



Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com



Electronics and Electrical Engineering and Control—Research Article

Hybrid partition- and network-level scheduling design for distributed integrated modular avionics systems

Xuan ZHOU, Huagang XIONG, Feng HE *

School of Electronic and Information Engineering, Beihang University, Beijing 100191, China

Received 19 October 2018; revised 14 December 2018; accepted 17 June 2019

Available online 11 October 2019

KEYWORDS

Distributed integrated modular avionics;
End-to-end delay;
Incremental integration cost;
Maximum satisfiability problem;
Mixed integer programming;
Scheduling algorithms

Abstract Distributed Integrated Modular Avionics (DIMA) develops from Integrated Modular Avionics (IMA) and realizes distributed integration of multiple sub-function areas. Time-triggered network provides effective support for time synchronization and information coordination in DIMA systems. However, inconsistency between processing resources and communication network destroys the time determinism benefiting from partitions and time-triggered mechanism. To ensure such time determinism and achieve guaranteed real-time performance, system design should collectively provide a global communication scheme for messages in network domain and a corresponding execution scheme for partitions in processing domain. This paper firstly establishes a general DIMA model which coordinates partitioned processing and time-triggered communication, and then proposes a hybrid scheduling algorithm using Mixed Integer Programming to produce feasible system schemes. Furthermore, incrementally integrating new functions causes upgrades or reconfigurations of DIMA systems and will generate integration cost. To control such cost, this paper further develops an optimization algorithm based on Maximum Satisfiability Problem and guarantees that the scheduling design for upgraded DIMA systems inherit their original schemes as much as possible. Finally, two typical cases, including a simple fully connected DIMA system case and an industrial DIMA system case, are constructed to illustrate our DIMA model and validate the effectiveness of our hybrid scheduling algorithms.

© 2019 Chinese Society of Aeronautics and Astronautics. Production and hosting by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Distributed Integrated Modular Avionics (DIMA) is an architectural approach consisting of distributed integrated processing resources connected by a mixed-critical communication network such as Time-Triggered Ethernet (TTE).¹ It inherits the flexibility, modularity, reusability and interoperability of Integrated Modular Avionics (IMA), a mature architectural

* Corresponding author.

E-mail address: robinleo@buaa.edu.cn (F. HE).

Peer review under responsibility of Editorial Committee of CJA.



Production and hosting by Elsevier

approach that has been successfully applied in A380 and B787, makes improvements in the aspects of coupling barrier, hierarchical isolation as well as distributed integration, and provides philosophy for the development of future avionics system architecture.²

The DIMA system groups avionics applications running on the core processors into several partitions according to grouping strategies, such as functional coupling or safety criticality, and ensures that the partitions are isolated in time and space through dedicated scheduling. Considering Time-Triggered (TT) mechanism and logical isolation based on virtual links in TTE as a practice of network “partition”, the isolation guarantee mechanism in DIMA systems can extend from processing resource towards network domain. This hierarchical isolation provides support for the complete time determinism in DIMA systems, such as the end-to-end delay and jitter of messages from source partitions to destination partitions. However, without synchronization with processing scheduling, traditional communication scheduling methods of TT messages in TTE using Satisfiability Modulo Theories (SMT),^{3–5} Mixed Integer Programming (MIP),^{6,7} heuristic algorithm^{8,9} or other theoretical methods^{10,11} may cause worse partition-level transmission delay and jitter than those when the messages are communicated through other rate-constrained network, such as Avionics Full Duplex Switched Ethernet (AFDX). The deterioration becomes especially conspicuous in the case that the message sent from its producer partition just misses its time-triggered window. The reason is that the TT message in TTE adheres to a strict scheme and has to wait for the next scheduling round, whereas in AFDX it can be sent once the sending port is idle. Therefore, hybrid partition- and network-level scheduling is crucial for DIMA systems design.

The hybrid scheduling also supports extending the end-to-end communication mechanism from network-level towards partition-level. However, it cannot guarantee that the TT messages originated from higher requirements can be promptly transmitted to obtain timely response, which is an important performance evaluation of Quality of Service (QoS). Furthermore, in practical avionics context, new functions will be added to a DIMA system through its life cycle to satisfy new avionics requirements. This addition leads to system upgrades or reconfiguration and increases the complexity of DIMA system design. The highly coupled hybrid partition- and network-level scheduling method facilitates incremental integration^{12,13} of the DIMA system and helps control such complexity. However, for the new DIMA system incrementally integrated with new functions, the original schedule scheme should be modified to meet all requirements. This may cause recertification cost of DIMA system design, which can be represented by the cost of changes in scheduling parameters for partitions and messages. Consequently, how to accomplish the hybrid partition- and network-level scheduling design with lower transmission delay for DIMA systems and less incremental integration cost for their upgrades becomes an urgent problem to be solved.

Since the essence of scheduling design is to make execution decision for a set of activities under dependency and resource constraints, it can be solved by constraint programming approach which searches for a global state that satisfies numerous constraints at the same time. For different types of constraints, a variety of theoretical methods are proposed, including Integer Linear Programming (ILP), Binary Integer

Programming (BIP), MIP for Linear domains, Boolean Satisfiability Problem (SAT) and its extension such as Maximum Satisfiability Problem (MAXSAT) for Boolean domains, SMT for Mixed domains and many others. In view of the problem mentioned above, the MIP method and MAXSAT method have their own natural advantages. The former optimizes the global state for a certain objective function on the basis of satisfying all constraints, whereas the latter supports searching for a global state that satisfies the most number of constraints when there are no solutions for all constraints. Therefore, they can be applied to the hybrid scheduling design of DIMA system.

In this paper, we study the hybrid partition- and network-level scheduling design for DIMA systems. We firstly establish a general DIMA system model which coordinates the relationship between partitioned processing and time-triggered communication and illustrates the inter-partition communication mechanism on different modules. Based on the model, we design a MIP-based hybrid scheduling algorithm, which includes a set of mandatory constraints for scheduling design and uses partition-level end-to-end transmission delay as the optimization objective. We also propose a new MAXSAT-based optimization scheduling algorithm, which is specially designed for upgraded DIMA systems. It includes the same set of hard constraints with the MIP-based algorithm and introduces a set of soft constraints in order to guarantee the system consistency and control the cost of integrating new functions. Experimental evaluation and analysis with traditional scheduling algorithm are established from the perspective of compactness, complexity and efficiency on the basis of two typical cases, including a simple fully connected DIMA system case and an industrial DIMA system case. The results can provide guidance for practical application.

This paper is organized as follows. After a short introduction about the related work in [Section 2](#), we establish a general DIMA system model in [Section 3](#), and respectively focus on the MIP-based scheduling algorithm for independent DIMA systems as well as the MAXSAT-based scheduling algorithms specifically for their upgrades in [Section 4](#). A simple fully connected DIMA system case and an industrial DIMA system case are used to illustrate our approach in [Section 5](#), where some discussions are included. Finally, conclusions are presented in [Section 6](#).

2. Related works

The research of scheduling design within avionics system contexts has been addressed in literatures, among which many methods are specifically proposed for communication network, typically TTE. Each communication participant in TTE network is synchronized with each other and dispatches its messages according to the predefined scheduling scheme to avoid sharing resource conflicts and guarantee mutual exclusive. Considering this basic principle, Steiner³ designed a basic scheduler and an incremental one for time-triggered multi-hop networks based on a SMT specification with formal scheduling constraints to achieve back-to-back schedule of time-triggered message. Pozo et al.¹⁴ applied a network decomposition approach and synthesized multiple segmented schedules into the final TT-schedule for extremely large networks. They further introduced period constraints to break traditional

hypothetical restrictions that schedule length is shorter than the smallest frame period in Ref. ¹⁵ and combined them together to improve the size and compactness of the solvable schedules in Ref. ¹⁶. Drawing on previous working experience, Beji et al. ¹⁷ proposed a model-driven engineering approach to support the automatic synthesis of programs and resolved the iterative integration of TT flows.

