A System-Level Protocol-Based Methodology for Noise Analysis of Digital Networked Systems

Giovanni. B. Vece, Eros Mazza, and Massimo Conti
Università Politecnica delle Marche, Dipartimento di Ingegneria dell’Informazione, Ancona, Italy
Email: m.conti@univpm.it

Abstract—Reliability and noise tolerance represent important requirements for electronic systems and in particular for networked systems, especially in critical working conditions. Reliability and noise tolerance mostly concern the communication tasks between the network nodes, which are implemented complying with formal protocol rules. The challenging target for a reliability analysis is to provide comprehensive evaluations and acceptably accurate results. However, the current complexity of the protocol and the complexity of the networks with many nodes make it very difficult to achieve this target.

This paper presents a novel system-level methodology for noise analysis of digital networked systems. The methodology is based on a simulation and analytical approach capable to address a fast and comprehensive study of the reliability properties complying with the protocol specifications. As a test case, the MOST 150 protocol has been considered. MOST 150 protocol is currently widely used for multimedia networks in automotive applications.

Index Terms—System-Level Noise Analysis; Simulation/Analytical Approach; Digital Networked Systems; MOST Protocol; SystemC Language

I. INTRODUCTION

The last decades have seen a continuous progress of digital networked systems, with the development of more and more advanced wireless and cabled communication protocols. Noise effects may have a relevant impact on the performances of these systems, especially in critical working conditions. A typical case is represented by the automotive applications, which are often characterized by stress-prone environments and several interference sources. In such applications reliability is a primary requirement, emphasizing the importance of analysis techniques for studying the noise effects on network communications. However, the complexity reached by modern networked systems makes it difficult to perform this kind of analysis. In particular, it is hard to achieve both comprehensive evaluations and accurate results. Furthermore, the capabilities of many techniques allow the analysis only of a restricted number of working scenarios.

It is convenient to adopt a system-level approach to study the performances of a complex networked system and, in particular, the reliability properties. A system-level approach describes and studies a behavioral network model that represents the overall system functionality without focusing on low-level details [1], [2]. As will be discussed in the next section, two main options can be considered for a system-level representation, i.e. analytical performance models and virtual system prototypes.

An effective way to realize a system-level analysis is to focus on the protocol rules defining the network communication tasks. In fact, this solution allows the study of the system properties on the basis of the core functional specifications, without references to specific implementation details. Furthermore, it would be possible to achieve analysis results of general validity, applicable to all the networks compliant with the examined protocol.

This paper presents a novel system-level methodology for noise analysis of networked systems. More specifically, we have defined an approach that combines the potentialities of fault-injection techniques on virtual prototypes and analytical models, considering only the information provided by the protocol specifications.

The content of the paper is organized as follows. Section 2 introduces the main fault-injection techniques used in digital communications, with some references to their application in CAN and Bluetooth networks. Sections 3 and 4 illustrate the procedural flow of the proposed methodology and some relevant conceptual aspects. A concrete case study on MOST networks is dealt with in Sections 5 and 6. Section 7 reports an experimental verification of the analytical relationships determined for the MOST protocol. The appendix reports the derivation of the analytical relationships.

II. RELATED WORKS

Noise analysis on digital networked systems is mostly based on fault injection techniques [3]. We can distinguish two ways to perform fault injection, depending on whether we deal with a behavioral model or a physical network. These alternatives present complementary features and can be applied in different phases within the design of a digital communication system.

System-level noise analysis is typically related to fault injection in behavioral models, because of the better capabilities to monitor the overall system functionality. A behavioral model can consist in either an analytical performance model or a virtual system prototype. The latter option is based on the use of an executable
description language and it is often more suitable to reproduce the real system behavior. More precisely, a virtual prototype allows the description of system features that could not be modeled with the same precision and flexibility through other representation modalities. The main limitation of this option is usually due to excessive simulation times, especially if the used language poorly supports high-abstraction constructs. Furthermore, in comparison with an analytical model, the analysis conducted using a virtual prototype provides reduced possibilities to infer performance results beyond the simulated scenarios.

Focusing on automotive applications, noise analysis on networked systems mainly concerns networks based on the CAN protocol [4] – [10]. Fault injection techniques on physical networks are described in [4] – [6], by proposing testing devices capable to introduce controllable errors in the data stream of a CAN network. A system-level analysis based on virtual prototyping is illustrated in [7], where a Matlab/Simulink model of CAN network is instrumented for programmable fault injection simulations. All these contributions are essentially aimed to study the network reliability while executing particular instructions.

Examples of system-level analysis based on analytical evaluations can be found in [8] – [10]. In these works analytical models of the network activities are defined including typical noise sources. The study of such models allows to evaluate some performance measures and to derive optimization guidelines for contrasting the considered noise sources.

Other relevant examples of noise analysis based on virtual prototyping are illustrated in [11] - [14], in which some tasks of the Bluetooth and ZigBee protocols are examined in a noisy channel. These analysis are realized through statistical fault injection simulations, in which the BER impact on the examined tasks is evaluated in specific network architectures. In compliance with [15], these two contributions highlight the good capabilities of the SystemC language in the modeling and simulation of a complex networked system.

In our researches on system-level noise analysis, we have studied how to merge the potentialities of the techniques based on virtual prototyping and analytical models. With this intent, we have developed a simulation/analytical methodology in which the analysis requirements depends only on the protocol specifications that define the communication tasks. The simulation part consists in the study of particular noise configurations by means of simulations conducted on a virtual prototype. The analytical part extends the simulation results for estimating the performances of arbitrary noise configurations. As main benefit, this approach makes possible an exhaustive study of the reliability properties of complex networks with reduced simulation time and with a good accuracy level. These capabilities have been verified through an application on networked systems based on the MOST 150 protocol.

