Computers in Industry 74 (2015) 186–200

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Model checking response times in Networked Automation Systems using jitter bounds

Seshadhri Srinivasan [a,*], Furio Buonopane [b], Juri Vain [c], Srini Ramaswamy [d]

[a] Department of Engineering, University of Sannio, Benevento, Italy
[b] Department of Engineering, University of Naples, Federico II, Naples, Italy
[c] Department of Computer Science, Tallinn University of Technology, Tallinn, Estonia
[d] ABB Inc., USA

ARTICLE INFO

Article history:
Received 14 January 2015
Received in revised form 15 June 2015
Accepted 30 June 2015
Available online 15 September 2015

Keywords:
Networked Automation Systems (NAS)
Jitter
Timed-automata
Model checking
Network latencies
Industrial automation
Response time (RT)

ABSTRACT

Response time (RT) of Networked Automation Systems (NAS) is affected by timing imperfections induced due to the network, computing and hardware components. Guaranteeing RT in the presence of such timing imperfections is essential for building dependable NAS, and to avoid costly upgrades after deployment in industries.

This investigation proposes a methodology and work-flow that combines modelling, simulation, verification, experiments, and software tools to verify the RT of the NAS during the design, rather than after deployment. The RT evaluation work-flow has three phases: model building, modelling and verification. During the model building phase component reaction times are specified and their timing performance is measured by combining experiments with simulation. During the modelling phase, component based mathematical models that capture the network architecture and inter-connection are proposed. Composition of the component models gives the NAS model required for studying the RT performance on system level. Finally, in the verification step, the NAS formal models are abstracted as UPPAAL timed automata with their timing interfaces. To model timing interfaces, the action patterns, and their timing wrapper are proposed. The formal model of high level of abstraction is used to verify the total response time of the NAS where the reactions to be verified are specified using a subset of timed computation tree logic (TCTL) in UPPAAL model checker. The proposed approach is illustrated on an industrial steam boiler deployment.

1. Introduction

Networked Automation Systems (NAS) in industrial automation refer to systems with networked sensors, actuators, and controllers that interact with strict timing requirements [1]. NAS need to accomplish control actions ranging from small logic manipulation to complex computation. Leveraged by the dependability, adaptability, and flexibility, NAS are being increasingly deployed in all industries. NAS differ from networked control systems (NCS) as the software used for automation are written following standards such as IEC 61131 [2,3]. Further, NAS deals not only with network induced timing imperfections, but are also concerned with the timing discrepancies due to hardware and computation.

In NAS there is a strong synergy of control, computing, computing and cognition (C^4) as the controller is basically a computing unit that makes decisions considering changes in operating conditions, environment, and system states by exchanging information with the various components. As a result, design decisions and constraints in one domain affect the performance of the other and vice-versa. Though, C^4 synergy has many significant advantages, timing imperfections induced due to the confluence of these components affect the NAS performance as a whole.

Response time (RT)[1] which is in the order of few milliseconds is an important requirement that needs to be met by the NAS during its entire life-cycle. RT is an important ingredient of the specification in industrial automation systems such as servo [4] and other important applications [5]. Current practice in industry to verify RT from data-sheets or experiments during deployment phase of the automation life-cycle shown in Fig. 1 is too late to

[*] Corresponding author.
E-mail addresses: seshadhri.srinivasan@unisannio.it (S. Srinivasan), furio.buonopane@gmail.com (F. Buonopane), vain@ioc.ee (J. Vain), srini@ieee.org (S. Ramaswamy).

[1] RT is defined as time taken from the generation of sensor signal to the completion of actuation in a technical process (a system being controlled)

http://dx.doi.org/10.1016/j.compind.2015.06.012
0166-3615/© 2015 Elsevier B.V. All rights reserved.

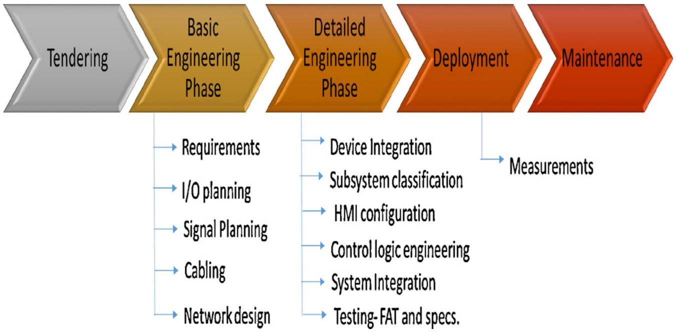

Fig. 1. Typical automation life cycle.

guarantee faithful replication of design specification and to modify NAS design [6]. Therefore, methods that can verify RT during design phase, rather after deployment are required in automation industries.