Furthermore, TTE supports mixed-criticality transmission. In order to avoid the starvation problem of Rate-Constrained (RC) messages under back-to-back TT-scheduling, Steiner ⁴ introduced the concept of schedule porosity in which RC messages use the intervals between the adjacent scheduling windows of two TT messages. He also designed three general implementation methods, including priori variation, posteriori variation as well as interpretation. Tamas-Selicean et al. ⁸ further proposed a Tabu Search-based heuristic optimization scheduling strategy that broke the restricts of periodic even-spaced intervals in TT-schedule ⁴ to minimize the end-to-end delay of RC messages. They also constructed a more comprehensive method considering message fragmenting and packaging as well as virtual link routing in Ref. ⁹. Zhang et al. ⁵ improved traditional static SMT-based TT-scheduling methods ^{3,4} to obtain the optimal discrete intervals, where dynamic Weighted Round Robin and Programming Porosity scheduling for heterogeneous RC messages were executed to ensure high resource utilization, low packets loss rate and short end-to-end delay in TTE. Besides, various scheduling methods are designed for other communication networking solutions such as AFDX ^{18,19}, Audio Video Bridging (AVB) ²⁰ and Time Sensitive Networks (TSN). ²¹

Focusing on the arrangement of processing resources in IMA systems, many scholars have been studying how to conduct scheduling in consideration of partition allocation and task assignment effectively. For example, Lu et al. ²² allowed a partition to be interrupted by other partitions with minimal interruptions and investigated a strictly periodic and preemptive partition scheduling strategy based on the Particle Swarm Optimization. This strategy enhanced the schedulability of complex partition sets, whereas it cannot break the restriction of only harmonic periods or even favor grouping harmonic ones on the same processor. Sheikh et al. ²³ presented an exact MIP formulation and a heuristic partition scheduling algorithm inspired from Game Theory to suit all kinds of periods in an avionics context, including harmonic and non-harmonic. Considering application-level requirements with respect to the allowed safety margins, García-Valls et al. ²⁴ integrated the middleware communication overhead into the partition scheduling using IMA systems under Future Airborne Capability Environment (FACE), which defines the software computing environment and interfaces for easing integration in avionics systems. Most methods firstly performed scheduling and then completed end-to-end delay analysis, and their scheduling scheme occasionally unsatisfied delay constraints, thus resulting in rescheduling. Aiming at this problem, Deroche et al. ²⁵ managed the static allocation of partitions altogether with a guaranteed applicative end-to-end delay.

Additionally, multi-core scheduling design as an independent topic has also been widely concerned about in recent years. Given shared platform resources in cluster-based multi-core systems, Giannopoulou et al. ²⁶ proposed a mixed-criticality scheduling policy. Their policy enforced global tim-

ing isolation between different criticality-level applications and therefore sufficed to provide certification attributes. Targeted partitions with both harmonic and non-harmonic periods, Chen et al. ²⁷ put forward an equilibrium-based heuristic scheduling method, which determined schedulability and obtained feasible solutions rapidly. Moreover, the IMA systems may be mistakenly declared unschedulable under the assumption of worst-case contentions which can never occur in practice. Therefore, Rouxel et al. ²⁸ presented two contention-aware scheduling using an ILP formulation and a heuristic, improving the schedule makespan by 19% on average. Some other schedulability analysis and verification methods for real-time systems can also be extended to IMA systems, such as the Petri nets based parametric model analysis ²⁹ and the temporal logic based satisfiability checking ³⁰ for cyber-physical systems, as well as the utilization based and rate monotonic analysis for middleware-based service oriented distributed real-time systems. ³¹

The isolated scheduling of communication network or processing resources may severely limit the overall performance and feasibility of whole avionics systems. Consequently, a handful of literatures have studied hybrid scheduling on both perspectives, which is also the focus of our research. Hu et al. ³² designed an Unfixed Start Time algorithm to schedule tasks and messages in a flexible way. They also applied rescheduling and backtracking methods to modify the algorithm in order to tolerant assignment conflicts and improve schedulability. Mathematical methods, especially SMT and MIP, are also commonly used in industrial-scale application-level combined scheduling design. For example, Zhang et al. ⁶ supported one or multiple optimization objectives such as application response time, end-to-end delay and their combinations, Craciunas et al. ^{7,33} emphasized incremental scheduling approach based on utilization demand bound analysis for asynchronous periodic tasks, Blikstad et al. ³⁴ introduced a restriction that the slot in each communication module sent or received at most one message.

At present, the research of DIMA systems is still at the beginning. Most relevant methods are concerned about system optimal design ³⁵⁻³⁸ and real-time performance analysis, ^{39,40} whereas few are related to scheduling design, let alone the hybrid handling of processing resources and communication network. As DIMA is not simply equivalent to multi-IMA, traditional scheduling methods, such as constraint programming approach and heuristic algorithms, cannot be directly applied to DIMA systems. This paper deals with a novel hybrid scheduling method using partition and message frames as scheduling entities, so as to avoid the loss of time determinism caused by asynchronous processing and communication in DIMA systems. It also studies how to design scheduling scheme at a relative high speed with lower transmission delay for DIMA systems and less incremental integration cost for their upgrades.

3. DIMA system model

The DIMA system is essentially a distributed IMA system that interconnects and communicates through time-triggered network based on the accurate global clock synchronization. Its typical features are partitioned processing and time-triggered communication.

Partitioned processing is embodied in the real-time operating system standard defined in ARINC653. It distributes different avionics applications into corresponding partitions according to their requirements and coupling relations. It also ensures the independence of applications through reasonable partition scheduling arrangements. The periodic pattern of all partitions $\mathcal{P}_i = \{P_{i,1}, \dots, P_{i,j}, \dots, P_{i,N_p}\}$ executing on the module \mathcal{M}_i is called MAJOR Frame (MAF). The length of MAF usually equals to the Least Common Multiple (LCM) of all partition periods. As shown in Fig. 1, the partition $P_{i,j}$ is round-robin activated in

$$\mathcal{I}_{i,j} = \bigcup_{k=1}^{n_{i,j}} \left(P_{i,j}^{\text{offset}} + k \times P_{i,j}^{\text{period}}, P_{i,j}^{\text{offset}} + P_{i,j}^{\text{length}} + k \times P_{i,j}^{\text{period}} \right) \quad (1)$$

within the MAF of \mathcal{M}_i , where

$$n_{i,j} = \frac{\text{LCM}_{1 \leq x \leq N_p} (P_{i,x}^{\text{period}})}{P_{i,j}^{\text{period}}} \quad (2)$$

$P_{i,j}^{\text{offset}}$ denotes the offset of $P_{i,j}$ relative to the initial time of MAF, $P_{i,j}^{\text{period}}$ its execution periods, and $P_{i,j}^{\text{length}}$ its duration time.