A first presentation of our methodology was presented in [16]. This paper provides a better and more accurate exposition of the proposed approach, with a more in depth discussion of the procedural steps and the conceptual aspects. Moreover, the validation of the methodology has been enforced with further experimental results.

III. DESCRIPTION OF THE METHODOLOGY

Noise analysis on a virtual network prototype is typically based on the two-step flow of Fig. 1. In the first step, a noise model must be defined for each component of the network (noise characterization). The second step requires the integration of the noise models into a virtual prototype of the network and then the simulation of this prototype in specific working scenarios (system simulation). In this latter phase it is also necessary to fix two relevant analysis aspects, i.e. the monitored activities and the performance measures. The monitored activities should be the network tasks more critical for system reliability, whereas the performance measures are the indicators used to quantify the reliability level. Monitored activities and performance measures are strictly dependent on the communication protocol and can be fixed on the basis of the protocol specifications.

The methodology discussed in this paper has been developed from this operative framework. With respect to noise characterization, we have considered modeled with bit error rate (BER) the noise contribution of each network node (intended as the associated device and the adjacent communication channel). This choice is suitable for a noise characterization compliant with the scope of a system-level analysis, not depending on the specific system implementation details. Regardless of the intrinsic features of a node, the BER can represent the joint contributions of all the noise sources in the node. In this way, we can handle each node as a black box with regard to noise sources, considering only the resultant effects. Furthermore, the use of BER as noise measure has been widely studied and several methods are available for its estimation [17], [18].

The system simulation phase requires, preliminary, to implement a virtual prototype of the network and to establish the monitored activities and the performance measures. In our approach, these points are realized by exploiting only the information achievable from the protocol specifications, which are usually consistent enough to implement a system-level prototype of the network as well as to determine the activities more critical for system reliability.

On the basis of a BER noise characterization, a networked system with N nodes can be associated to a BER distribution $D(BER_1, BER_2, ... BER_N)$. Each BER element is associated to a distinct network node and it can assume arbitrary values in a continuous range. In general, in the study of a performance measure ($P$) related to noise effects, we should expect a dependence on the single BERs:

$$P(BER_1, BER_2, ... BER_N)$$

(1)
In a digital communication protocol, the network tasks are mapped at bit level on the frame format specific of the protocol. Consequently, the study the BER impact on the frames transmitted in the network communications is a proper way to evaluate relationship (1) in a protocol-level noise analysis. More specifically, it is necessary to focus on those frame fields connected to the monitored activities.

Unfortunately, it is often impossible to derive a precise explicit expression of (1) and accurate evaluations can be achieved only in experimental way, by means of simulations on a virtual prototype of the network. In order to properly define the BER distributions to be simulated, we can consider only some discrete BER values for each node. These values could be fixed through a sampling process based on a constant step, which should be kept adequately low to guarantee a strict proximity of the examined BER distributions. Finally, at the end of the simulations, the achieved results could be stored in lookup table. By means of interpolation, these data could be used to estimate the performances also for BER distributions not examined in the simulation phase.

This modus operandi can be used for a network with few nodes and allows to study (1) in comprehensive way. In fact, in this case it would be feasible to simulate all the possible BER distributions, i.e. all the possible combinations of the discrete BER values of the nodes. However, if the network has tens or hundreds of nodes, as happens in many real applications, an exhaustive analysis would require a huge amount of simulations and prohibitive execution time efforts. Consequently, for medium and large size networks, a system simulation covered only a restricted subset of the possible BER distributions.

Taking into account these limitations, in our methodology the simulation part is carried out only for uniform BER distributions, in which all the nodes are associated to the same BER. This solution allows the representation of the noise contributions in the network only through two variables, i.e. the number of nodes (N) and the associated BER. In this way, we can evaluate a functional dependence much simpler than (1):

\[ P(N, BER) \]  

In other words, the system simulation is restricted to particular cases (reference configurations), as reported in Fig. 2, in which a univocal association is determined between a (N, BER) couple and the P value. Under this simplification, it is possible to address an exhaustive study of the performance measure also for medium and large-size networks.

A noise analysis restricted to uniform BER distributions could be meaningful enough and it could provide useful indications. Furthermore, we have considered how to exploit the results achieved from the reference configurations to evaluate a more general case. This possibility is addressed by the analytical part of our methodology, which is aimed to define a transition formula \( F \) that establishes a connection between uniform and arbitrary BER distributions:

\[ BER = F(BER_1, BER_2, ..., BER_n) \]  

The procedure for achieving (3) requires, preliminarily, the definition of an analytical noise expression that can be referred to as equivalent network BER (NET_BER). This entity should represent a measure of the noise on the network activities as a function of the primary BERs.

The definition of a NET_BER expression is not based on a fixed procedure and could be obtained in several ways. A simple solution may consist of a linear combination of the primary BERs, in which the weight coefficients could be proportional to the average traffic loads of the related nodes. As other possibility, a NET_BER expression can be obtained by evaluating how the primary BERs are propagated in the transmission of a data stream. This latter solution should guarantee a higher accuracy, because it is more capable of matching the communication modalities specified by the protocol.

Figure 1. System-level noise analysis on networked systems by applying a virtual prototyping approach

Figure 2. Original network and reference configuration
Accordingly, in our analysis on MOST-based networks we have followed a BER-propagation procedure to define a \( \text{NET}_\text{BER} \) expression.

