
Abouelabbas Ghanaim*, Geovany A. Borges**, Georg Frey*
*Saarland University, Chair of Automation, D-66123, Saarbrücken, Germany
{abouelabbas.ghanaim, georg.frey}@aut.uni-saarland.de
**University of Brasilia, Department of Electrical Engineering, Brasilia – DF, Brazil
gaborges@ene.unb.br

Abstract

This paper presents a novel two-step approach for modeling forward and backward network delays in networked control systems (NCS). The first step is to build a colored Petri net (CPN) structural model for the simulation of Ethernet based networked control systems. The modular model captures the most important features of industrial networked control systems, such as client/server input/output scanning and cyclical execution of the control algorithm on a Programmable Logic Controller (PLC). CPN tools software is used to build and simulate the model with different parameter sets. This first easy to build model is used in a second step to identify a compact phenomenological model from an extensive set of simulation results. In this second step a finite state Markov chain delay model (FSMD) is used to capture the delay patterns generated with the CPN model. The resulting model is more compact than the CPN model and due to its mathematical form it can be integrated in existing design and analysis methods for NCS.

1. Introduction

Networked control systems (NCS) as shown in Figure 1 are a type of distributed control systems where sensors, actuators and controllers are interconnected via communication networks. Several advantages of these systems include: reduced systems wiring, increased system agility and ease of system diagnosis and maintenance. On the other hand, the insertion of the communication network in the feedback control loop makes analysis and design of NCS complex because of the network-induced delays, which occur while exchanging data among devices connected to the shared medium. These delays can degrade the dynamic system performance and are a source of potential instability as discussed for example in [5] and [6].

One of the well known approaches for modeling network delay times in addition to measurement, static analysis, and discrete modeling, is Markov chain modeling. Markov models are quite popular and have been used recently to model performance of communication channels as shown in [3]. Colored Petri nets (CPN), on the other side are high-level Petri nets with graphical form and well-known semantic which make them suitable to describe concurrent and non-deterministic systems as shown in [8].

Figure 1. Basic structure of a networked control system.

In this paper, the two approaches, explicitly CPN and Markov chains are used for modeling network delays. First, the software CPN tools developed by Kurt Jensen at the University of Aarhus in Denmark 1982 is used with its existing simulation and monitoring features to build a hierarchical timed colored Petri net (HCTPN) model for a NCS. The easy to build structure-conserving model explores the most interesting features of NCS. Currently it concentrates on the client/server approach of most real time networks and the interactions with the cyclic execution of control algorithms on standard controller hardware as e.g. PLCs. Second, the delay sequences resulting from the simulation of the CPN model are used to build a compact phenomenological model based on Markov chains to be used in controller design. To this end, the simulation results are divided into two sets. One set is used to train a proposed finite state Markov model scheme. The other delay sequence set is used to validate the model.

The rest of this paper is organized as follows: The next section describes the CPN modeling approach in more detail. Section 3 presents the Markov modeling
approach for CPN-model based delay patterns. Section 4 explains simulation and modeling results. In the concluding section the results are commented and an outlook on future work is given.

2. CPN Modeling and Simulation

Recently, it has been seen in literature an increase of research involving CPN modeling and simulation schemes for networked automation systems (NAS) and communication channels. Some of the published papers have discussed detailed Automation system construction using CPN models which are built based on object oriented concepts as shown in [7] or in a component-based structure-conserving way as shown in [4], [9].

In contrast to these models we propose a reduced structure conserving model to capture the main effective features in networked control systems, especially those using cyclic (time-driven) execution platforms. In the model, the interaction between the IO scanning cycle and the program executing cycles is clearly demonstrated.

The model is composed of three interacting modules called “pages” – all of them further structured in a hierarchical way – to achieve simplicity, clearness in modifying variables and ease of running simulations. Figure 2 shows the top level page of the CPN model for a typical networked control system, in which the controller is a PLC-CPU, the network is an Ethernet and the input/output module is PLC Remote IO which is connected directly to sensors and actuators of the process. The places A and B of type packet in the figure show that the controller commands and process feedback signals are interchanged among the system components in the form of data packets through the network.

![Figure 2. Top level page of the CPN based model for NCS.](image)

Figure 3 shows the controller page of the NCS model, which is the main part of the model. The controller is constructed of two parts. The left side part of the page is the CPU scanner which is dedicated to scanning of the remote IO modules. The scanning is done by sending client request packets from place IO_Scan through the firing of the Send_R transition. The packets are sent as Modbus TCP/IP protocol, i.e. each packet has a data field (input or output), command field (read or write) and an acknowledge field. The scan cycle begins with sending data request packets to the input/output modules and waits until all IO modules reply with acknowledge packets. The new IO scan will start after the scanner receives all the acknowledgment of the sent packets and the specified scan time that appears in the inscription of the Send_R transition has elapsed.