Time-triggered communication is typically implemented through TTE, which is highly suitable for safety-critical applications due to its support for mixed-criticality communication, such as TT, RC and Best-Effort (BE) message. TT messages has the highest-level priority and can avoid contention with other messages on output ports through prior-scheduling; asynchronous RC message has a middle-level priority and guarantees its transmission with predefined bandwidth; BE message has the lowest-level priority and supports traditional Ethernet communications. TTE can be described as an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ as shown in Fig. 2, where \mathcal{V} denotes network nodes including End Systems (ES) and SWitches (SW) and \mathcal{E} full duplex physical links. To facilitate the representation of network communication, a set of unidirectional dataflow links l_r from v_m to v_n is defined as

$$\begin{aligned} \mathcal{L} &= \{l_1, \dots, l_r, \dots, l_{N_l}\} \\ &= \{[ES_1, SW_1], \dots, [v_m, v_n], \dots, [SW_1, ES_3]\} \end{aligned} \quad (3)$$

where v_m and v_n can be ES or SW. As the executed instants of applications in the corresponding end systems, tasks send messages from one sender to one or multiple receivers via the dataflow path $dp_s = [l_p, \dots, l_x, \dots, l_q] \in \mathcal{DP}$. A feasible communication scheme for a set of periodic message frames $\mathcal{F} = \{f_1, \dots, f_k, \dots, f_{N_f}\}$ can be seen around the links in

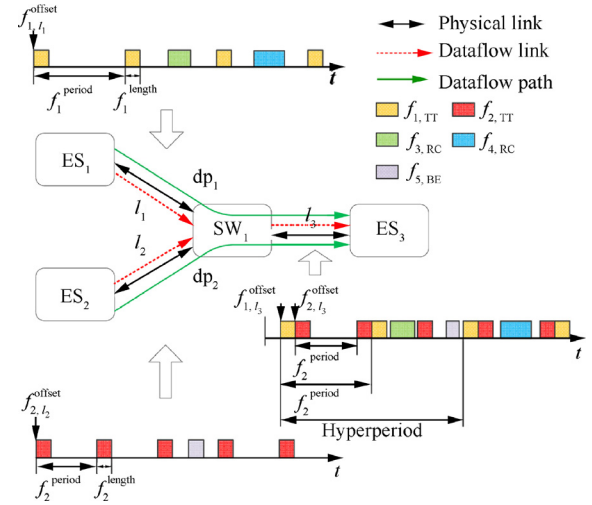


Fig. 2 Diagram of TTE and a feasible communication scheme on it.

Fig. 2. For frame f_k , f_k^{period} denotes its transmission period, f_k^{length} means its processing time and can be calculated on the basis of the frame size f_k^{size} and link rate C_r according to the conversion equation

$$f_k^{\text{length}} = \frac{f_k^{\text{size}}}{C_r} \quad (4)$$

and f_k^{offset} shows its offset on link l_r relative to the time base of TTE.

A simple DIMA system model is shown in Fig. 3. The partitions are resided into end systems, which are considered as basic modules in DIMA, and can communicate with each other on the basis of distribution middleware, such as Data Distribution Service (DDS). Different modules are scattered throughout the fuselage and clustered into distributed Integrated Modular Processing Regions (IMPR) according to application requirements and their coupling relations. TTE is applied as the universal network to connect different IMPRs with Backbone Switches (BSW) and different Modules in the same IMPR with Access Switches (ASW). Aiming at the hybrid partition- and network-level scheduling problem to be solved in this paper, we weaken the implementation of tasks and emphasize time-triggered communication between partitions on different modules based on the assumption of global clock synchronization; that is to say, all scheduling offsets of

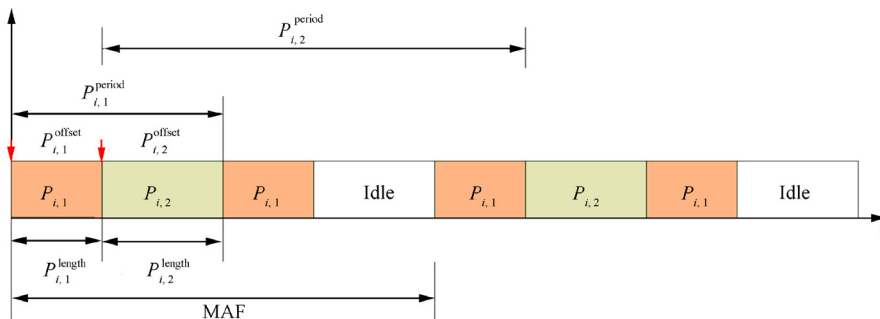


Fig. 1 Partitions are round-robin activated within the MAF.

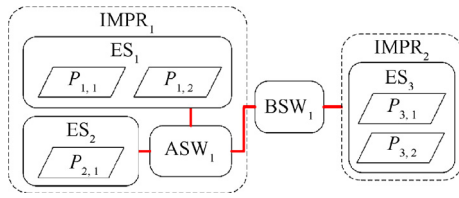


Fig. 3 A simple DIMA system model.

partitions and frames are relative to a common time point. Typically, the synchronization can be realized by periodically calling Permanence Function (PF) and Compression Function (CF) assisted by Protocol Control Frame (PCF) in TTE.

The communication between different modules is shown in Fig. 4, where f_k^{ETE} denotes the partition-level end-to-end delay of f_k and f_k^{\max} its maximum acceptable delay.

4. Schedule algorithms

Note that in [Section 3](#), the DIMA system model (see [Fig. 3](#)) and its inter-partition communication mechanism on different modules (see [Fig. 4](#)) directly follow the notations defined in the partition processing (see [Fig. 1](#)) and time-triggered communication (see [Fig. 2](#)) to illustrate the corresponding relationship. In this section, in order to facilitate the representation of scheduling constraints and clearly present the hierarchies and affiliations in the DIMA system, we redefine the notations used in our schedule algorithms and explain their meanings in [Table 1](#).

4.1. MIP-based hybrid scheduling algorithm

Hybrid partition- and network-level scheduling design for DIMA systems described in [Section 3](#) is actually seeking feasi-

ble solutions to offsets of partitions and frames under a series of temporal constraints. At present, some scholars, such as Steiner³ and Craciunas-Valls,²⁹ have proposed widely accepted constraints for software tasks and communication messages. Based on their pioneering work, we define a set of constraints for the DIMA system, and apply MIP method to construct a formal specification of these constraints. We also design an optimization function to guarantee the swift response of TT frames from produced by their source partitions.

4.1.1. Basic constraints

Obviously, the offset for any partition or frame to be scheduled should be non-negative and its corresponding time window should not exceed its periods. These are formally expressed by the following set of constraints.

$$\begin{aligned} & \forall \text{IMPR}_i \in \mathcal{IMP}\mathcal{R}, \forall \text{ES}_{ij} \in \mathcal{E}\mathcal{S}_i, \forall P_{i,j,k} \in \mathcal{P}_{ij}; \forall f_p \in \mathcal{F}, \\ & \forall \text{dp}_{p,q} \in \mathcal{DP}_p, \forall l_{p,q,r} \in \mathcal{L}_{p,q} : \\ & \{P_{i,j,k} \cdot \text{offset} \geq 0\} \wedge \{f_{p,q,r} \cdot \text{offset} \geq 0\} \wedge \\ & \{P_{i,j,k} \cdot \text{offset} + P_{i,j,k} \cdot \text{length} \leq P_{i,j,k} \cdot \text{period}\} \\ & \wedge \{f_{p,q,r} \cdot \text{offset} + f_p \cdot \text{length} \leq f_p \cdot \text{period}\} \end{aligned} \quad (5)$$

4.1.2. Non-overlapping execution constraints

There should be at most one active partition at a given time within the same end system, which can be reflected by an “Either-Or” statement. In other words, one partition either completes its execution before the beginning of the other partition or begins after the end of the other. Based on the alternative constraints method³⁸ which has been successfully used in Refs. 7,8, the “Either-Or” statement can be converted into the following inequities to conform MIP paradigm by introducing a series of binary decision variables $\{\alpha_{i,j,k_1,k_2,a,b}\}$.