The following equation establishes an analytical equivalence between uniform and arbitrary noise distributions:

\[
\text{NET}_\text{BER}(\text{BER}_1, \text{BER}_2, ..., \text{BER}_n) = \text{NET}_\text{BER}(\text{BER}, \text{BER}, ..., \text{BER})
\]

At this point, a transition formula between uniform and arbitrary BER distributions can be obtained by solving (4) with respect to the BER term, in explicit way or through a numerical method.

In the case of a MOST network with \( N \) nodes, we have determined, as reported in appendix, the following transition formula:

\[
\text{BER} = \frac{N^2 - 4N(N-1)\left[ \sum_{i=1}^{N} \text{BER} - 2\sum_{i=1}^{N} \sum_{j=1}^{i-1} \text{BER}_i \text{BER}_j \right]}{2N(N-1)}
\]

This formula has been achieved considering the statistical independence of the primary BERs, which is an assumption appropriate for the MOST protocol, as discussed in appendix. Nonetheless, such assumption does not represent a mandatory constraint in our methodology. In general, the analytical part could be developed considering also a statistical correlation between the primary BERs.

Given a \( D(\text{BER}_i, \text{BER}_2, ..., \text{BER}_n) \) distribution, a corresponding BER value is obtained using (5) only if the following condition is true:

\[
N^2 - 4N(N-1)\left[ \sum_{i=1}^{N} \text{BER} - 2\sum_{i=1}^{N} \sum_{j=1}^{i-1} \text{BER}_i \text{BER}_j \right] \geq 0
\]

This condition is not verified if the primary BERs are extremely high and violate some constraints introduced to obtain (5). However, relationship (6) is guaranteed if the BER distribution is compliant with the validity ranges for the lock condition, as discussed in Section VI.

\[
P(\text{BER}_1, \text{BER}_2, ..., \text{BER}_n) = P(N, \text{BER})
\]

The target result is thus obtained through fast computations, without requiring an ad-hoc simulation. In this way, the analysis flow of Fig. 1 can be considerably simplified to the study of a performance measure related to noise effects. In fact, the system simulation can be carried out, once and for all, only for the reference configuration. Subsequently, these simulation results can be reused to estimate the performances for arbitrary noise distributions, by applying the transition formula (5).

Figure 3 reports the flow of the proposed methodology. All the information necessary to perform a noise analysis are derived from the protocol specifications. The simulation phase also requires an executable language suitable to describe the communication tasks both at functional and bit level.

IV. DETAILED ANALYSIS OF \( \text{NET}_\text{BER} \) COMPUTATION

This section deals with a more detailed discussion of the \( \text{NET}_\text{BER} \) expression and the modalities followed for its computation through a BER propagation approach. For this purpose, we can consider a reference application on a network structure consisting in a chain of \( N \) adjacent nodes (Fig. 4). This structure constitutes the core architecture of a ring network and also represents the basic constituent for several network topologies. Each network node can be associated to an intrinsic BER; in the most general case, we can assume there is not statistical independence on the error occurrence between contiguous node.

![Figure 4. Reference network structure: a chain of N adjacent nodes.](image)

We compute the \( \text{NET}_\text{BER} \) expression of this reference network, following a BER propagation approach. For this purpose, let us define the following probabilities for the network path with the first two nodes, which correspond to the possible error events on a bit transmitted across these nodes:

- \( P_{\text{EO}} \): probability of transitions without errors (the bit retains its correct value at the end of the path).
- \( P_{\text{EO}} \): probability of an error occurred only in the first node.
- \( P_{\text{EC}} \): probability of an error occurred only in the second node.
- \( P_{\text{EC}} \): probability of an error occurred in both the first and the second node.

The situations related to \( P_{\text{EO}} \) and \( P_{\text{EX}} \) can be interpreted according to how the protocol rules define the error management policy. More precisely, the second node could simply retransmit the wrong bit towards the next node, without performing any error detection. Such retransmission could be realized without errors (\( P_{\text{EO}} \) or
also introducing a further error on the bit \((P_{XX})\); in this latter case the bit would assume its correct value at the end of the path. Alternatively, the second node could perform an error detection procedure on the frame to which the bit belongs (for example a CRC check). After detecting the error, the whole frame could be rejected or, possibly, a correction task could be executed to recover from the error and retransmit the frame. If the frame were rejected \(P_{XX}\) would be zero, because the second node does not retransmit the bit and so it could not introduce a further error.

In analytical terms, the probabilities on the error events can be defined through a specific joint probability function [19].

\[
\begin{align*}
P_{00} &= P_{2,1}(E_2^', E_1^') \\
P_{XX} &= P_{2,2}(E_2^', E_1^') \\
P_{OX} &= P_{2,3}(E_2^', E_1^') \\
P_{OX} &= P_{2,3}(E_2^', E_1^')
\end{align*}
\]

where \(E_i\) and \(\overline{E}_i\) represent the occurrence or non-occurrence of an error caused by the \(i\)-th node, respectively. These functions can be modeled through expressions depending on the primary BERs of the two nodes:

\[
\begin{align*}
P_{00} &= P_{2,1}(E_2^', E_1^') \\
P_{XX} &= P_{2,2}(E_2^', E_1^') \\
P_{OX} &= P_{2,3}(E_2^', E_1^') \\
P_{OX} &= P_{2,3}(E_2^', E_1^')
\end{align*}
\]

Figure 5. Topological equivalences in reference to noise effects, path with two nodes (a) and three nodes (b)

\[
\begin{align*}
P_{2,1}(E_2^', E_1^') &= P_{2,1}(BER_1, BER_1) \\
P_{2,2}(E_2^', E_1^') &= P_{2,2}(BER_1, BER_1) \\
P_{2,3}(E_2^', E_1^') &= P_{2,3}(BER_1, BER_1) \\
P_{2,3}(E_2^', E_1^') &= P_{2,3}(BER_1, BER_1)
\end{align*}
\]

The explicit form of these expressions can depend on several factors, such as the frame structure and the possible manipulations made by each node on the retransmitted frames.

