

Markov Chain-based Performance Evaluation of FlexRay Dynamic Segment

Jimmy Jessen Nielsen Hans-Peter Schwefel
Aalborg University, Department of Electronic Systems
Fredrik Bajers Vej 7, DK-9220, Aalborg
{jjni03, hps}@kom.aau.dk

Amen Hamdan
BMW Car IT GmbH
Petuelring 116, D-80809, München
Amen.Hamdan@bmw-carit.de

Abstract

With the launch of the new BMW X5, the FlexRay protocol for in-car communication has found its way to series-produced vehicles. The FlexRay protocol supports both deterministic and non-deterministic transmission of data frames. FlexRay data frames that are being transmitted in the non-deterministic dynamic segment might become displaced under adverse circumstances. In the design of a FlexRay network it is important to have sound understanding of any implications of certain design decisions on the performance of the overall network, here specifically the displacements in the dynamic segment.

This paper proposes a novel approach to performance analysis of the dynamic segment based on Markov chain transient analysis. The model of the dynamic segment is a two-dimensional discrete-time Markov chain, where the discrete time steps represent minislots of the dynamic segment. Model properties and assumptions are discussed and expressions for calculating performance metrics are provided. Finally, measurements obtained from a real FlexRay network are used to test model assumptions and to validate the accuracy of model output.

Keywords - flexray, in-car networks, dynamic segment, markov chain, transient analysis, performance analysis

1 Introduction

The increasing number of x-by-wire applications in cars entails that requirements on safety and hard real-time of the in-car network are becoming more strict [4]. Protocols such as Byteflight and FlexRay, where FlexRay is a recent extension of Byteflight, have been developed and used for satisfying the requirements of x-by-wire applications. The first series car to use a FlexRay in-car network is the new BMW X5, which has recently been introduced by BMW Group. In near future FlexRay is expected to connect multiple Electronic Control Units (ECUs) implementing chassis, powertrain, and driver assistance applications [6].

Besides providing strict determinism, the FlexRay protocol supports priority based media access in the dynamic segment via a Flexible Time Division Multiple Access (FTDMA) scheme. The properties of the FTDMA scheme is discussed in [2]. The flexibility provided by this media access scheme allows frames to be displaced and thus delayed in adverse situations. This dynamism combined with the increasing number of applications motivates the need for quantitative performance evaluation of the network.

The Byteflight protocol is nearly identical in functionality to the FlexRay dynamic segment as it is also based on FTDMA. The authors in [3] present an analysis of the Byteflight protocol, where they investigate the performance-related consequences of different design choices. Particularly, they identify that network designers need to make a trade-off between flexibility and performance. This issue applies similarly to the FlexRay dynamic segment, and is discussed further in Section 2. Currently available tools for performance evaluation of specific FlexRay traffic models focus on simulation. Simulation tools such as [8] and [7] may be used to make a quantitative performance analysis, by defining the traffic model using virtual traffic generators. However, the process of obtaining a sufficient level of confidence with a simulation is typically very time-consuming compared to analytical approaches.

This paper presents an approach to performance assessment of the dynamic segment in FlexRay that is based on transient analysis of a Markov Chain (MC). This performance assessment is less time-consuming than simulation and can be made prior to implementing a simulation or prototype.

Specifically, this paper contributes to the literature by 1) demonstrating how the FlexRay communication cycle can be modeled as a MC, 2) specifying how performance metrics are extracted from MC model, and 3) validating the results obtained from the model by comparing to measurements from an actual FlexRay network.

In Section 2 a brief description of the relevant FlexRay properties is given and the main motivation for the model is presented. Next, the FlexRay model is discussed and pre-

sented in Section 3. The following Section 4 specifies how performance metrics are calculated from the model. Section 5 presents a validation of the model based on measurements from an actual FlexRay network. Finally, Section 6 contains the conclusion of this paper.

2 FlexRay Dynamic Segment

The following gives a brief introduction to FlexRay and especially the dynamic segment, which is the focus of this paper. More information on FlexRay is available in [5].

In FlexRay the communication cycle is the fundamental element of the media access scheme. The communication cycle is divided into four segments as depicted in Fig. 1.