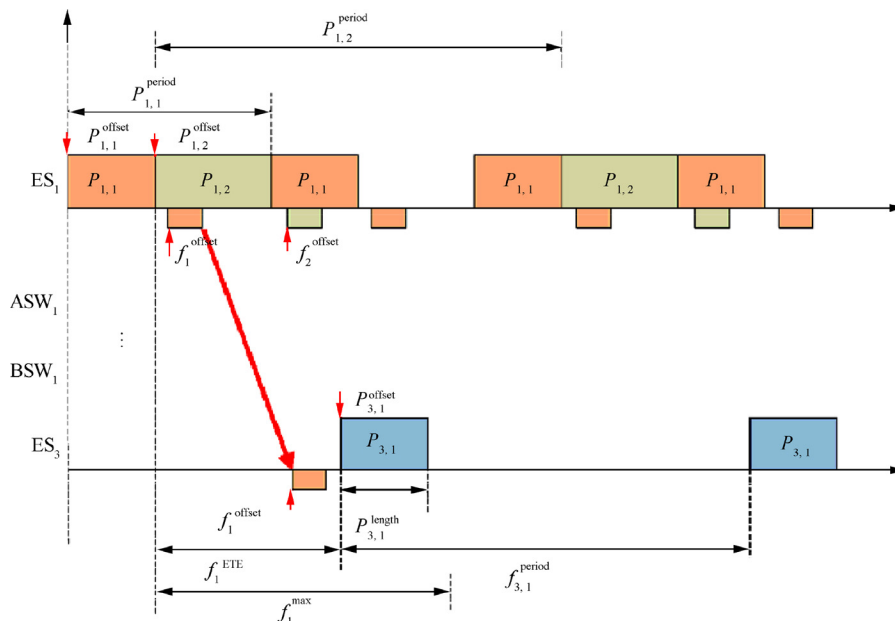


Fig. 4 Communication between different modules in DIMA system.

Table 1 Notations.

Notation	Meaning
IMPR	The set of all IMPRs in the DIMA system
IMPR_i	An IMPR in the DIMA system
\mathcal{ES}_i	The set of all End Systems in IMPR_i
$\text{ES}_{i,j}$	An End System in IMPR_i
$\text{MAF}_{i,j}$	The Major Time Frame of $\text{ES}_{i,j}$, whose length equals to the least common multiple of all $P_{i,j,k}$ · period in $\text{ES}_{i,j}$
$\mathcal{P}_{i,j}$	The set of all Partitions in $\text{ES}_{i,j}$
$P_{i,j,k}$	A partition in $\text{ES}_{i,j}$
$P_{i,j,k}$ · offset	The offset of $P_{i,j,k}$, which characterizes the activation time of $P_{i,j,k}$ relative to the global clock
$P_{i,j,k}$ · length	The length of time window assigned to $P_{i,j,k}$ for its execution
$P_{i,j,k}$ · period	The period of $P_{i,j,k}$
\mathcal{F}	The set of all TT message frames in the DIMA system
f_p	A TT message frame in the DIMA system
f_p · length	The length of time window assigned for f_p
f_p · period	The period of f_p
Src_p	The source partition that produces f_p
$\text{Des}_{p,q}$	The destination partition that consumes f_p through $\text{dp}_{p,q}$
\mathcal{DP}_p	The set of all dataflow paths of f_p
$\text{dp}_{p,q}$	A dataflow path of f_p
$\mathcal{L}_{p,q}$	The ordered set of all dataflow links along $\text{dp}_{p,q}$
$l_{p,q,r}$	The r-th dataflow link along $\text{dp}_{p,q}$
$l_{p,q,\text{first}}$	The first dataflow link along $\text{dp}_{p,q}$
$l_{p,q,\text{last}}$	The last dataflow link along $\text{dp}_{p,q}$
\mathcal{L}	The ordered set of all dataflow links in the DIMA system
l_m	A dataflow link in the DIMA system
\mathcal{F}_m	The set of all message frames transmitted through l_m
HP_m	The hyper-period of all frames transmitted through l_m
$f_{p,q,r}$ · offset	The offset of f_p on the dataflow link $l_{p,q,r}$
min_{tech}	The minimal technological latency at the switch
max_{tech}	The maximal technological latency at the switch
$\text{max}_{\text{delay}}^p$	The allowed maximal end-to-end delay of the message frame f_p
$\mathcal{P}'_{i,j}$	The set of all Partitions in $\text{ES}_{i,j}$ after integrating new functions
$\mathcal{P}'_{i,j,k}$ · offset	The upgraded offset of $P_{i,j,k}$ after integration
\mathcal{F}'	The set of all TT message frames after integration
$f'_{p,q,r}$ · offset	The upgraded offset of f_p on the dataflow link $l_{p,q,r}$ after integration

$$\begin{aligned}
& \forall a \in \left\{ x \in \mathbb{N} \mid 0 \leq x \leq \left(\frac{\text{MAF}_{i,j}}{P_{i,j,k_1} \cdot \text{period}} - 1 \right) \right\}, \\
& \forall b \in \left\{ y \in \mathbb{N} \mid 0 \leq y \leq \left(\frac{\text{MAF}_{i,j}}{P_{i,j,k_2} \cdot \text{period}} - 1 \right) \right\} \\
& \forall \text{IMPR}_i \in \mathcal{IMPR}, \forall \text{ES}_{i,j} \in \mathcal{ES}_i, \\
& \forall P_{i,j,k_1}, P_{i,j,k_2} \in \mathcal{P}_{i,j} (k_1 \neq k_2), \exists \alpha_{i,j,k_1,k_2,a,b} \in \{0, 1\} : \\
& \{ P_{i,j,k_1} \cdot \text{offset} + a \times P_{i,j,k_1} \cdot \text{period} + \alpha_{i,j,k_1,k_2,a,b} \times \theta \\
& \geq P_{i,j,k_2} \cdot \text{offset} + P_{i,j,k_2} \cdot \text{length} + b \times P_{i,j,k_2} \cdot \text{period} \} \wedge \\
& \{ P_{i,j,k_2} \cdot \text{offset} + b \times P_{i,j,k_2} \cdot \text{period} + (1 - \alpha_{i,j,k_1,k_2,a,b}) \times \theta \\
& \geq P_{i,j,k_1} \cdot \text{offset} + P_{i,j,k_1} \cdot \text{length} + a \times P_{i,j,k_1} \cdot \text{period} \}
\end{aligned} \tag{6}$$

where θ is large enough to ensure that the first inequality is permanently established for $\alpha_{i,j,k_1,k_2,a,b} = 1$, and the second for $\alpha_{i,j,k_1,k_2,a,b} = 0$. That is to say, only one of the two inequalities is true in any circumstances. Typically, the value of θ can be $\text{MAF}_{i,j}$.

4.1.3. Contention-free transmission constraints

The scheduling design for frames in DIMA system should avoid conflict among multiple TT messages on the same dataflow link. Therefore, the end systems or switches can respectively dispatch or relay the succeeding frame only after the preceding frame is processed. This can be specified similar to non-overlapping execution constraints for partitions by introducing a series of binary decision variables $\{\beta_{m,p_1,p_2,a,b}\}$.

$$\begin{aligned}
& \forall a \in \left\{ x \in \mathbb{N} \mid 0 \leq x \leq \left(\frac{\text{HP}_m}{f_{p_1} \cdot \text{period}} - 1 \right) \right\}, \\
& \forall b \in \left\{ y \in \mathbb{N} \mid 0 \leq y \leq \left(\frac{\text{HP}_m}{f_{p_2} \cdot \text{period}} - 1 \right) \right\} \\
& \forall l_m \in \mathcal{L}, \forall f_{p_1}, f_{p_2} \in \mathcal{F}_m (p_1 \neq p_2) \Rightarrow l_{p_1,q_1,r_1} \\
& = l_{p_2,q_2,r_2} = l_m, \exists \beta_{m,p_1,p_2,a,b} \in \{0, 1\} : \\
& \left\{ f_{p_1,q_1,r_1} \cdot \text{offset} + a \times f_{p_1} \cdot \text{period} + \beta_{m,p_1,p_2,a,b} \times \varphi \right. \\
& \geq f_{p_2,q_2,r_2} \cdot \text{offset} + f_{p_2} \cdot \text{length} + b \times f_{p_2} \cdot \text{period} \} \wedge \\
& \left\{ f_{p_2,q_2,r_2} \cdot \text{offset} + b \times f_{p_2} \cdot \text{period} + (1 - \beta_{m,p_1,p_2,a,b}) \times \varphi \right. \\
& \geq f_{p_1,q_1,r_1} \cdot \text{offset} + f_{p_1} \cdot \text{length} + a \times f_{p_1} \cdot \text{period} \}
\end{aligned} \tag{7}$$

where the value of φ can be HP_m .