These aspects are mostly specified at the level of communication protocol. Accordingly, in many applications the protocol rules should provide the basic information to derive an explicit expression for the functions in (9).

From the previous relationships, in the most general case, we can define the equivalent BER of the path constituted by the first two nodes (BER_2) as:

\[
\begin{align*}
BER_{-2} &= P_{XX} + P_{OX} + P_{XX} \\
&= P_{2,1-XX} + P_{2,1-XX} + P_{2,1-XX}
\end{align*}
\]

In regard to noise effects, equation (10) allows to derive a topological equivalence in which the two nodes can be replaced by a single node (N2) with BER equal to BER_2, as shown in Fig. 5a.

We can extend the BER propagation procedure, considering a network path with three nodes. In order to compute the equivalent BER of this path (BER_3), we can refer to a network path with two nodes. More precisely, the first two nodes can be merged in the N2 node, with BER equal to BER_2, as reported in Fig. 5b. In this way, by applying (10), BER_3 can be expressed as:

\[
\begin{align*}
BER_{-3} &= P_{3-N2-xx} + P_{3-N2-XX} + P_{3-N2-xx} \\
&= P_{2-1-N2-xx} + P_{2-1-N2-XX} + P_{2-1-N2-xx}
\end{align*}
\]

Equation (12) could be further developed to get a final expression based on the primary BERs of the single nodes. The NET_BER expression of equation (12) can be regarded as a measure of the noise impact on the network communications in terms of the primary node BERs. We can expect that networks with higher NET_BER are more affected by noise and they show worst performances. At the same time, networks with the same NET_BER could show similar performances with respect to noise effects and reliability properties. From this consideration we derive the idea to correlate networks with different BER distributions, as in equation (4).

It is important, of course, to consider the statistical nature of a NET_BER expression and the average validity of the resulting measures. This means that from NET_BER we cannot expect detailed indications, for example, on specific message transmissions in the network. Actually, the use of NET_BER can provide effective results considering the network activity for a suitable period of time, in which we have a consistent number of interactions between the network nodes.

The BER propagation procedure, shown in this Section, is strictly valid for a network constituted by a chain of \(N\) adjacent nodes, as in the case of a ring topology. However, it is possible to extend this procedure for more articulated networks too, by applying the BER propagation procedure on the node chains that constitute the network.

As an example, we can apply this approach to the network architecture in Fig. 6a. First of all, we could apply the BER propagation for the primary node chains of the network, i.e. the chains of nodes always involved in the data transmissions between two nodes. In this way, we reduce all the primary node chains to single nodes associated to an equivalent BER, as shown in Fig. 6b. At this point, we could derive a NET_BER expression for
the whole network by summing up all the BER contributions in the network (i.e. the equivalent BERs of the primary node chains and the BERs of the single nodes not involved in the reduction process). Alternatively, a NET_BER expression could be achieved by repeating the BER propagation procedure on the equivalent network of Figure 6b. However in this case the network is not constituted by primary node chains, because the data stream between two distinct nodes can proceed along different paths. For example, a data transmission between Node 1-3 and Node 5-6 could pass through Node 4, Node 7-8, or also the path given by Node 7-8 and Node 4. Nevertheless, we can apply the BER propagation procedure, for example choosing one of these paths. In particular, we could choose the path more involved in the data transmissions, at least from a statistical viewpoint, in the examined network activity. By applying this solution iteratively, the whole network can be reduced to a single node with BER equal to the NET_BER expression.

The next sections illustrate a case study in which our methodology is applied to evaluate the reliability performances of MOST-based networks.

Figure 6. Extension of the BER propagation procedure for a generic network architecture; (a) network taken as example (b) equivalent network after the reduction of the primary node chains.

V. OVERVIEW OF THE MOST PROTOCOL AND MODELING IN SYSTEMC LANGUAGE

A. Basic Features of a MOST Network

MOST is a communication protocol promoted by a consortium of several car manufacturers and suppliers (the MOST Cooperation), with the primary intent to standardize such technology for multimedia networking in the automotive industry [20]. The latest developments have led to the MOST 150 specifications, which introduce important enhancements in bandwidth and transport capabilities [21], [22].

A MOST network is constituted by distinct devices communicating via an optical fiber channel. Each device can include several application units, identified by specific entities called Fblocks. All the transmissions in a MOST network take place at Fblock level. The network architecture has a ring topology, where a Timing Master device generates a flow of frames used by the other devices for synchronization and communication, see Fig. 7. In this structure all the nodes share the same physical channel, avoiding a strict contention for the communication resources.

The transmissions in a MOST network take place onto three logical sub-channels for the transport of control, packet and streaming data. These sub-channels are allocated in specific segments of the MOST frame, as shown in Fig 8. Control messages are multiplexed in 4 bytes/frame pieces and are placed in a header field including also a preamble and some administrative flags. The architecture of a MOST network does not guarantee a direct connection for Fblocks placed in distinct devices, causing the transition of the exchanged messages across all the intermediate nodes. Each of these transitions entails several elaborations on the transmitted frames, e.g. opto-electric conversions, with the possibility of a significant contribution to the overall noise effects. Due to the sum of the single noise contributions, the effects on the communication reliability can be more relevant for large networks, within the limit of 64 nodes fixed by the MOST protocol.

To apply our noise analysis methodology on MOST networks, it is firstly necessary to fix the activities to be monitored, which should be those network tasks more critical for the system reliability. In our analysis we have examined two network tasks fundamental for the feasibility of MOST applications, i.e. device locking and network creation. As will be explained in Section VI, the noise effects on such activities can be studied by evaluating the BER impact on the preamble and control bytes of the transmitted frames.