The second part (right side) is an implementation of the PLC CPU program scan cycle which is divided into three time periods; read time, execute time and write time. In the execute phase, the executed program is modeled. In this simple example a unit function is used; i.e. the output is the same as the input.

![Figure 3. Controller model with built-in input/output scanner.](image)

For simplicity, the network part of the model as shown in Figure 4 is modeled as a constant delay in the two directions. The most interactions in the NCS model are in the client IO scanning and in the server IO acknowledge, so the network part is not dedicated for any specific network details.

![Figure 4. Network model page.](image)

The last part of the CPN model is the input/output module as shown in Figure 5. The IO module is built as a place IO_B of type “Data” which stores the integer pairs of input/output data. The two transitions: IO_Read and IO_Write implement the required IO scan cycle time. The IO module contains also a built-in
input/output adapter. The IO_Server transition models the IO adaptor that receives the requests and sends the acknowledge packets. The IO module has a transition called Sensor that is used to send a time-stamped sampled ramp signal with 1ms sampling interval.

The idea is to record the arrival time of the consecutive received data values at the beginning of the CPU execute phase and also at the output port of the IO unit, hence we can accurately compute the network backward and forward delays. The backward delay is the time elapsed between reading a new data value from the sensor (i.e. at the process) and the start of the execution phase of the controller using this data. The forward network delay is the time elapsed between the start of the execution phase of the controller and the arrival of the newly generated output value at the output port of the IO module, i.e. at the process

Figure 5. Input/output module model with built-in input/output adapter.

Recently the problem of finding the probability distribution for the response time i.e. the round trip time elapsed by the data to reach from input port to the output port was addressed as shown in [10], in which the probability distribution is estimated using independent random data measurements without any assumption about the memory property of the system.

In this paper we consider the time performance of the network delays i.e. the backward and forward network delays separately. The time performance means the relationship between consecutive delay samples, and the memory property of the system and that is the motive to use a Markov chain to model the delay patterns.

Table 1 shows the main parameters of the model that can be easily modified to study the effect of different system configurations. Figure 6 shows the result of the simulation of the CPN model for the parameters given in this table with a ramp sampled data with 1ms interval. The figure shows the periodicity and the time dependence (memory property) for the delays.

The model is also tested for different scan cycle times and the results show that in many cases the total response time which can be considered as the sum of the two delay is constant as shown in Figure 7.

Table 1. CPN model parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLC read time</td>
<td>2ms</td>
</tr>
<tr>
<td>PLC execute time</td>
<td>8ms</td>
</tr>
<tr>
<td>PLC write time</td>
<td>2ms</td>
</tr>
<tr>
<td>IO Client scan cycle</td>
<td>17ms</td>
</tr>
<tr>
<td>Packet processing time in IO client</td>
<td>2ms</td>
</tr>
<tr>
<td>Network processing time</td>
<td>2ms</td>
</tr>
<tr>
<td>IO unit read time</td>
<td>1ms</td>
</tr>
<tr>
<td>IO unit write time</td>
<td>1ms</td>
</tr>
</tbody>
</table>

Figure 6. Network delays resulting from CPN model.

Figure 7. Delays for IO scan = 15ms and PLC scan = 10ms.
3. Markov Chain Model

The Markov chain model is a discrete, finite state model that describes a probability distribution over a number of possible sequences [1]. Model states might correspond to network load states that lead to network induced delays or resource sharing delays due to cyclic execution of PLC controllers. Each state is directly observed and the state change is probabilistic with state transition probabilities.

The structure of the regular time-homogenous Markov chain model is defined as

\[
\pi = \begin{pmatrix}
\pi_0 \\
\pi_1 \\
\pi_2 \\
\vdots \\
\pi_N
\end{pmatrix}
\]

\[
P = \begin{pmatrix}
p_{00} & p_{01} & \cdots & p_{0N} \\
p_{10} & p_{11} & \cdots & p_{1N} \\
\vdots & \vdots & \ddots & \vdots \\
p_{N0} & p_{N1} & \cdots & p_{NN}
\end{pmatrix}
\]

where \( \pi = \{\pi_0, \pi_1, \ldots, \pi_N\} \) is a finite set of states, \( P \) is the time-independent matrix of state transition probabilities where \( p_{ij} \) is the transition probability from state \( s_i \) to state \( s_j \) and \( \pi_0 \) is the initial state distribution vector. The property of regular Markov chain means the ability to go from any state to any other state in exactly \( n \) steps which is a very important property to find a stationary (limiting) distribution \( \pi \) for steady state behavior of the model [1]. The stationary distribution can be found from the balance equation:

\[
\pi P = \pi,
\]

which means that, the stationary distribution is the normalized left eigenvector for the transition matrix \( P \) corresponding to unity eigenvalue.