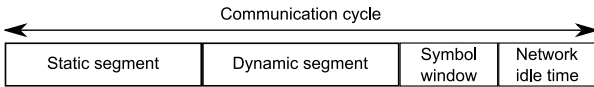


Figure 1. The FlexRay communication cycle.

Within a communication cycle, ECUs may transmit frames in the static and dynamic segments. At design-time an ECU has been assigned one or more slot IDs in which it may transmit. Each slot ID is only used by one ECU, and in this way collisions will never occur. In the *static segment*, where a Time Division Multiple Access (TDMA) scheme is used, the transmission of frames is completely deterministic. The *dynamic segment* uses a dynamic mini-slotting scheme, also generally referred to as FTDMA. This scheme is not deterministic since the offset from the beginning of a communication cycle until an ECU may transmit, varies with the number of frames already sent in the same communication cycle. Fig. 2 exemplifies the concept of this scheme.

An unused dynamic slot has the duration of one minislot, which is the common time unit in a FlexRay network, and the length of the communication cycle is a fixed number of minislots. If a dynamic slot is used for transmitting a frame, the duration of the dynamic slot is extended to several minislots, depending on the payload size of the frame. This is exemplified for dynamic slots $m + 3$ and $m + 6$ in the figure. Every ECU maintains a local counter of both the current minislot ID and the current dynamic slot ID. Both are reset at the beginning of each cycle.

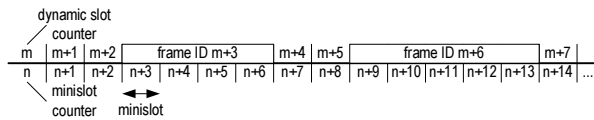


Figure 2. Dynamic minislotting scheme.

Since the payload length and the number of frames trans-

mitted in each communication cycle varies, it may occur that a frame cannot be successfully transmitted within the available minislots. In that case the frame will not be transmitted in the current communication cycle. Instead the frame is displaced to a later communication cycle with enough available minislots. More specifically, a frame is displaced if the minislot counter exceeds the threshold $pLatestTx$ that is configured in each FlexRay node. $pLatestTx$ is the last minislot in which the node can successfully transmit a frame with the maximum allowed frame size.

Displacements may occur even with a low number of used dynamic slots. If for example the communication cycle is configured for a length of 250 minislots, and a frame has a frame ID of 230 and takes up 5 minislots to transmit, no more than 15 earlier minislots may be included in frame transmissions in a communication cycle, before the frame cannot be transmitted and is displaced.

In order to reduce the number of displacements, the FlexRay network designer should aim at using as low frame IDs as possible. However, this is typically inappropriate with respect to compatibility and upgradability. The designer therefore needs to determine a configuration of the network that provides an acceptable trade-off between flexibility and performance. Besides displacements, other aspects of performance could be relevant to consider. In this paper the last dynamic slot is also calculated.

The modeling approach proposed in this paper provides the network designer with a tool for making quantitative performance evaluation of different network configurations and traffic models in order to rapidly iterate over the design of the network.

3 FlexRay Model

In this section the approach to modeling the dynamic segment is described in detail. The performance metrics described in the following are the desired outputs from the model.

Last Dynamic Slot (LDS) distribution: The LDS is the value of the dynamic slot counter at the end of a communication cycle. The more minislots that have been used for transmitting frames, the lower the value of the LDS. The network designer may use the LDS as an indicator of the level of utilization and as an indication of which frames could not have been transmitted in the communication cycle. If the LDS has a lower slot ID than the ID used by a certain ECU, that ECU could have had a frame scheduled for transmission that it was not allowed to transmit in the corresponding communication cycle.

Frame displacement probability: The objective of this metric is to quantify the risk of frame displacements for a

specific frame ID. The network designer may use this metric to determine the quality of service in relation to displacements that the network delivers for a given configuration.

3.1 Model Framework

The following description of the FlexRay model assumes a basic understanding of transient analysis of discrete-time MCs, see e.g. [1]. In this paper the transition probability matrix is denoted by \mathbf{P} and the state probability vector for time k by $\pi(k)$. The state probabilities are calculated using the standard equation

$$\pi(k) = \pi(0)\mathbf{P}^k \quad (1)$$

The main features of the FlexRay model are described in the following.