B. Modeling of MOST networks in SystemC

SystemC is a C++ based language usable for the description of embedded HW/SW systems, which has known wide diffusion in the last years [23], [24]. The SystemC modeling capabilities are suitable to address simulation efficiency and analysis accuracy, by allowing the coexistence of different abstraction levels in the same description. In particular, the system functionality can be
defined through a basic high level representation, in which only the tasks to be monitored can be represented in detailed way. In comparison to Matlab, often used as descriptive language in system-level analysis, SystemC is more specialized in the design of an electronic digital system. This is mainly due to modeling constructs more effective for the description of hardware components as well as a simulation engine optimized for the run-time behavior of digital systems.

In our analysis, we implemented in SystemC language a virtual prototype of MOST network, considering as main guideline the efficient evaluation of noise effects. Each network device has been defined through a sc_module (the SystemC component for the modeling of modular units), with an internal implementation constituted by two layers, see Fig. 9.

![SystemC model of a single MOST network node](image)

The direct interaction with the channel is covered by a sub-module called INIC [25], which is responsible for all the fundamental elaborations on the incoming frames. We described in detail the main INIC tasks, with the addition of a configurable routine emulating the BER of the node. The second layer comprises the Fblocks instanced in the device, which have been modeled in compliance with the MOST Function Catalog specifications [26].

We modeled through functional descriptions the coordination Fblocks (the NetBlock, Timing Master, Network Master and Connection Master), that are primarily involved in the management tasks. Only the communication primitives of some source/sink components have been defined in the application Fblocks. The interactions between the instanced Fblocks and the INIC sub-module are implemented via a connection interface compatible with a variable number of Fblocks.

Finally, the optical fiber links are modeled by means of basic SystemC channels, called sc_signals.

The resulting network prototype presents a flexibility appropriate for the aims of our analysis. In particular, each device can be easily adapted to different BERS and Fblock instances. Moreover, the number and the position of the devices in the network can be modified through simple code rearrangements. These features allow an efficient and fast reconfiguration of the simulation scenarios for different assignments of the noise variables (N, BER). The realization of the whole prototype consists of about 8000 lines of SystemC/C++ code, developed by a two-developers team over one month.

In the system simulation phase, this network prototype has been set up to reproduce the monitored network activities, i.e. device locking and network creation, in the reference noise configurations. From these simulations, we have derived statistical relationships between the noise variables (N, BER) and significant performance measures. All the simulations have been carried out in few weeks, on a PC unit with a 2500 MHz dual core processor and 4GB of RAM. As compilation and execution tool, we have used a Visual C++ development environment. Each simulation has required an execution time depending on the size of the tested network architecture, varying from few minutes up to two hours in the case of networks with 60 nodes. The results achieved from these simulations are discussed in the next section, which also describes more in detail the monitored activities and the considered performance measures.

VI. MONITORED ACTIVITIES AND SIMULATION RESULTS

A. Device Locking

The lock condition is a major prerequisite for the correct working of a network device. Briefly summarizing, in a MOST network each device must be synchronized with the data flow propagating along the network, such that the single frames can be properly recognized. The failure of such synchronization entails an unlock state and the impossibility to process the input data stream. As a consequence, frequent unlock events should be prevented as much as possible, being very harmful in terms of elaboration times and transmission quality.

The lock condition comprises four main states, i.e. normal unlock, critical unlock, normal lock and stable lock. Unlock events can be easily related to the noise effects at bit level since, for typical INIC implementations, the lock states are defined by the recognition of the frame preamble [25]:

- **Normal Unlock**: declared if for two consecutive frames the correct preamble at the correct time is not received.
- **Critical Unlock**: declared if the time sum of Normal Unlock exceeds 70 ms without gaining a stable lock in-between. The main consequence is a network shutdown.
- **Normal Lock**: declared if three consecutive preambles are received with the correct timing.
- **Stable Lock**: stated after 100 ms continuous lock.

Every node downstream from the location that caused the unlock, up to the Timing Master, detects the unlock. The nodes downstream of the Timing Master up to the location that caused the unlock does not detect the unlock [21].

These rules determine precise correspondences between the lock states in a network device and the noise variables (N, BER). As performance measure, we have considered the average percentage of the lock times in which a device operates, along a period of time constituted by the transmission of 1000 frames from the Timing Master. The choice of such measure is motivated by the good trade-off between simulation effort, result relevance and statistic validity.
The simulation results are reported in Fig. 10, in which the horizontal and vertical axes report respectively the number of network nodes and the associated BER. The traced curves represent the (N, BER) couples for definite percentages of the average lock time per device. These curves are derived by the interpolation of points representing the mean value of simulations with the same (N, BER).

The curves in Fig. 10 circumscribe the (N, BER) zones related to different lock condition. From this results we can achieve two useful indications on system reliability: the (N, BER) couples that guarantee the best performances (lock times higher than 90%) and the limits for sustainable network activities. Quite reasonably, these limits can be fixed in the (N, BER) zone with lock times not lower than 50%, thus introducing a restriction for the acceptable (N, BER) values. For any network size, it is also evident how an acceptable lock imposes the constraint BER << 10^{-1} for all the nodes. In other words, the functionality of the whole network can be compromised if one or few nodes present an excessive noise level. This conclusion shows the higher vulnerability of large networks, in which such risk is more probable.

![Figure 10. Average percentages of the lock times per device as a function of the noise variables](image)

These simulation results can be extended to an arbitrary BER distribution, whose position in the lock zones can be estimated by applying (5). If the condition (6) is not verified, we can conclude that the considered distribution is outside the lock zone for sustainable network activities.