The proposed finite delay Markov model is based on a finite number of delay values with the assumption of perfect measurement which means that the CPN-delay patterns are directly observed without uncertainty.

By defining the Markov model parameters, we now have a concrete analytical description for the network induced delays. This analytical model can be integrated into the state space equation of the closed-loop control system for analysis and design of system controllers as shown in [5].

This analytical description of the network delays is the goal of the second step of the proposed procedure. In other words, the first step uses the CPN tools for its powerful simulation capability to model the NCS system and generate the network delay patterns. Thereafter, the second step uses Markov modeling to overcome the drawbacks of the CPN model which are inability to deal with real numbers and poor analytical analysis description. These limitations of the CPN approach make it not sufficient alone for the controller design process of the networked control systems especially when continuous processes under control are considered.
4. Markov Model Validation Results

The verification of the proposed Markov model is done by estimating the stationary distribution for the two Markov models and comparing these distributions with probability density functions of the CPN generated delay patterns. There are two ways generally to estimate the stationary distribution as shown by Gilks in [2]. The first is analytically by solving the balance equation for eigenvector corresponding to a unity eigenvalue. The second way is by using Markov simulation of the model λ using the initial distribution π₀ and the transition matrix P to obtain large sequences and then calculating the frequencies and estimating the probability distribution. The two methods are used here for estimating the stationary distribution using Matlab statistics toolbox [11]. The transition matrices \( P_\text{for} \) and \( P_\text{bak} \), are solved for normalized eigenvector that corresponds to unity eigenvalue, and the stationary distribution vector \( \pi \) is obtained. The probability density function was estimated using other 10000-sample arrays of CPN consecutive delay sequences which are different from those used for the estimation of the models.

Figure 9 and Figure 10 show the probability density function of CPN generated delay sequences compared with the stationary distribution for Markov models for forward and backward models respectively. The figures show a good capability for Markov chain models to estimate network delay sequences with a good stationary distribution compared to the probability density function of CPN generated sequences.

The second part of validation of the proposed finite state Markov delay models uses the current/next state space graphs as a measure of performance, which means the current/next state spaces are drawn for the CPN estimated delays and that obtained by simulating Markov models.

Figure 11 and Figure 12 show the current/next state space of CPN delay data patterns for forward and backward models respectively.
The delay values appear separated and distinguishable, which validate the assumption of data uncertainty and the use of direct Markov chain models instead of using Hidden Markov Models (HMM) in this case. Hidden markov models can be used for purpose of reducing Markov model states and in case of uncertainty in data, which is left for future work with more details for network and controller parts of the CPN model. In addition, the dashed lines show the current state = next state lines, which show that the current state go to next state(s) as assumed in the ring model assumption shown in Figure 8, which is also clear from the almost diagonal form of the estimated transition matrices.

5. Conclusion and Future Work

The paper presented a novel two step approach for modeling delays in networked control systems. First a structural model is built concentrated on the effect of client/server input/output scanning and program execution cycles for typical NCS cyclic controllers. Secondly a finite state Markov delay model is proposed to identify the delay sequences.

The results of the CPN model simulation show an interesting delay performance for the consecutive transmitted data. The delay sequence patterns show high periodicity in the resulting sequences for different IO/PLC scan cycles combination. This periodicity in delay patterns was a motivation to address the memory property of such delays through Markov modeling.

The results of the finite state Markov delay models show that the models are satisfactorily capable to capture the main system properties and to estimate a good stationary delay distribution for a certain IO/PLC scan cycle combination. This two step procedure can be used in a very simple and fast way to study the effect of different system parameter variations in Table 1, thanks to the fast simulation time of CPN models and ease modifying of CPN model parameters.

In future work, a more detailed CPN model will be constructed to cover network details and controller program execution details. In addition, a real delay data patterns for a typical case study will be compared with CPN/Markov model results, also the Markov model will be modified to a hidden Markov model to accommodate with uncertainty of the real data and to reduce the number of model states.

References