4.1.4. Bound-restricted relay constraints

Taking into account the relay of the switches, the scheduling offset on two adjacent dataflow links within the same dataflow path of the frame should be well-timed. The scheduling offset on the succeeding dataflow link should consider not only the end time of the frame on the preceding dataflow link but also other two aspects. (A) The minimal technological latency min_{tech} at the switch which limits the lower-bound between two adjacent scheduling offsets. (B) The maximal technological latency max_{tech} at the switch which limits the upper-bound between two adjacent scheduling offsets. Note that min_{tech} and max_{tech} just consider the restriction of switch hardware, such as its inherent switching performance, memory size as well as buffer depth, and can be described in time units through quantifying conversion.

$$\begin{aligned}
& \forall f_p \in \mathcal{F}, \forall \text{dp}_{p,q} \in \mathcal{DP}_p, \forall l_{p,q,r}, l_{p,q,r+1} \in \mathcal{L}_{p,q} : \\
& \{ f_{p,q,r+1} \cdot \text{offset} - f_{p,q,r} \cdot \text{offset} \leq \text{max}_{\text{tech}} \} \wedge \\
& \{ f_{p,q,r+1} \cdot \text{offset} - f_{p,q,r} \cdot \text{offset} - f_{p,q,r} \cdot \text{length} \geq \text{min}_{\text{tech}} \}
\end{aligned} \tag{8}$$

4.1.5. Simultaneous relay constraints

In the case of broadcasting, the TT frame may be sent from one producer partition to multiple consumer partitions through different dataflow paths. Due to the synchronous relay principle of switches, the scheduling offsets on the same dataflow links and the first detached dataflow link for any pair of dataflow paths should be matching.

$$\begin{aligned}
& \forall f_p \in \mathcal{F}, \forall dp_{p,q_1}, dp_{p,q_2} \in \mathcal{DP}_p(q_1 \neq q_2) \\
& \Rightarrow l_{p,q_1, \text{first}} = l_{p,q_2, \text{first}}, \dots, l_{p,q_1, r} = l_{p,q_2, r}, l_{p,q_1, r+1} \neq l_{p,q_2, r+1} : \\
& \left\{ f_{p,q_1, \text{first}} \cdot \text{offset} = f_{p,q_2, \text{first}} \cdot \text{offset} \right\} \wedge \dots \\
& \wedge \left\{ f_{p,q_1, r} \cdot \text{offset} = f_{p,q_2, r} \cdot \text{offset} \right\} \\
& \left\{ f_{p,q_1, r+1} \cdot \text{offset} = f_{p,q_2, r+1} \cdot \text{offset} \right\}
\end{aligned} \quad (9)$$

4.1.6. End-to-end delay constraints

This constraint is introduced to ensure the timely transmission of the TT frame f_p from its producer partition to its consumer partitions with the allowed maximum end to end delay \max_{delay}^p , which is predefined according to application requirements.

$$\begin{aligned}
& \forall f_p \in \mathcal{F}, \forall dp_{p,q} \in \mathcal{DP}_p : \\
& \left\{ \text{Des}_{p,q} \cdot \text{offset} - \text{Src}_p \cdot \text{offset} - \text{Src}_p \cdot \text{length} \leq \max_{\text{delay}}^p \right\}
\end{aligned} \quad (10)$$

4.1.7. System compatibility constraints

The compatibility between partitions and frames is of great importance for hybrid partition- and network-level scheduling. According to the DIMA system model established in Section 3, the TT message frame carries the information generated by its producer partition, through a sequence of switches, including access switches and possibly backbone switches for inter-IMPR communications, to reach its consumer partition. The consumer partition receives the message from its producer partition and processes it to complete system tasks and implement system functions. Therefore, the scheduling offset of the TT frame on its first dataflow link should be after the completely execution of its source partition, while the end of its processing on the last dataflow link should be before the execution of its destination partition.

$$\begin{aligned}
& \forall f_p \in \mathcal{F}, \forall dp_{p,q} \in \mathcal{DP}_p : \\
& \left\{ \text{Src}_p \cdot \text{offset} + \text{Src}_p \cdot \text{length} \leq f_{p,q, \text{first}} \cdot \text{offset} \right. \\
& \leq \text{Src}_p \cdot \text{offset} + \text{Src}_p \cdot \text{length} + \text{Src}_p \cdot \text{period} \left. \right\} \wedge \\
& \left\{ f_{p,q, \text{last}} \cdot \text{offset} + f_p \cdot \text{length} \leq \text{Des}_{p,q} \cdot \text{offset} \right. \\
& \leq f_{p,q, \text{last}} \cdot \text{offset} + f_p \cdot \text{length} + f_p \cdot \text{period} \left. \right\}
\end{aligned} \quad (11)$$

4.1.8. Optimization function

In end-to-end delay constraints, we limit the end-to-end delay of a frame f_p from produced to consumed into an upper bound \max_{delay}^p . In order to guarantee that TT messages matched with higher requirements can be promptly transmitted to obtain timely response, we further define the optimization objective function as the following formulation.

$$\min_{\forall f_p \in \mathcal{F}, \forall dp_{p,q} \in \mathcal{DP}_p} (\text{Des}_{p,q} \cdot \text{offset} - \text{Src}_p \cdot \text{offset} - \text{Src}_p \cdot \text{length}) \quad (12)$$

4.2. MAXSAT-based hybrid scheduling algorithm

MIP-based hybrid scheduling algorithm focuses on the scheduling design for the independent DIMA system. However, for the upgraded DIMA system, which is developed from

the corresponding original DIMA system by integrating new partitions and messages, the MIP-based algorithm cannot handle the system consistency well. In order to better suit the upgraded DIMA system, this section constructs an optimization scheduling algorithm on the basis of Weighted Partial MAXSAT approach. In addition to the hard constraints specified in the MIP-based algorithm, we construct a set of soft constraints associated with weights. These soft constraints are not necessarily satisfied, but can be used to ensure that the upgraded DIMA system maintains the scheduling invariance with its original system as much as possible through maximizing total weight of satisfied constraints, and therefore minimize the cost of integration.

The hard constraints in this algorithm are in accordance with the constraints specified in the MIP-based hybrid scheduling algorithm. Since MAXSAT approach supports logical clauses, the non-overlapping execution constraints and the contention-free transmission constraints can be described as the following ‘‘Either-Or’’ statements. This equivalent transformation can avoid the introduction of too many binary decision variables, such as $\{\alpha_{i,j,k_1,k_2,a,b}\}$ and $\{\beta_{m,p_1,p_2,a,b}\}$, and effectively reduce the number of variables involved in the scheduling design. Other constraints can be directly dragged here.

4.2.1. Non-overlapping execution constraints

$$\begin{aligned}
& \forall a \in \left\{ x \in \mathbb{N} | 0 \leq x \leq \left(\frac{\text{MAF}_{ij}}{P_{i,j,k_1} \cdot \text{period}} - 1 \right) \right\}, \\
& \forall b \in \left\{ y \in \mathbb{N} | 0 \leq y \leq \left(\frac{\text{MAF}_{ij}}{P_{i,j,k_2} \cdot \text{period}} - 1 \right) \right\} \\
& \forall \text{IMPR}_i \in \mathcal{IMPR}, \forall \text{ES}_{i,j} \in \mathcal{ES}_i, \forall P_{i,j,k_1}, P_{i,j,k_2} \in \mathcal{P}_{i,j}(k_1 \neq k_2) : \\
& \left\{ P_{i,j,k_1} \cdot \text{offset} + a \times P_{i,j,k_1} \cdot \text{period} \right. \\
& \geq P_{i,j,k_2} \cdot \text{offset} + P_{i,j,k_2} \cdot \text{length} + b \times P_{i,j,k_2} \cdot \text{period} \left. \right\} \vee \\
& \left\{ P_{i,j,k_2} \cdot \text{offset} + b \times P_{i,j,k_2} \cdot \text{period} \right. \\
& \geq P_{i,j,k_1} \cdot \text{offset} + P_{i,j,k_1} \cdot \text{length} + a \times P_{i,j,k_1} \cdot \text{period} \left. \right\}
\end{aligned} \quad (13)$$