Figure 11 reports the percentage of the average lock time per device as a function of the number of nodes in the network for a BER=1/500.

### B. Network Creation

Network creation is the preliminary task in a MOST application and it consists in a massive exchange of control messages that allow the Network Master to register all the enabled Fblocks. This procedure involves extensively all the network devices and its study is appropriate to get indications on network size and Fblock disposition.

In our analysis we have considered network layouts with 10, 30 and 60 application Fblocks, taken as reference for small, medium and large operative contexts. For each network layout, we have simulated three possible dispositions given by one, two and three Fblocks per device. Each of these dispositions corresponds to a specific number of nodes.

This study can be useful to understand how some implementation options can influence the system reliability. More precisely, including more Fblocks in a device leads to smaller network sizes, with less noise accumulation. On the other hand, longer and more error-vulnerable messages are necessary, because each device must specify the features of all its Fblocks.

| TABLE I. PERFORMANCE RESULTS IN A NETWORK CREATION WITH 10 FBLOCKS |
|------------------|------------------|------------------|------------------|
| Fblocks per node | 3 Fblocks per node | 2 Fblocks per node | 1 Fblock per node |
| 10 nodes         | 15 nodes          | 30 nodes          |                  |
| 73 messages      | 83 messages       | 113 messages      |                  |
| total retries    | total retries     | total retries     | total retries    |
| peak retries     | peak retries      | peak retries      | peak retries     |
| 200              | 400              | 600              |                  |
| ERROR            | ERROR            | ERROR            | ERROR            |
| 39               | 8                | 22               | 5                |
| ERROR            | ERROR            | ERROR            | ERROR            |
| 8                | 15               | 11               | 113              |
| ERROR            | ERROR            | ERROR            | ERROR            |
| 4                | 3                | 3                | 14               |
| 48               | 16               | 12               | 19               |
| ERROR            | ERROR            | ERROR            | ERROR            |
| 1500             | 1000             | 1500             |                  |
| 7                | 6                | 4                | 3000             |
| 2                | 2                | 2                | 4                |
| 2                | 10               | 3                | 2                |
| 3                | 9                | 5                | 19               |
| 19               | 5                |                  |                  |
| 39               | 9                |                  |                  |
| 113              |                  |                  |                  |
| 113              |                  |                  |                  |

To define the performance measure, we have considered the mechanisms provided by MOST to secure the control messages, constituted by CRC check and ACK/NAK transmissions with automatic retry. These error-handling capabilities allow the definition of two performance measures for evaluating the noise effects. They are the sum of all the message retries occurred during the network creation (total retries), and the highest number of retries occurred for a single message (peak retries). Total retries is an indicator that makes possible an overall comparison of different network architectures, providing also indirect estimations of latency and power dissipation. Peak retries can be used to fix the maximum
number of retries per message, thus to guarantee a correct functionality without loss of information. In this way, peak retries can define also a sustainability limit to mark malfunctioning states. In each simulation this limit has been set up to 20 retries, which represents a typical constraint in the communications between Fblocks in a MOST network. If the retransmission of a message exceed 20 retries, an error condition (ERROR) is raised causing the failure of the network creation.

Tables I-III show the results achieved from the simulations, i.e. the average total retries and peak retries with different network layouts and BER. For each network architecture, the number of messages exchanged during the procedure is also specified.

<table>
<thead>
<tr>
<th>TABLE II. PERFORMANCE RESULTS IN THE NETWORK CREATION WITH 30 FBLOCKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/BER</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>1200</td>
</tr>
<tr>
<td>1500</td>
</tr>
<tr>
<td>1800</td>
</tr>
<tr>
<td>2000</td>
</tr>
<tr>
<td>2500</td>
</tr>
<tr>
<td>3000</td>
</tr>
<tr>
<td>4000</td>
</tr>
<tr>
<td>5000</td>
</tr>
<tr>
<td>10000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III. PERFORMANCE RESULTS IN THE NETWORK CREATION WITH 60 FBLOCKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/BER</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>3000</td>
</tr>
<tr>
<td>3500</td>
</tr>
<tr>
<td>4000</td>
</tr>
<tr>
<td>5000</td>
</tr>
<tr>
<td>7500</td>
</tr>
<tr>
<td>10000</td>
</tr>
<tr>
<td>12500</td>
</tr>
<tr>
<td>15000</td>
</tr>
</tbody>
</table>

In regard to noise tolerance, the networks with more Fblocks per node and smaller sizes show better results, since the performance measures assume valid values for higher BERs. Moreover, for each network layout, the networks with one Fblock per node show by far the highest total retries, implying that the influence due to the high number of messages and the large path length is definitely predominant in this Fblock disposition.

VII. VERIFICATION OF THE ANALITICAL MODEL

This section deals with the verification of (5), by comparing performances estimated through this formula with measures achieved via simulation. For this purpose, we have examined three non-uniform BER distributions, respectively associated to networks with 10, 20 and 30 nodes:

<table>
<thead>
<tr>
<th>D1 (N=10, BER=1/2000)</th>
<th>D2 (N=20, BER=1/3500)</th>
<th>D3 (N=30, BER=1/2500)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N=20, BER=1/2000)</td>
<td>(N=20, BER=1/3500)</td>
<td>(N=30, BER=1/2500)</td>
</tr>
<tr>
<td>1/2000, 1/3000, 1/6000, 1/3500, 1/2800, 1/1500, 1/4000, 1/3500, 1/3200, 1/3000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/1500, 1/5500, 1/1000, 1/6000, 1/5000, 1/3500, 1/3000, 1/2500, 1/5000, 1/4000, 1/1500, 1/3800, 1/4500, 1/3500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/2500, 1/3000, 1/1000, 1/3200, 1/5000, 1/3500, 1/2000, 1/2800, 1/2100, 1/2400, 1/4000, 1/1200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