Time step: Each time step in the MC has the duration of exactly one minislot. After k time steps, the state probability vector $\pi(k)$ describes the possible outcomes of the communication cycle via state probabilities. An example of information that could be derived from $\pi(k)$ is the probability of the frame with ID n having been transmitted after exactly k time steps.

Idle and transmission states: Each state in the MC constitutes either an idle minislot, or a minislot that is used for transmitting a frame. Typically, several consecutive minislots are involved in the transmission of a single frame. In the MC these consecutive minislots are mapped into a sequence of connected states that accurately represent the length of the frame transmission.

Frame ID priorities: The prioritization of frames via dynamic mini-slotting is included in the model by defining the transition probabilities so that states in the MC are visited in the order specified by their corresponding slot IDs.

Simplifying assumptions: The following simplifying assumptions are made for the FlexRay model:

- For each frame ID a a constant frame length l_a is used. The FlexRay specification allows a node to use a variable frame size in the dynamic segment. However, the model assumes that frames within a frame ID use the same payload length.
- For each frame ID a a constant arrival probability p_a is used. Further, frame arrival probabilities are assumed to be independent and inter-arrival times to be geometrically distributed.
- The pLatestTx check is not integrated in the model structure. This is discussed further in Section 4.

The validity of these simplifying assumptions is discussed further in Section 5.

3.2 Detailed Model Definition

The model is based on a non-homogenous MC using a two-dimensional state space (a, b) . Here, a equals the ID of the dynamic slot relative to the beginning of the dynamic segment, i.e. the first dynamic slot should have $a = 1$. The value range of a is $[0; s_{max}]$ (where s_{max} is short notation for the Flexray specific parameter $gNumberOfMinislots$). b relates to the states within a dynamic slot. $b = 1$ designates an idle state and $b > 1$ designates transmission states. The value range of b is $[1; l_{max} + 1]$, where l_{max} is the maximum allowed length of a frame measured in minislots.

The initial state of the MC is $(0, 1)$.

The transition probabilities of the MC correspond to the arrival probabilities of each dynamic slot.

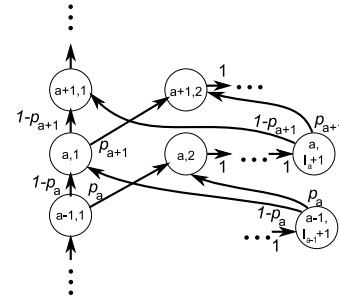


Figure 3. FlexRay MC structure

The structure of the FlexRay MC is shown in Fig. 3. For a dynamic slot a , there is an idle state with index $(a, 1)$ and l_a transmission states from $(a, 2)$ to $(a, l_a + 1)$. l_a is the transmission length in minislots of the frame in dynamic slot a . The arrival probabilities for dynamic slot a is p_a . If a dynamic slot is unused, the arrival probability becomes $p_a = 0$, i.e. there are no transmission states.

The frame length l_a that is needed to create the MC, is calculated from the payload length. However, this calculation is not completely trivial, since it depends on the low-level parameters of the concerned FlexRay network. This calculation may be found in [5, Appendix B4.14].

In order to calculate the state probability vector π , the transition probability matrix \mathbf{P} needs to be generated from the traffic model. One possible approach to generating \mathbf{P} is to iterate over the set of dynamic slots, and with a starting point in the associated idle state, consider the state transitions initiated in this state. For each slot s one of the following three situations apply:

- a) No ECUs are assigned to slot s and only a transition to the idle state in slot $s + 1$ is possible.

- b) Frame transmission is possible in slot $s + 1$, but not in slot $s + 2$. Transitions to idle and transmission states in slot $s + 1$ and to the idle state in slot $s + 2$ should be created.
- c) Frame transmission is possible in slot $s + 1$ and in slot $s + 2$. Transitions to idle and transmission states in slot $s + 1$ and to the idle and transmission states in slot $s + 2$ should be created.