4.2.2. Contention-free transmission constraints

$$\begin{aligned}
& \forall a \in \left\{ x \in \mathbb{N} | 0 \leq x \leq \left(\frac{\text{HP}_m}{f_{p_1} \cdot \text{period}} - 1 \right) \right\}, \\
& \forall b \in \left\{ y \in \mathbb{N} | 0 \leq y \leq \left(\frac{\text{HP}_m}{f_{p_2} \cdot \text{period}} - 1 \right) \right\} \\
& \forall l_m \in \mathcal{L}, \forall f_{p_1}, f_{p_2} \in \mathcal{F}_m(p_1 \neq p_2) \Rightarrow l_{p_1,q_1,r_1} = l_{p_2,q_2,r_2} = l_m : \\
& \left\{ f_{p_1,q_1,r_1} \cdot \text{offset} + a \times f_{p_1} \cdot \text{period} \right. \\
& \geq f_{p_2,q_2,r_2} \cdot \text{offset} + f_{p_2} \cdot \text{length} + b \times f_{p_2} \cdot \text{period} \left. \right\} \vee \\
& \left\{ f_{p_2,q_2,r_2} \cdot \text{offset} + b \times f_{p_2} \cdot \text{period} \right. \\
& \geq f_{p_1,q_1,r_1} \cdot \text{offset} + f_{p_1} \cdot \text{length} + a \times f_{p_1} \cdot \text{period} \left. \right\}
\end{aligned} \quad (14)$$

4.2.3. Soft constraints

The following soft constraints contain partition consistency and frame consistency. For the upgraded DIMA system, the

consistency of partitions should be ensured as much as possible in order to control the integration cost caused by recertification of original partitions. Namely, the scheduling offsets of original partitions should try to remain unchanged.

$$\forall \text{IMPR}_i \in \text{IMPR}, \forall \text{ES}_{i,j} \in \text{ES}_i, \forall P_{i,j,k} \in \mathcal{P}_{i,j} \cap \mathcal{P}'_{i,j} : \quad (15)$$

$$\{P_{i,j,k} \cdot \text{offset} = P'_{i,j,k} \cdot \text{offset}\}$$

Similar to partitions, frames should adopt similar method to keep their original parameters.

$$\forall f_p \in \mathcal{F} \cap \mathcal{F}', \forall d_{p,q} \in \mathcal{DP}_p, \forall l_{p,q,r} \in \mathcal{L}_{p,q} : \quad (16)$$

$$\{f_{p,q,r} \cdot \text{offset} = f'_{p,q,r} \cdot \text{offset}\}$$

5. Experiments and evaluations

In this section, the performance of our proposed algorithms is evaluated by comparison with traditional scheduling algorithm, which generally uses SMT theory for system scheduling design. In order to make the comparison experiments credible and effective, we take SMT-based algorithms as benchmark, including traditional network-level scheduling algorithm (TSA)^{3,13,14,33} and its derivative scheduling algorithm with the introduction of hybrid concept (DSA). We also develop a multi-algorithm scheduling tool by integrating Z3 SMT solver⁴¹ v4.5.0, Gurobi MIP solver⁴² v7.5.1 and vZ MAXSAT solver⁴³ in Eclipse platform v4.5.2 on a 64 bit 4-core 3.20 GHz Intel Core i5 PC with 8 GB RAM. Note that to avoid the MIP solver running timeouts, we set up a callback to monitor optimization progress and implement a custom termination once at least one of the following two conditions has been satisfied. (A) The optimality gap is less than 10%. (B) At least 1×10^4

nodes have been explored, and a feasible solution has been found.

Two test cases of different scales, including a simple fully connected DIMA system case and an industrial DIMA system case, are designed to illustrate the scheduling results and performance. Additionally, we emphasize and analyze the comparison of benchmark algorithms and our proposed MIP-based Hybrid Scheduling Algorithm (HSA) for original DIMA systems, and further comparison of benchmark algorithms, HSA and MAXSAT-based Optimization Scheduling Algorithm (OSA) for upgraded DIMA systems.

5.1. Configuration of test cases

5.1.1. A fully connected DIMA system case

We use the topology shown in Fig. 3 as the original topology of the fully connected DIMA system case, and then add a partition $P_{2,2}$ with a message f_6 , highlighted by red fonts, to form the upgraded DIMA system, as shown in Fig. 5. In this case, the message frame is produced by its corresponding source partition in IMPR_1 and consumed by $P_{3,1}$ or $P_{3,2}$ through the transmission path indicated by the directed lines. For example, the message frame f_1 is produced by $P_{1,1}$ and consumed by $P_{3,1}$ in chronological order through l_1 , l_3 and l_4 .

The essential configuration parameters of partitions and message frames associated with the hybrid scheduling are given in Table 2, including period, length of allocated time window, and recertification cost. The recertification cost of each partition is set to 1, 3 or 5 units depending on the number of messages that the partition produces or consumes. Besides, all message frames have an identical recertification cost on each link of their transmission paths. Other constant parameters related to the hybrid scheduling are set as follows: $\min_{\text{tech}} = 1$ ms, and $\max_{\text{tech}} = 20$ ms.

5.1.2. An industrial DIMA system case

The industrial DIMA system case consists of 4 Integrated Modular Processing Regions and 3 interconnected backbone switches as shown in Fig. 6. Each IMPR contains 1 access switch, IMPR_1 and IMPR_4 contain 3 End Systems, and IMPR_2 and IMPR_3 contain 2 End Systems. Moreover, there are 3–5 partitions in each ES.

From the perspective of communication with other ES in the same IMPR or with other IMPRs, the partitions naturally have the directional property, i.e. either source or destination, which is labeled with red or black borders respectively in Fig. 6. And the configuration of this property in case 2 ensures that each ES can send and receive messages frames. In addition, the periods of partitions are selected from a set of periods {20 ms, 40 ms, 80 ms}, filled with orange, green and blue

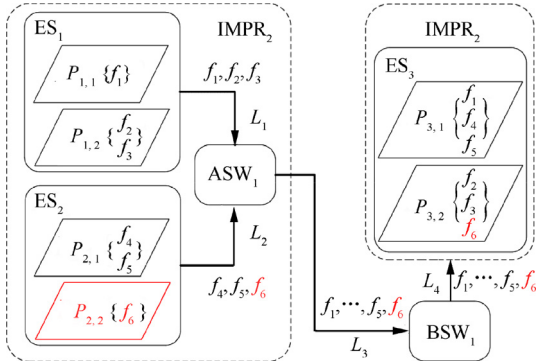


Fig. 5 A fully connected DIMA system case.

Table 2 Configuration parameters of partitions and message frames in Case 1.

