Journal of Systems Architecture 88 (2018) 23–32

Contents lists available at ScienceDirect

Journal of Systems Architecture

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

Verification of network end-to-end latencies for adaptive ethernet-based cyber-physical systems

Martin Manderscheid, Gereon Weiss*, Rudi Knorr

Fraunhofer Institute for Embedded Systems and Communication Technologies ESK Hansastrasse 32, Munich, Germany

ARTICLE INFO

Keywords:
Cyber-physical systems
Embedded systems
Runtime variability
Network performance analysis
Network performance-verification
Real-time system architecture

ABSTRACT

As Cyber-Physical Systems (CPS) are evolving towards flexible and smart systems, their dependable communication becomes a decisive factor. In order to still guarantee a predictive and real-time behavior, verifying the network performance of such adaptive systems is vital. Therefore, the performance-verification has to consider the runtime variability while scaling for larger number of applications and networks in CPS. We introduce a novel performance-verification approach with integrated variability enabling the analysis of adaptive Ethernet-based CPS. It incorporates a formal model capturing all relevant characteristics for deriving safe communication bounds. Its soundness has been evaluated in an extensive automotive case study and several changing test setups targeting scalability. The results show that this integrated variability approach is superior to a common static analysis and previously utilized heuristic. In direct comparison it outperforms static analysis by up to 95 percent within the evaluated automotive system. Moreover, the results show that it scales well and provides a profound basis for analyzing larger adaptive networked systems.

1. Introduction

Today's cyber-physical systems (CPS) enable a broad range of applications based on software and electronics. To that end, CPS are integrating more and more actuators and sensors with increasing communication demands, such as high definition cameras or laser scanners in the automotive domain. Thus, the needed network bandwidth rises to a level which can only be satisfied by high bandwidth communication technologies, such as Ethernet [1–3]. While Ethernet provides high bandwidth to applications, the planning of such interactive networks can no longer rely on static traffic assumptions. CPS have to adapt dynamically to various contexts (cf. [4]). Hence, a static network planning cannot adequately address these flexibility concerns anymore. Modern automobiles for example, may activate different combinations of driver assistance functions, depending on the driving context. For instance, while backing into a parking space the rear view camera function and the park distance control functions are simultaneously activated. During automated highway driving, the activation of the navigation system and the cruise control would rather make sense.

CPS are often produced at high volume while production costs are tried to be kept at a low level. This in turn, has a great impact on the development process. Here, one tries to predict the required hardware resources as exact as possible. At the same time, CPS involve high dependability demands and their functionalities - or also so-called *features* - must in most cases meet time-critical requirements. For example, when pictures recorded through a rear view camera are sent to the head-unit of an automobile, they have to be displayed within a defined time interval. The fulfillment of those requirements by the network architecture design is verified within the *network performance-verification*. The objective of this step inside the design process of CPS is to meet the requirements of all applications with the lowest possible amount of hardware resources. At this process step, engineers are facing the challenge of mastering the complexity of a vast amount of runtime configurations and an exact network performance-verification for a cost-efficient and dependable network architecture at the same time. In this work, we contribute to this challenge by introducing:

- An improved scalable heuristic model for the network performance-verification of dynamic CPS, which does neither rely on manual runtime configuration definitions nor does it imply artificial restrictions on network topology or medium access methods,
- a corresponding method enabling the calculation of safe bounds for the minimum runtime configuration switching transition delay, and
- an automotive case study representing a typical use-case of real-time CPS proving the applicability of the presented approach.

In the following Section 2, we introduce related approaches and background to the introduced network performance-verification. As a

---

* Corresponding author.
E-mail addresses: martin.manderscheid@esk.fraunhofer.de (M. Manderscheid), gereon.weiss@esk.fraunhofer.de (G. Weiss), rudi.knorr@esk.fraunhofer.de (R. Knorr).

https://doi.org/10.1016/j.sysarc.2018.05.004
Received 16 September 2017; Accepted 10 May 2018

running example, we provide a case study of an automotive video system in Section 3. In the subsequent Section 4, we present our approach for the network performance-verification with integrated variability for dependable and adaptive CPS. Its validity, applicability and scalability is evaluated within the automotive case study and synthetic test-setups in Section 5. Afterwards, we conclude our work in Section 6.

2. Background and related work

Looking a decade or more back, network performance-verification [5–7] of distributed embedded systems was mainly covering static scenarios. That is, the analytic model covered one static software configuration consisting of a number of tasks competing for shared resources provided by the system architecture. Since then, distributed embedded systems are becoming more and more dynamic and open, thus, evolving to CPS. This step offered a multitude of possibilities, which made the performance-verification simultaneously very complex. Since CPS are often multi-functional and dynamic, design engineers are facing an immense number of features ending in an exploding number of possible runtime configurations. System designs of CPS often host safety related features and demand high dependability. Thus, worst-case bounds are generally required. This fact makes it difficult to find approximations, since all runtime configurations have to be covered by the performance-verification. To solve this, different kinds of approaches have been proposed.