Having outlined the procedure for specifying the transition probability matrix \mathbf{P} for the FlexRay MC, only the initial state probability vector $\pi(0)$ is left to be specified. With the initial state of $(0, 1)$, the state probability vector should be set as

$$\pi(0) = [1, 0, \dots, 0] \quad (2)$$

4 Calculating Performance Metrics

The calculation of the performance metrics consists of two steps. First, the state probabilities $\pi(k)$ are calculated, and secondly the performance metrics *frame displacement probability* and *LDS distribution* are computed from $\pi(k)$. Formulas for computing these metrics are presented in subsections 4.2 and 4.3. However, first the simplifying assumption that the *pLatestTx* check is not included in the MC is discussed.

4.1 pLatestTx Check

In the MC, frame transmission are being initiated even if they should not, due to the *pLatestTx* check not being performed. Thus, in the MC, the transmission states before the last transmission state may contain probability mass at the end of the communication cycle, even though this would not occur in an actual network. The start of the arrows in Fig. 4, show which states contain excess probability mass, while the arrow ends show where the probability mass should reside, had the *pLatestTx* check been carried out.

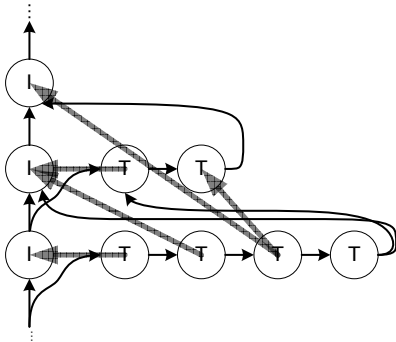


Figure 4. *pLatestTx* issue.

However, with relation to frame displacements, the missing *pLatestTx* check allows a more accurate calculation of the frame displacement probability, and the *pLatestTx* check should therefore purposely be left out of this calculation. On the contrary, the calculation of the LDS needs to have the MC reflect the behavior of an actual FlexRay network. Subsection 4.3 describes possible approaches for bringing the *pLatestTx* check into the calculation of the state probabilities when calculating the LDS.

Now the calculation of the performance metrics is described.

4.2 Frame Displacement Probability

In order to determine if any frames have been displaced in a communication cycle, the outcome of the MC must be calculated for the time $k = s_{max}$, i.e. when the communication cycle has finished. This is obtained using Eq. (1). The displacement probability for a dynamic slot is given by all state probabilities existing before the last transmission state of slot s . The state probabilities before slot s should be conditioned on the transmission of a frame in s , hence the sum of the state probabilities is multiplied by the arrival probability p_s . The frame displacement probability for frames transmitted in dynamic slot s can be obtained from

$$P(s \text{ is displ.}) = \sum_{b=2}^{l_s} \pi(k)_{s,b} + p_s \cdot \sum_{a=1}^{s-1} \left(\sum_{b=1}^{l_{max}+1} \pi(k)_{a,b} \right) \quad (3)$$

4.3 Last Dynamic Slot

The LDS is the value of the dynamic slot counter at the end of a communication cycle, i.e. for time $k = s_{max}$. Since the last dynamic slot in a communication cycle in an actual network may have been used either for finishing the transmission of a frame or being idle, the LDS for slot s may be calculated via

$$\mathbf{S}_s = \pi(k)_{s,1} + \pi(k)_{s,l_s+1} \quad (4)$$

However, as discussed in Subsection 4.1, the *pLatestTx* is not performed when calculating $\pi(k)$. Eq. (4) assumes that the *pLatestTx* check has been performed during the calculation of $\pi(k)$. There are several ways to include the *pLatestTx* check. In the following two exact approaches, and one approximate approach are described.

The first approach is to use multiple transition probability matrices that prevent transitions to the transmission states when the *pLatestTx* values of the concerned slots have been reached. The second approach is to perform a post-processing operation that propagates the residual probability mass according to the transition probabilities as exemplified in Fig. 4. After applying either of these two approaches, Eq. (4) may be used to calculate the LDS. The

third approach is a simplified post-processing that adds the residual probability mass to the relevant idle states only. This approach does not consider cases where probability mass needs to be propagated to transmission states and is therefore only an approximation in such cases. However, it is simpler to implement than the two other solutions and may provide the exact solution if the used network configuration and traffic model does not contain such cases. The probability of slot s being the LDS is calculated via

$$\mathbf{S}_s \approx \pi(k)_{s,l_s+1} + \sum_{a=0}^{l_{max}} \pi(k)_{s-a,a+1} \cdot I(a \leq l_{s-a}) \quad (5)$$