Partition	Period (ms)	Length (ms)	Cost (unit)	Frame	Period (ms)	Length (ms)	max _{delay} (ms)	Cost (unit)
$P_{1,1}$	100	20	1	f_1	100	3	200	1
$P_{1,2}$	200	60	3	f_2	200	3	200	1
$P_{2,1}$	100	30	3	f_3	200	3	200	1
$P_{2,2}$	200	60	/	f_4	100	3	200	1
$P_{3,1}$	100	30	5	f_5	100	3	200	1
$P_{3,2}$	200	50	3	f_6	200	3	200	/

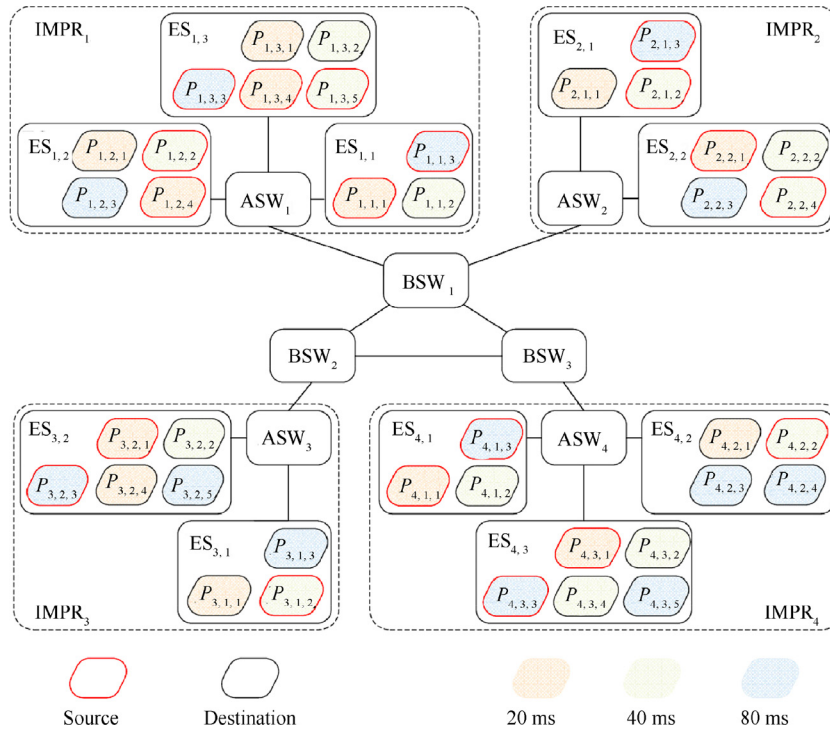


Fig. 6 An industrial DIMA system case.

Table 3 Bandwidth utilization of backbone network as well as the configuration ID for different experiments in Case 2.

	Group 1 (50)			Group 2 (100)			Group 3 (200)		
	Average (%)	Max (%)	ID	Average (%)	Max (%)	ID	Average(%)	Max (%)	ID
A	1.59	2.01	1	2.72	3.38	5	5.36	6.90	9
B	3.93	4.88	2	6.68	8.32	6	13.2	16.51	10
C	6.28	7.76	3	10.62	13.23	7	21.05	26.12	11
D	8.63	10.64	4	14.57	18.14	8	28.89	35.71	12

respectively in Fig. 6. And the assignment of $P_{i,j,k}$ period ensures that each IMPR can send and receive various periodic message frames. The setting of $P_{i,j,k}$ length guarantees the balanced proportion of non-idle time in the major frames for all end systems. Furthermore, considering that the recertification cost of partitions is usually higher than that of message frames, we set the recertification cost of partitions equivalent to 3.

In order to compare the scheduling performance of the three algorithms with different network bandwidth utilization, we conduct three groups of experiments, respectively tagged as Group I (50 messages), Group II (100 messages) and Group III (200 messages). We also set four possible length ranges of message frames for each group, including Class A (64–427Bytes), Class B (428–791Bytes), Class C (792–1155Bytes), and Class D (1156–1517Bytes). Considering that the typical link rate of TTE is 100 Mbps, the length ranges of time windows assigned for message frames can be Class A (5.12–34.16μs), Class B (34.24–63.28μs), Class C (63.36–92.4μs), and Class D (92.48–121.36μs) according to the conversion Eq. (4) in Section 3. Furthermore, we randomly set source partition and destination partitions for all message frames, and 30% of them are multicast, that is, they have multiple destination

partitions. The routing of messages follows the principle of load balancing, and every message frame has a harmonic period with its source partition and an identical recertification cost (1 unit) on each link of its transmission path. Table 3 shows the average bandwidth utilization and maximum bandwidth utilization of the backbone network. It also labels the corresponding configuration identifications subject to different length ranges in different experimental groups. It is evident that with the increment of frame length in the same group or message count for different groups, the bandwidth utilization tends to rise linearly. Configuration 11 and 12 can reflect the practical scenario where TT message accounts for 20%–30% of the entire network bandwidth.

Taking the frame length of class A, the original distribution of messages for different experimental groups and its upgrade when integrating new partitions and messages are demonstrated in Fig. 7. The interconnected line between the end system and the access switch, the access switch and the backbone switch or two backbone switches represents the physical link, and the thickness reflects the magnitude of bandwidth utilization on this link, i.e. the thicker the line, the higher utilization on it. However, the interconnected line between the partition

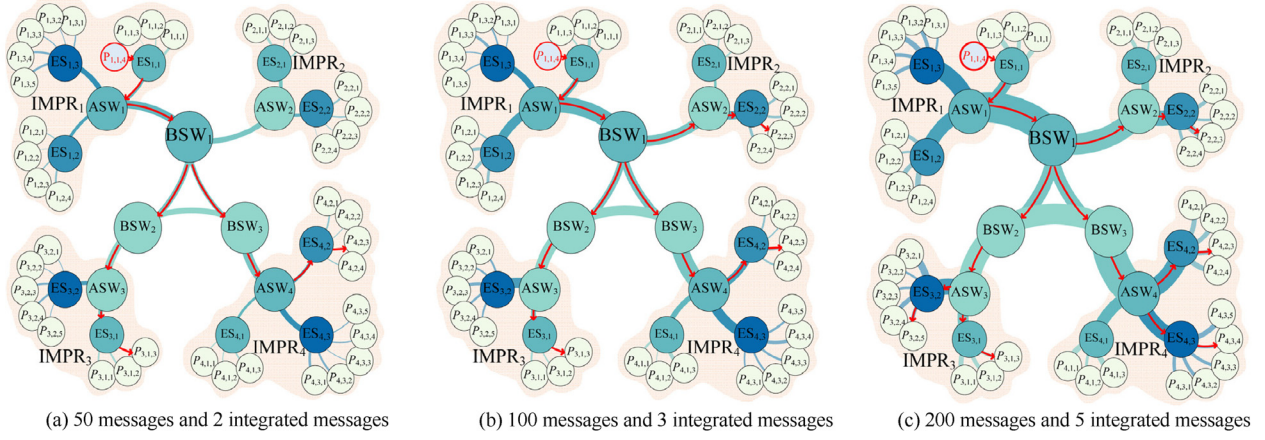


Fig. 7 Distribution of messages and upgraded configuration for different experimental groups in Case 2.

and the end system simply represents the virtual connection, not actual transmission link, and its thickness can only indicate how many messages the partition produces or consumes. In addition, the highlighted $P_{1,1,4}$ denotes the partition to be integrated for system upgrade, and it has the same period of 80 ms and time window of 2.7 ms for every experimental group. It respectively produces 2, 3 and 5 messages in Group I, Group II and Group III. The messages have the same time window of 80 μ s for different experimental groups, and their transmission are identified by red directed lines in Fig. 7. Other constant parameters related to the hybrid scheduling are set as follows: $\min_{\text{tech}} = 1 \mu$ s, and $\max_{\text{tech}} = 100$ ms.

5.2. Performance evaluation of HSA for DIMA systems

Through the performance comparison with the benchmark algorithms, i.e. TSA and DSA, we emphasize the scheduling advantages of our proposed HSA for original DIMA system design. The comparison results of two cases are respectively presented in Fig. 8 and Fig. 9.

For the fully connected DIMA system case, we detailed describe the scheduling offsets and the assigned time windows for partitions and message frames in Fig. 8. All of the three algorithms complete the scheduling design within 30ms, and their scheduling schemes are feasible. However, when using TSA, the scheduling scheme clearly shows the mismatch between message frames and partition execution. For example,

$P_{2,1}$ is the source partition of f_4 and remains active during [0,30] ms, while the scheduling sending time of f_4 starts prematurely at 3 ms. For another example, f_3 is forwarded through its last transmission link at 103 ms, missing the activated time window of its destination partition $P_{3,2}$ in current period. Those will seriously deteriorate the partition-level end-to-end delay but can be optimized by introducing hybrid scheduling constraints, as shown by DSA and HSA. Further, when using HSA, the average end-to-end delays of message frames at the partition-level and the network-level are 16 ms and 12.4 ms respectively, which are significantly more compact than those when using DSA with an optimization of 55.56% and 8.82%.

