Microprocessors and Microsystems 52 (2017) 401–415

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

Design optimization for security- and safety-critical distributed real-time applications

Wei Jiang[a,*], Paul Pop[b], Ke Jiang[c]

[a] School of Information and Software Engineering, University of Electronic Science and Technology of China, China
[b] Department of Compute, Technical University of Denmark, Denmark
[c] ÅF-Technology AB, Sweden

ARTICLE INFO

Article history:
Received 11 February 2016
Revised 23 July 2016
Accepted 9 August 2016
Available online 10 August 2016

Keywords:
Embedded system
Security
Safety
Energy
Design optimization

ABSTRACT

In this paper, we are interested in the design of real-time applications with security, safety, timing, and energy requirements. The applications are scheduled with cyclic scheduling, and are mapped on distributed heterogeneous architectures. Cryptographic services are deployed to satisfy security requirements on confidentiality of messages, task replication is used to enhance system reliability, and dynamic voltage and frequency scaling is used for energy efficiency of tasks. It is challenging to address these factors simultaneously, e.g., better security protections need more computing resources and consume more energy, while lower voltages and frequencies may impair schedulability and security, and also lead to reliability degradation. We introduce a vulnerability based method to quantify the security performance of communications on distributed systems. We then focus on determining the appropriate security measures for messages, the voltage and frequency levels for tasks, and the schedule tables such that the security and reliability requirements are satisfied, the application is schedulable, and the energy consumption is minimized. We propose a Tabu Search based metaheuristic to solve this problem. Extensive experiments and a real-life application are conducted to evaluate the proposed techniques.

1. Introduction

Embedded systems are increasingly used in critical areas, e.g., automotive and avionic electronic systems. Such systems are facing emerging challenges from security, reliability, energy, and timing requirements. In this paper, we refer to such systems as security- and safety-critical systems (SSCSs). SSCSs have to function correctly while meeting timing constraints even in the presence of faults. Such faults can be permanent, intermittent, or transient. In this paper, we focus on protecting the system against the most common type of faults, namely, the transient faults [1].

With the integration of new communication interfaces, SSCSs are exposed to increasingly severe security threats, hence the need of protecting sensitive communication information becomes of utmost importance [2]. The snooping, spoofing, or alteration of critical data does not only compromise security but can also lead to system failure, resulting in great financial loss and potentially endangering human life and the environment. For example, disclosure or tampering of critical messages, e.g., of braking or acceleration, in automotive electronic systems can compromise vehicles security and, consequently, safety. Although embedded system security has been addressed in literature [3,4], security issues in distributed embedded system communication, especially the internal communication, have not received as much attention. In [5], several attack scenarios in automotive networks were discussed, and the importance of utilizing cryptography to protect the internal bus communication was highlighted. To provide system-affordable security protection for the internal communication of distributed SSCSs, it is of critical importance to find efficient solutions at early design stages [6,7]. Considering the security issues on mixed-criticality real-time applications, authors of [8] proposed a GA (Genetic Algorithm) based efficient heuristic algorithm to address the system-level optimization of the security-sensitive mixed-criticality real-time applications. However, the underlying energy and reliability issues in the context of distributed SSCSs were not addressed in these works.

Energy efficiency is a fundamental requirement of many embedded systems, not only in battery powered systems. In SSCSs, quick energy depletion or early exhaustion of batteries may compromise security and even cause failure of mission-critical tasks, resulting in unexpected outcomes. One of the most common approaches in energy management is to utilize dynamic voltage

* Corresponding author.
E-mail addresses: weijiang@uestc.edu.cn (W. Jiang), paupop@dtu.dk (P. Pop), dalingog@gmail.com (K. Jiang).

http://dx.doi.org/10.1016/j.micpro.2016.08.002
0141-9331/

& frequency scaling (DVFS) [9]. The effectiveness of DVFS for the modern processors is debatable because of their reduced dynamic power range [10]. However, safety-critical systems typically use older-generation architectures (for which detailed failure data is available). DVFS can have a negative impact on the system reliability, since lowering the voltage leads to exponential increase in the number of transient faults as shown in [11]. Based on such DVFS-related reliability model, efficient reliability management mechanisms have been presented for energy-critical uni-processor systems and distributed systems [12,13]. However, the security issues were seriously overlooked in these works.

In this paper, we are interested in optimizing the energy consumption for security- and reliability-critical applications on DVFS-enabled heterogeneous distributed real-time embedded systems, where both tasks and messages are statically scheduled. We consider both security and reliability issues together for DVFS-enabled real-time embedded systems with a general application model, which is highly different to the work on energy optimization of security-related mixed-criticality real-time applications [8]. Due to the huge complexity of the problem, we propose an efficient Tabu Search-based heuristic that decides on the system variables, i.e., the voltage levels and start time of tasks, the transmission time of messages and their security levels, such that the energy consumption is minimized, while the reliability, real-time, and security requirements of the application are satisfied. The main contributions of this paper are: (1) we introduce a vulnerability based method to quantify the security performance of communications on distributed systems; (2) we address a unified design problem for security- and safety-critical embedded systems which simultaneously considers the security, reliability, energy, and timing requirements; (3) we evaluate the proposed approach on several synthetic benchmarks and one real-life case study (a vehicle cruise controller application) and compare it to other approaches from related work.