Here, the indicator function $I(a \leq l_{s-a})$ evaluates to 0 when the expression is false and 1 when it is true. The plots of the LDS distribution that are included in Section 5, have been calculated using this approximate approach. Regardless of which approach is used to calculate the LDS probability for each of the s_{max} dynamic slots, the complete distribution of last dynamic slots is given as

$$\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_{s_{max}}] \quad (6)$$

5 Model Validation

The validation of the FlexRay MC model is divided into two steps. The first step is to investigate how well the assumptions on traffic properties match the actual network traffic. The second step is a comparison of results calculated using the MC and results obtained from measurements on a real FlexRay network.

5.1 Assumptions

The first assumption is that frames with the same frame ID use the same payload length. The test data was found to comply to this assumption, which should not cause any inaccuracies in the model prediction. However, since a fixed payload length is application-dependent, this assumption may limit the applicability of the model, or at least provide inaccurate results in cases where dynamic frame sizes are used. Note however, that the Markov Chain model can also be extended to arbitrary frame size distributions via a modification of the 'upward' transitions in 4.

Further, assumptions were made that frame arrivals are independent. This covers both independence between frames with identical frame IDs, but also between frames not sharing frame IDs. These assumptions were tested using correlation analysis. In order to test the assumptions, a binary arrival sequence was created for each frame ID in the measurements. That is, the communication cycles in which a frame with the given frame ID has arrived are represented

by a 1 in this sequence, while lack of arrival is a 0 in the sequence.

The autocorrelation was used to test the correlation of frames with identical frame IDs. Autocorrelation plots have been created for all frame arrival sequences. The frame arrivals were found to be highly correlated and showing a large degree of periodicity. This contradicts the assumption of independence between frame arrivals within each frame ID, and is expected to introduce some level of inaccuracy to the results of the model, compared to an actual network.

Finally, the correlation for frames with different frame IDs was investigated by computing the cross-correlation coefficient between all pairs of frame arrival sequences. The result is depicted in Fig. 5, where the darkness of each square expresses the degree of correlation between a pair of arbitrarily indexed arrival sequences. The results show that besides the auto-correlations along the diagonal, most frame arrivals were not or only weakly correlated. This result supports the assumption of independence between different frame IDs. However, it could be relevant to investigate the cross-correlation for other lags than 0.

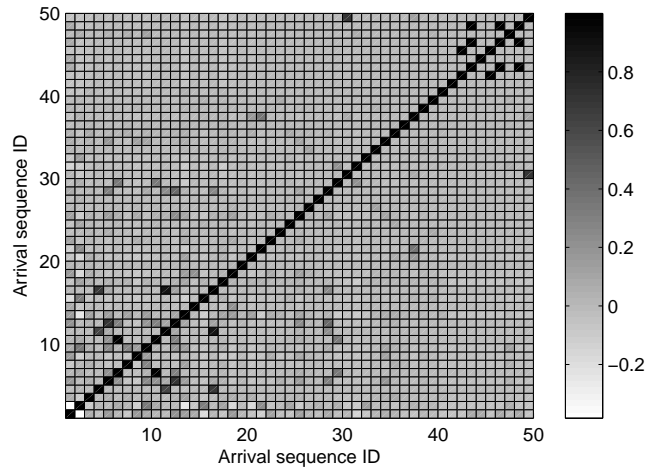


Figure 5. Cross-correlation plot of frame arrival sequences for lag 0.

5.2 Model Output Comparison

The following presents two comparison plots, Fig. 6 and Fig. 7 that have been selected from a larger result set. The plots are based on measurement and model results for two traffic models that differ in the assigned frame IDs and arrival rates. The plots in the figures show the Cumulative Distribution Function (CDF) of LDS. Here, the horizontal distance between the lines is interesting since it shows how accurately the model mimics the behavior of the actual system. The results presented here show the model predictions

from the result set that have the least horizontal difference (best) and the largest difference (worst).

