065-1068

While Figure 8 describes the distribution for the network failure alone, Figure 9 depicts the failure of the sensor. The parameters signal changing rate p_{ch} and minimal hold time τ_{min} are chosen to 0.01 and 35 time steps, respectively. The probability of a sensor failure is assumed to be 0.01 (compared to 0.0001 for the network). Therefore, the graph in Figure 9 rises to higher probabilities than the one in Figure 8. Both figures describe the probability of a signal loss over the number of time steps occurred since the beginning of the simulation. The value of 0.0037% at time step 100 of Figure 8 means that a significant package loss can occur in the intervall from time step 95 to 100 with a probability of $3.7 \cdot 10^{-5}$. There are three regions of interest: Up to 35 time steps, a signal loss will not occur. In the second region Figure 8 and Figure 9 differ. In Figure 8 this second region contains 50 time steps, only caused by the recovery time after a network's failure of 50, which is twice the cycle time. In Figure 9 the recovery time after a sensor's failure is assumed to 35 time steps. Together with the cycle time of 25 the second step in the graph can be explained. In the third regions the probabilities nearly remain constant.

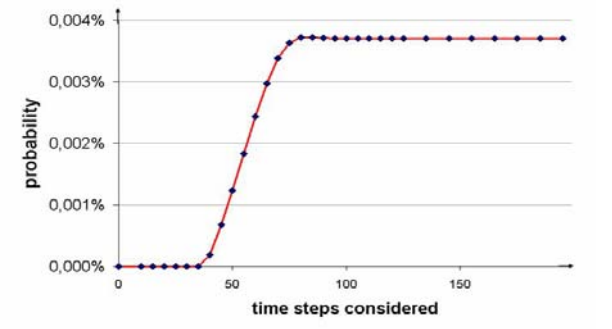

Figure 8. Probability of signal (type III) loss due to a network failure.

Concluding, the probability of losing a signal change is approximately $3.7 \cdot 10^{-5}$ for a network failure and $1.3 \cdot 10^{-3}$ for a sensor failure.

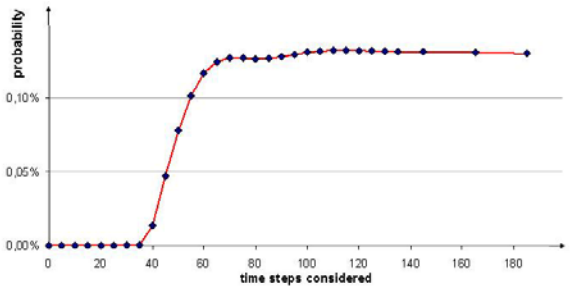

Figure 9. Probability of signal (type III) loss due to sensor failure.

Finally, Table 1 illustrates the influence of the minimal hold time τ_{min} on the probability p_1 of losing at least one value within the first 151 time steps ([ts]) due to a network failure and on the probability p_2 that a significant package loss can occur in the intervall from time step 150 to 155.

τ_{min} $[ts]$	35	41	47	53
$p_1 \cdot 10000$	72.8	54.4	39.3	27.1
$p_2 \cdot 10000$	3.71	2.83	2.09	1.47

Table 1. Probability of signal loss.

4 Conclusion and outlook

In analysing the consequences of component failures in networked automation systems it is important to take the nature of the exchanged signals into account. In the presented work three main types of signals are identified and models describing the associated network transfer under consideration of different kinds of failures are proposed. A case study shows the feasability of PMC answering questions on the reliability of distributed network automated systems by avoiding worst case analysis.

Currently, an empirical work is done, analysing a system build from several I/O-Cards, sensors, controllers and switches, normally used in student lab. The different types of delays in the system were measured, depending on different configurations and external Ethernet traffic. These results are used to build a modular model of the system in Prism.

The next aim for the presened analysis is to study the influence existing between the length of the PLC cycle and the reliability of the system. A very short cycle rises the chance to discover a binary signal or receive an important signal within a specified time bound, respectively. However, a short cycle will also rise the network traffic and thereby the probability of packet losses by the network. Even if a TCP/IP based network protocol should not lose any packages, the process of resending a package lost on the physical or data link layer may last longer, than it could be tolerated. To do so, more detailed network models (including the dependence of delays on traffic load) will be built.

References

[1] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability, *Formal Aspects of Computing*, 6, no. 4: 512–535, 1999.

[2] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic symbolic model checker. *Proc. TOOLS02, LNCS*, vol. 2324: 200-204. Springer, 2002.

[3] PRISM-Webside: http://www.cs.bham.ac.uk/ dxp/prism/

[4] M. Kwiatkowska, G. Norman, and D. Parker. Modelling and Verification of Probabilistic Systems, in: Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems. *CRM Monograph Series*, vol 23. American Mathematical Society, 2004.

[5] R. Alur, C. Courcoubetis, and D. Dill. Model-checking for probabilistic real-time systems. *ICALP91, LNCS*, vol 510: 1–100, Springer, 1991.

[6] T. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. What good are digital clocks? *Proc. ICALP'92, LNCS* vol. 623: 545–558, Springer, 1992.

Greifeneder, J. and Frey, G.: *Probabilistic Delay Time Analysis in Networked Automation Systems.* Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2005, Catania, Italy, Vol. 1, pp.1065-1068