By applying (5), such distributions can be considered equivalent to the following reference configurations:

<table>
<thead>
<tr>
<th>D1 (N=10, BER=1/2000)</th>
<th>D2 (N=20, BER=1/3500)</th>
<th>D3 (N=30, BER=1/2500)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1 (N=10, BER=1/2000)</td>
<td>D2 (N=20, BER=1/3500)</td>
<td>D3 (N=30, BER=1/2500)</td>
</tr>
<tr>
<td>1/2000, 1/3000, 1/6000, 1/3500, 1/2800, 1/1500, 1/4000, 1/3500, 1/3200, 1/3000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/1500, 1/5500, 1/1000, 1/6000, 1/5000, 1/3500, 1/3000, 1/2500, 1/5000, 1/4000, 1/1500, 1/3800, 1/4500, 1/3500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/2500, 1/3000, 1/1000, 1/3200, 1/5000, 1/3500, 1/2000, 1/2800, 1/2100, 1/2400, 1/4000, 1/1200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We have instrumented the SystemC network prototype with the distributions D1, D2 and D3, simulating each configuration in the device locking and network creation activities. Table IV compares the resulting performances with the counterparts related to the equivalent reference configurations and reported in Tables I-III. The Fblock dispositions related to network creation are reported too.

We can notice a good matching between the estimated performances and the simulation results. The correspondence is very strict for the lock condition and is good enough as concerns total and peak retries. All that shows the validity of the transition formula we have derived and the compliance with the noise effects in a MOST network.

<table>
<thead>
<tr>
<th>TABLE IV. COMPARATIVE EVALUATION OF THE PERFORMANCES PREDICTED BY THE TRANSITION FORMULA</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1 (N=10, 1 Fblock per node) &amp; non-uniform BER</td>
</tr>
<tr>
<td>equivalent uniform BER</td>
</tr>
<tr>
<td>D2 (N=20, 3 Fblocks per node) &amp; non-uniform BER</td>
</tr>
<tr>
<td>equivalent uniform BER</td>
</tr>
<tr>
<td>D3 (N=30, 2 Fblocks per node) &amp; non-uniform BER</td>
</tr>
<tr>
<td>equivalent uniform BER</td>
</tr>
</tbody>
</table>

VIII. CONCLUSIONS

This paper has presented a novel methodology for system-level noise analysis of networked systems, entirely based on protocol specifications. In order to...
address comprehensive and fairly evaluations, we have proposed a simulation/analytical approach in which the reliability properties are studied as a function of the BERs of the network nodes. The simulation phase is carried out on a virtual prototype and is restricted to uniform BER distributions. The performances for generic BER distributions can be quickly estimated by means of an analytical transition formula, which allows to reuse the simulation results. All the methodological steps have been illustrated and verified through a case study on the MOST 150 protocol, from which we have achieved meaningful indications on the reliability properties of MOST-based networks.

As future developments, further researches could be conducted to refine the methodology, especially for the analytical part. In our work we have derived a NET_BER expression for the ring architecture of MOST networks by following a BER propagation procedure; the experimental results have shown the validity of the achieved expression. In Section IV we have also provided some guidelines for applying a BER propagation approach on complex network topologies.

It would be interesting to consider other approaches for deriving a NET_BER expression (possibly defining formal rules for their application). In this way, it would be possible to carry out comparative performance evaluations on the MOST protocol as well as on other communication protocols. All that could provide further validations of the proposed methodology and valuable indications for the analytical part.

### Appendix

This appendix describes the passages that lead to the transition formula (5), by applying a BER propagation procedure in compliance with the MOST protocol rules. As starting point, we have to consider the basic equations for a two-node path (9) – (11), which have general applicability. First of all, let us consider a network path with two nodes and BERs equal to BER_1 and BER_2. In order to express the equivalent BER of the path as a function of BER_1 and BER_2, we have to recall the transmission modalities in a MOST network; in particular, we can focus on the exchange of messages between a sender and a receiver node. In such task, each intermediate node simply retransmits the messages; the data stream is not subject to elaborations that could be influenced by possible errors introduced in the previous nodes. This implies that the BER of a node should have negligible effects on the BER of the next node. Accordingly, the statistical independence of the BERs of adjacent nodes can be considered a valid assumption. Furthermore, the intermediate nodes are not allowed to suppress or correct a message in case of corrupted bits. Only when the message reaches the receiver node, a CRC check is performed in order to validate the data integrity.

Taking into account these communication rules, we can derive the following expressions for (8) in the case of a MOST-based network:

\[
P_{go} = (1 - BER_1)(1 - BER_2) \\
P_{xo} = BER_1 (1 - BER_2) \\
P_{ox} = (1 - BER_1)BER_2 \\
P_{xx} = BER_1BER_2
\]

From these relationships, we can define the equivalent BER of the path (BER_2) as:

\[
BER_2 = P_{xo} + P_{ox} = BER_2 + BER_1 - 2BER_1BER_2
\]

In (A1) we have applied the statistical independence of the primary BERs. In this case the event P_{xx} is not to be included in the definition of BER_2, because a double error on the same bit would result in the bit keeping its correct value. At this point, repeating the procedure shown in Section IV, we can achieve the equivalent BER for a three-node network path:

\[
BER_3 = BER_1 + BER_2 - 2BER_1BER_2 - 2BER_1BER_3 (A3)
\]