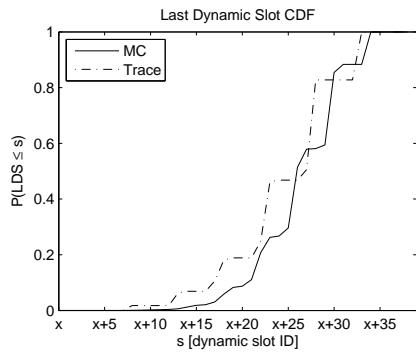


Figure 6. Best estimate of LDS distribution.

The plots are indexed relatively from x , since the specific technical details are not of interest here. The result in Fig. 6 shows a high degree of resemblance between the results of the model and the trace. The horizontal difference seems to be limited to 5 minislots and within 2-3 minislots for the most part.

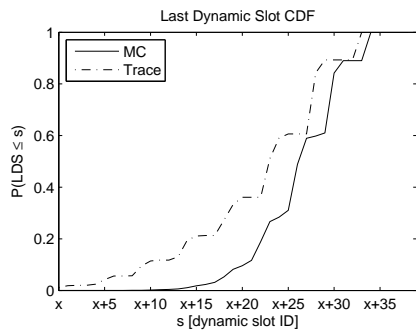


Figure 7. Worst estimate of LDS distribution.

The results in Fig. 7 show a lower degree of similarity, where the horizontal difference is as high as 10 minislots. Common for both plots is a tendency of the model towards a more optimistic result, which is caused by the simplifying assumption regarding independence of frame arrivals being inconsistent with the actual network. However, these are only minor issues that will not hinder the network designer in making qualified decision on the design of the network.

6 Conclusion and Further Work

A Markov chain based model for quantitatively analyzing the performance of the FlexRay dynamic segment has been presented as a tool for the network designer to make early predictions on network behavior. Results are obtained via transient analysis of this two-dimensional Markov chain.

Expressions for calculating the performance metrics *distribution of Last Dynamic Slot* and *frame displacement probability* have been discussed and presented.

The model is based on assumptions regarding mutually independent frame arrivals and constant payload lengths within each frame ID. This allows for the traffic within each frame ID to be described by just three constant parameters *frame ID*, *frame length* and *arrival probability*. A validation of model output against traces obtained from a real FlexRay network shows that the accuracy of the prediction of the distribution of LDS is within 5 minislots in one case and within 10 minislots in another. This level of accuracy appears sufficient for a network designer to make qualified decisions in the early phases of network design.

Further work could add additional performance metrics such as jitter, which is caused by displacements and variations in the number of minislots that a dynamic slot is offset from the beginning of the dynamic segment. Another topic could be to validate the model more thoroughly with a wider selection of network configurations and traffic models. Further it would be interesting to use the model for designing a network or predicting the performance of future network configurations and traffic models via extrapolation. Finally, it would be interesting to extend the model to allow the effect of errors on performance to be investigated.

The research work was in parts conducted in the context of a student project at the Department of Electronic Systems at Aalborg University, and as such supported by the Study Board at Aalborg University, see www.esn.aau.dk.

References

- [1] C. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Kluwer Academic Boston, 1999.
- [2] G. Cena and A. Valenzano. On the properties of the flexible time division multiple access technique. *IEEE Transactions on Industrial Informatics*, 2(2):86–94, 2006.
- [3] G. Cena, A. Valenzano, C. IEIIT, and I. Torino. Performance analysis of Byteflight networks. *IEEE International Workshop on Factory Communication Systems. Proceedings.*, pages 157–166, 2004.
- [4] G. Cena, A. Valenzano, and S. Vitturi. Advances in automotive digital communications. *Computer Standards & Interfaces*, 27(6):665–678, 2005.
- [5] F. Consortium. *FlexRay Communications System Protocol Specification Version 2.1 Revision A*. FlexRay Consortium, 2005.
- [6] W. Kuffner, G. Reichart, J. Berwanger, M. Peteratzinger, and A. Schedl. FlexRay Exploitation of a Standard and Future Prospects. In *Convergence 2006, Detroit, MI, USA, Session: International Standards*. SAE International, 2006.
- [7] Mirabilis Design. Visualsim. www.mirabilisdesign.com.
- [8] Vector. CANoe.FlexRay. www.vector-worldwide.com.