Traditionally, simulation [7], analytical methods [8,9], or network calculus [10] have been used for studying response time in NAS. These approaches focus on obtaining the bounds that model timing fluctuations. While, simulation and direct measurements based analysis methods provide RT estimates in specific use cases they usually can cover only a limited number of behaviours in reasonable time [6]. When system integration is in question the non-exhaustive state-space exploration methods alone are clearly insufficient specifically for safety and time-critical applications, where provably correct timing is utmost important. Moreover, recent trend in automation industry to use heterogeneous networks and multi-core processors for executing a control computation results in information flows through various cores and network. In these scenarios the conventional methods such as simulation, analytical and network calculus cannot be used to properly verify the performance.

Timed model-checking in contrast is an exhaustive state-space exploration based verification technique to analyze critical systems [11]. Verification is done using high level formal models, where the details not relevant for the analysis are abstracted away from the model. Therefore, model checking provides provably correct RT estimates, where all possible behaviours of the model are taken into account. But, the scalability of the model checking is sensitive to the size of the state spcae of NAS component models. Therefore, embedding the results of low level simulations and direct measurements from experiments of NAS components into abstracting integration models used in model checking opens up new prospects in studying RT behavior of complex systems.

Though formal verification methods have been used for verifying control system behaviours [12], their use to verify RT in NAS has been investigated only recently. To our best knowledge the first effort to study timing performance of NAS using methods from model checking was by Frey et al. [13], wherein the probabilistic model checking (PMC) was used to study the component failures. Applying model-checking for verifying the performance of a flexible manufacturing plant was presented in [14]. Parametric model checking by iterative proofs was proposed in [15]. The methods developed in these investigation are applicable to limited class of problems due to the absence of systematic modelling framework to capture the timing imperfections. Vogel-Heuser et al. [16] presented a component oriented modelling approach to capture the timing requirements and specifications that could be used to verify the timing performance of NAS. The advantage of the proposed modelling approach is that, it inherently captures the network architecture. The need to combine simulation with model driven verification using probabilistic model checking was highlighted in [16]. More recently, Ramaswamy et al. [1] used jitter[2] [17] to verify the RT of NAS using model checking. However, the role of embedding experiments and simulation within model checking was not studied in these investigations.

To overcome the shortcomings with the existing methods, this investigation combines experiments and simulations with timed-model checking. As a result, it is possible to exhaustively verify the RT of NAS for various scenarios and control algorithm flows. For example, the control flow through various sensors, heterogeneous network, and processors can be verified using the proposed approach. Further, the state-space explosion problem that typically limits the scalability of timed model checking is eliminated due to the embedding of aggregated results of simulation and experiments. The main contributions of this investigation are: (i) a component based modelling framework that captures the NAS architecture, components, timing requirements and specifications, (ii) jitter specification for component model of NAS using experiments, software tools, and simulation, (iii) procedure to verify the RT using timed-model checking where structural modelling by means of predefined model patterns helps in specifying jitter bounds (to model the timing imperfections), (iv) demonstrates the role of simulation in building models of higher abstraction and to identify the critical points of verification, and (v) a work-flow for simulation driven verification of timing performance in NAS.

The paper is organized into five sections including the introduction. Section 2 presents the jitter based timing model of NAS, while the use of model patterns is discussed in Section 3. The work-flow for NAS timing performance verification is presented in Section 4. Industrial boiler case study is presented in Section 5, and conclusions are drawn on obtained results are summarized in Section 6.

2. Modelling timing imperfections in NAS

Main idea behind model checking is built a model that abstracts away unwanted information and provides a model that can be used to study a given behaviour. NAS is a distributed system having interacting hardware, software and network components. Their implementation varies depending on the automation solutions provider, the designer and also the application. Therefore, models intended to study timing behaviour have to be more generic, scalable and capture the network architecture to study distirbuted systems. Still, they should capture the required information for model checking response time.

A good approach in this scenario would be to use component models along with the timing properties and specifications, later composition of these components can model the entire system. Component based approach has been studied by Ferrari et al. [18] to model the network jitter in Profinet class. More recently, Vogel-Heuser et al. [16] modelled NAS by composing component models. The resulting composed model was called the time chain of the NAS.[3] The advantage of the time-chain model is that, it captures the network architecture inherently in its design. Therefore, in this investigation, we extend the component model proposed in [16] by specifying jitter bounds, latencies, and jitter behaviour for each NAS component. The composition of these models gives a set of time chains that will be used in model checking RT.

The timing imperfections in NAS components are assumed to have two parts- an average value plus the jitter. Such an approach is widely used by many investigations (see [19] and references therein).

[2] According to IEEE “Jitter is the time related abrupt, spurious variation in related interval”.

[3] Time chain is a component model of the physical entity defining the timing properties and specifications.