One popular approach is to abstract from some dynamics of the system for verifying the system's performance in a rather but not completely static way [8–13]. The idea is, to manually group a set of system-wide software runtime configurations called modes [14] or scenarios [8], each configuration addressing a different context or situation during the runtime of the system. The performance of every runtime configuration has to be verified individually. Depending on the protocol for the configuration change, transitions have to be verified, too (cf. [15]). While those approaches extended the possibilities and allowed more resource-efficient system designs, their limitations are evident. The manual nature of describing runtime configurations limits the designable variability of the systems and thereby the resource-efficiency of the design.

Another branch of approaches (cf. [13,16]) tries to avoid the abstractions resulting from system-wide modes. Here, each software component owns a set of modes and thus, those approaches allow more dynamisms than system-wide mode approaches. However, for the performance-verification, all runtime configurations have to be covered. Since the overall runtime configurations are obtained by the cross product of the sets of component-modes, this results in a combinatorial explosion and makes it too complex finding an exact solution. In the approaches presented in [16] and [13], this has been solved through restrictions concerning the network topology and media access mechanisms. Both approaches rely on time division multiple access (TDMA). While these solutions help to reduce the complexity of network performance-verification, they entail significant drawbacks concerning resource-efficiency and bandwidth availability.

Additional works have been published [17–21] that are mainly based on confining the analyzed dynamics to local components. The presented approaches rely on interfaces between classical performance evaluation frameworks such as Real-Time Calculus [6] and more expressive concepts such as Timed Automata [22], Event Count Automata [23] or Lustre [24]. Through this hybrid approach, the combinatorial explosion can be avoided by the price of neglecting system-wide dynamics. Especially for the case of network performance-verification, this is obviously a significant disadvantage.

In summary, the concepts of system-wide modes, component-wide modes, and hybrid performance-verification imply significant drawbacks and leave a gap open that is worth spending a detailed look at. Targeting this opportunity, we will present an approach integrating variability concerns within the network performance-verification of CPS.

3. Automotive video system

Automobiles and their complex E/E architecture are prominent example systems evolving towards adaptive CPS with real-time requirements. Therewith, the in-vehicle video architecture plays an important part. It consists of several time-critical video functions, which all transfer data from different sources to the head-unit display, which visualizes the data. The following features are generally included in such a system:

- *Human Machine Interface (HMI)* feature generates a configuration menu video stream, which is displayed on the head-unit.
- *Navigation* feature calculates driving routes and generates a visualization for the driver.
- *Storage Movie* feature transfers video content from an entertainment electronic control unit (ECU) to the head-unit.
- *Bluray Movie* feature transfers video content from a Bluray Player to the head-unit.
- *Online Stream* feature transfers video content from an Antenna ECU to the head-unit.
- *Top-View* feature aggregates four video streams from four cameras mounted at different positions of the vehicle. Thereof, a merged video stream is generated which shows the vehicle from a bird's eye view.
- *Nightvision* feature sends a video stream from a nightvision camera to the head-unit.
- *Side-View* feature displays two video streams recorded from cameras at the vehicle's mudguards, thus, providing the driver a view into hardly visible areas, such as the case in crossroads.

All of these features have requirements on the maximum end-to-end transmission delay (cf. Table 1) and must be integrated in an adequate network architecture. We will introduce our variability-aware network performance-verification approach by further detailing this example in the following.

Table 1
Data dependencies and end-to-end delay requirements of the case study.

Source- and target-component	Avg. bandw. [Mbits]	Max. size appl.-frame [octet]	Max. Burst [octet]	L_{max} [octet]	Max. delay [ms]
SV Merge, SV Right CAM	6.1848	25,770	25,770	1526	33
SV Merge, SV Left CAM	6.1848	25,770	25,770	1526	33
TV Merge, TV Right CAM	6.1848	25,770	25,770	1526	33
TV Merge, TV Left CAM	6.1848	25,770	25,770	1526	33
TV Merge, TV Front CAM	6.1848	25,770	25,770	1526	33
TV Merge, TV Rear CAM	6.1848	25,770	25,770	1526	33
Head-Unit, Nightvision	6.1848	25,770	25770	1526	33
Head-Unit, Internet	25.7810	107,421	107,421	1526	100
Head-Unit, HMI	164.9762	687,401	687,401	1526	100
Head-Unit, Entertainment	164.9762	687,401	687,401	1526	100
Head-Unit, Navigation	164.9762	687,401	687,401	1526	100
Head-Unit, Bluray	164.9762	687,401	687,401	1526	100