Iterating the BER propagation for the whole N-node path, we can achieve a NET_BER expression for the network:

\[
NET_BER = BER_1 + BER_2(N - 1) - 2BER_1BER_2(N - 1)
\]

Equation (A4) can be further developed to get an expression based on the primary BERs:

\[
NET_BER = \\
BER_1 + BER_2 + BER_1(N - 2) - 2BER_1BER_2(N - 2) \\
- 2BER_1(BER_2 + BER_2(N - 2) - 2BER_1BER_2(N - 2)) \\
= BER_1 + BER_2 + BER_2(N - 2) - 2BER_1BER_2(N - 2) \\
- 2BER_1 + BER_2(N - 2) - 2BER_1BER_2(N - 2) \\
- 2BER_1 + BER_2(N - 2) \\
= BER_1 + BER_2 + BER_2(N - 2) - 2BER_1BER_2(N - 2) \\
- 2BER_1BER_2(N - 2) - 2BER_1BER_2(N - 2) \\
= BER_1 + BER_2 + BER_2(N - 2) - 2BER_1BER_2(N - 2)
\]

As experimentally shown in Section VI, the constraint BER_2 < 10^{-5} must be true to guarantee an acceptable lock condition. For this reason, also considering that N is limited to 64 nodes, the terms with more than two BER factors can be neglected, leading to the approximations made in eq. (A5).

By iterating the BER propagation for the whole N-node path, without considering the approximations in (A5), we would achieve the following NET_BER expression for the network:

\[
NET_BER = (BER_1 + BER_2 + BER_1(N - 2) - 2BER_1BER_2(N - 2) \\
- 2BER_1BER_2(N - 2) - 2BER_1BER_2(N - 2) \\
= BER_1 + BER_2 + BER_2(N - 2) - 2BER_1BER_2(N - 2)
\]
\[ \text{NET\_BER} = \sum_{i=1}^{N} \text{BER}_i - 2\sum_{i=1}^{N} \sum_{j=i+1}^{N} \text{BER}_i \text{BER}_j \]
\[ + (-2)^i \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=i+1}^{N} \text{BER}_i \text{BER}_j \text{BER}_k \]
\[ + (-2)^i \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=i+1}^{N} \sum_{m=k+1}^{N} \text{BER}_i \text{BER}_j \text{BER}_k \text{BER}_m \]
\[ + \ldots \]

Finally, by applying the simplification related to an acceptable lock, we achieve the following \text{NET\_BER} expression as a function of the primary BERs:

\[ \text{NET\_BER} = \sum_{i=1}^{N} \text{BER}_i - 2\sum_{i=1}^{N} \sum_{j=i+1}^{N} \text{BER}_i \text{BER}_j \]
\[ \text{BER}_1 = \text{BER}_2 = \ldots = \text{BER}_N = \text{BER}, \text{ we fall into a uniform BER distribution and equation (A7) assumes the form} \]

\[ \text{NET\_BER}^* = N\text{BER} - 2N\text{BER}^2 \sum_{i=1}^{N} (i-1) = \]
\[ = N\text{BER} - N(N-1)\text{BER}^2 \]
\[ \text{(A8)} \]

in which this identity has been applied:

\[ \sum_{i=1}^{N} (i-1) = \frac{N(N-1)}{2} \]
\[ \text{(A9)} \]

By setting up an equation between (A7) and (A8), we can establish a connection between uniform and arbitrary BER distributions related to the same \text{NET\_BER}:

\[ \begin{align*}
\text{NET\_BER} &= \text{NET\_BER} \\
N\text{BER} - N(N-1)\text{BER}^2 &= \sum_{i=1}^{N} \text{BER}_i - 2\sum_{i=1}^{N} \sum_{j=i+1}^{N} \text{BER}_i \text{BER}_j \\
\text{BER}_1 = \text{BER}_2 = \ldots = \text{BER}_N = \text{BER}. 
\end{align*} \]
\[ \text{(A10)} \]

From (A10) we obtain a transition formula between the noise variables of uniform and arbitrary distributions that can be considered equivalent in terms of noise effects. Introducing the following notations

\[ \begin{align*}
C &= \sum_{i=1}^{N} \text{BER}_i - 2\sum_{i=1}^{N} \sum_{j=i+1}^{N} \text{BER}_i \text{BER}_j \\
\alpha(N) &= N(N-1)
\end{align*} \]
\[ \text{(A11)} \]

equation (A10) can be rewritten as a second order equation in the BER variable:

\[ \alpha(N)\text{BER}^2 - N\text{BER} + C = 0 \]
\[ \text{(A12)} \]

In compliance with the constraint given by (6), the valid solution is equal to (5).

REFERENCES

Giovanni B. Vece received the Dr.Ing. degree in electronics engineering and the Ph.D. degree in electronic and telecommunication engineering from the Università Politecnica delle Marche, Ancona, Italy, in 2002 and 2005 respectively. He is currently involved in some collaborations with the Università Politecnica delle Marche, concerning research projects based on the SystemC framework. In particular, his activities are focused on system level methodologies and power analysis tools.

Eros Mazza received the Dr.Ing. degree in electronics engineering from the Università Politecnica delle Marche, Ancona, Italy, in 2009. After graduating he joined a spin-off company linked to the Università Politecnica delle Marche. His research activities concern networking applications for industrial contexts with reference to MOST and ZigBee standards.

Massimo Conti is Associate Professor at the Università Politecnica delle Marche, Ancona, Italy. His research activity in the field of Microelectronics is mainly devoted to System Level Design of low power Integrated Circuits, wireless sensor network and NFC. Coauthor of more than 150 papers on Int. Books, Journals or Conferences. Member of the editorial board of International Journals: “The Scientific World Journal”. Editor of 5 International Books or Journal Special Issue.