The rest of this paper is organized as follows. Section 2 describes the system application and hardware architecture. Sections 3 and 4 present the security model and DVFS-related reliability model we used in this paper, respectively. Section 5 formulates the design optimization problem, and depicts an illustrative example. Our TS-based heuristic and experimental results are given in Sections 6 and 7, respectively. The conclusions of the paper are drawn in Section 8.

2. System architecture

In this section, we describe both application model and hardware architecture for SSCSs. For future reference, we give the most used symbols and abbreviations of this paper in Table 1.

2.1. Application model

We model the application as a directed acyclic task graph $G(\mathbb{V}, \mathbb{E}, \mathbb{M})$. $\mathbb{V}$ is the set of vertices, and each node in $\mathbb{V}$ represents one task P_i. An edge $e_{ij} \in \mathbb{E}$ from P_i to P_j indicates that there is a data dependency between P_i and P_j. $\mathbb{M}$ is the set of messages that need to be transmitted over the communication bus. We assume that the mapping of tasks to processors is given by the designer based on his/her former experience, depending on the concrete execution requirements of the application and usage constraints. The worst case execution time (WCET) C_i of each task P_i is known. In this paper, we focus on the confidential security of messages delivered over the internal communication bus of distributed systems. Thus, messages exchanged by tasks on the same processor are assumed to be secure (e.g., protected by memory isolation, with no communication bus delivery), and their transmission time is ignored. These messages are not explicitly modelled in our application graph. Thus, we only model the messages between tasks on different processors, which are transmitted over the internal communication network. Such messages are denoted with a solid circle on top of the corresponding edges. A task is activated when all its inputs have arrived. Tasks are non-preemptive, and thus, cannot be interrupted during execution. Fig. 1(a) depicts an application consisting of five tasks, P_1 to P_5, and two communication messages, m_1 and m_2, which are supposed to be executed on two processors (N_1, N_2). The mapping and the WCETs (in μs, considering the highest operating mode) are also given in the grey table of Fig. 1(a). The application must finish all executions within D time units after released, known as its global deadline.

We concentrate on tolerating the transient faults of the tasks. Several fault-tolerant techniques have been proposed for this purpose [14], e.g., re-execution, replication and checkpointing with rollback recovery. The techniques have different advantages and disadvantages in terms of performance and energy consumption. In this paper, we have chosen to use active task replication for its simplicity in terms of scheduling. Note that our proposed techniques can be easily extended to support other fault-tolerant techniques. The designer specifies, for each critical task P_i, a number of K_i replicas to ensure execution reliability. For example in Fig. 1(a), task P_2 has two replicas, i.e., $K_2 = 2$. We assume that error detection is performed by each critical task and its replicas. In case a majority voting on the outputs is required, we assume that the designer will add such a voter task to the application graph. The mapping of these replicas is also given by the designer and can be on a different or on the same processor with P_i.

Table 1
Mostly used symbols and abbreviations.

Notation	Definition
G	Application graph
$\mathbb{V}$	The set of tasks
$\mathbb{M}$	The set of messages
$\mathbb{E}$	The set of dependency relations of tasks
$\mathbb{N}$	The set of DVFS-enabled processing nodes
P_i	The i-th task
N_i	The i-th processing node
m_i	The i-th message
D	Deadline of application G
C_i	Execution time of P_i
L_i	Security level of m_i
K_i	The number of replicas of P_i
SI_i	Security input task of m_i
SO_i	Security output task of m_i
RE	System reliability
RE_B	System reliability bound
En	System energy consumption
OV	Overall vulnerability
VD	Vulnerability deficiency
MOV	Maximal overall vulnerability
OV_B	The upper bound of overall vulnerability
VB_i	The vulnerability bound of m_i
η	System reliability degradation
Φ	DVFS mode assignment for all tasks
Υ	Security level assignment for all messages
x	The solution for the optimization problem
LS	List scheduling
TS	Tabu search
DVFS	Dynamic voltage & frequency scaling
SSCSs	Security- and safety-critical systems
TSHO	Tabu search-based hybrid optimization
GHO	Greedy-based hybrid optimizing
MVFS	Mapping, voltage and frequency scaling optimization
EO	Energy optimization
RSO	Reliability and security constrained optimization
VCC	Vehicle cruiser controller
WCET	Worst case execution time
CP	Critical path
MPCP	Modified partial critical path