For the industrial DIMA system case, we briefly show the performance comparison of TSA, DSA and HSA in terms of three aspects, including average end-to-end delays of message frames at the partition-level and the network-level as well as the running time of algorithms, as shown in Fig. 9. Through the comparison in Fig. 9(a), it can be observed that TSA and DSA only give feasible scheduling schemes and their average end-to-end delay at the partition-level respectively fluctuates between [50,65] and [18,30] ms with little change for different experiments. However, HSA aims to optimize partition-level delay and limits it within 6 ms, which is 95.63% lower than TSA and 89.34% lower than DSA on average. And the partition-level delay in HSA rises with the increment of frame length in the same experimental group (i.e. Group 1 {1–4}, Group 2 {5–8} and Group 3 {9–12}) or message count with

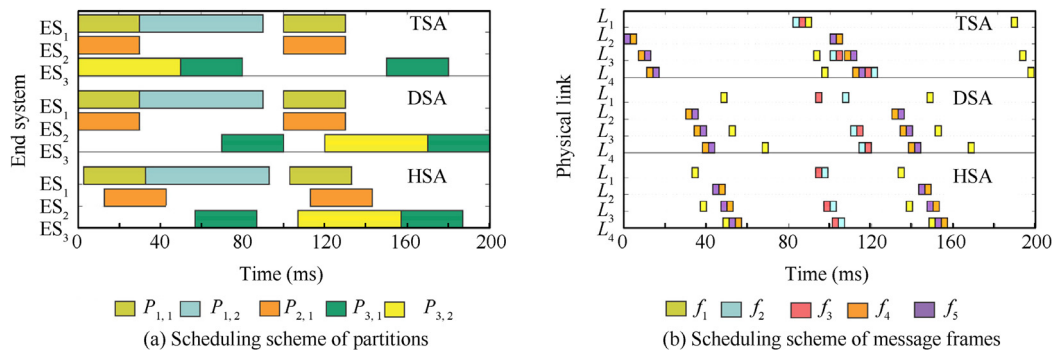


Fig. 8 The scheduling results of the original DIMA system in case 1 with TSA, DSA and HSA.

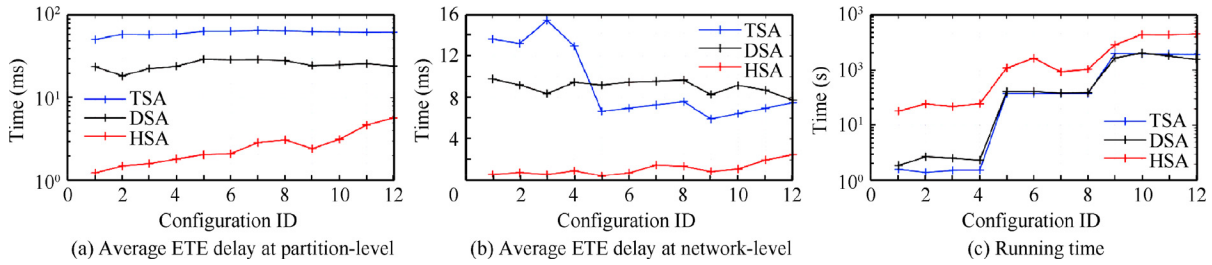


Fig. 9 Scheduling results of the original DIMA system in Case 2 with TSA, DSA and HSA.

the same length range class (i.e. Class A {1,5,9}, Class B {2,6,10}, Class C {3,7,11} and Class D {4,8,12}). Further, since the transmission of message frames are limited between the end of source partitions and the beginning of destination partitions, the compact arrangement of the partitions brings close scheduling of message frames in the network. Consequently, the average end-to-end delay of message frames at the network-level in HSA is also significantly better than that in TSA and DSA as shown in Fig. 9(b), and the average optimization comes up to 86.13% and 87.72% respectively. In addition, the rise of message count causes a sharp increase in the number of variables associated with the scheduling design. Therefore, the running time of all scheduling algorithms also tend to increase exponentially as shown in Fig. 9(c). Note that the introduction of the optimization function in HSA inevitably leads to a longer running time than TSA. However, HSA still can complete the scheduling design in 8min, which is acceptable for offline pre-scheduling of industrial DIMA systems.

5.3. Performance evaluation of OSA for upgraded DIMA systems

For upgraded DIMA system scheduling design, OSA need to inherit the scheme of its original DIMA system. Therefore, we apply the original scheduling scheme, obtained with TSA, DSA and HSA in Section 5.2, as the basis of the upgraded scheduling with OSA. And the upgraded scheduling results of OSA based on them are respectively marked as OSA-T, OSA-D and OSA-H. In order to compare and analyze the performance of different algorithms, we also directly use TSA, DSA and HSA in the upgraded scheduling, and respectively mark their results as TSA-I, DSA-I and HSA-I to distinguish from the original ones. The scheduling results of two cases are respectively presented in Fig. 10 and Fig. 11. In each case, the results are divided into three groups for comparative evaluation, i.e. the TSA group {TSA-I and OSA-T based on TSA}, the DSA group {DSA-I and OSA-D based on DSA} and the HSA group {HSA-I and OSA-H based on HSA}.

Fig. 10 detailed displays the scheduling offsets and the assigned time windows of partitions and message frames for the fully connected DIMA system case. The red legends in it represent the newly integrated partition $P_{2,2}$ and message frame f_6 . Obviously, TSA, DSA and HSA do not consider the relevance of the original scheduling scheme and completely reschedule the upgraded system. Their scheduling results are occasionally consistent with corresponding original schemes, and therefore cause higher integration cost. However, OSA

inherits the original scheme as much as possible to ensure the maximum consistency and has clear advantage in saving integration cost.

We further perform statistical analysis on the results of Fig. 10, and present the average end-to-end delays at the partition-level and the network-level, as well as the integration cost in each algorithm in Table 4. Since TSA, DSA and HSA do not rely on original schemes, TSA-I, DSA-I and HSA-I can be simply considered as the scheduling results of an independent DIMA system. The comparison of TSA-I, DSA-I and HSA-I show that the average partition-level delay and network-level delay in TSA are significantly worse than the other two algorithms, and further those delays in HSA are 45.71% and 35.29% better than TSA respectively. This also verifies the conclusion in Section 5.2; that is, HSA has distinct advantages for independent system scheduling. Then we compare the results in the TSA group, the DSA group and the HSA group. In each group, it can be found that the average end-to-end delay in OSA and the other algorithm, i.e. TSA, DSA or HSA, are numerically similar; while the integration cost in OSA is greatly lower: OSA-T and OSA-D are both 100% lower than TSA-I and DSA-I, and OSA-H is 86.36% lower than HSA-I. This shows that OSA performs better than TSA, DSA and HSA for upgraded system scheduling. Furthermore, we also focus on the performance difference of the DSA group and the HSA group, both of which introduce the hybrid scheduling design. Note that compared with OSA-D, OSA-H reduces 41.42% and 27.62% in terms of average partition and network delay respectively, and adds 3units in terms of integration cost. Though the integration cost of OSA-H increases slightly at an acceptable level, its end-to-end delays are greatly deduced. This can also be used as evidence that HSA is better than DSA for independent system scheduling.

Fig. 11 briefly shows the performance comparison of the four algorithms in terms of four aspects, including average end-to-end delays of message frames at the partition-level and the network-level, integration cost when integrating new functions and running time of different algorithm programs. From Figs. 11(a), (b) and (c), we can intuitively discover similar law with case 1. In each comparison group, the average end-to-end delays at the partition-level and the network-level in OSA are numerically similar with those in the corresponding basic algorithm, whereas the integration cost in OSA is greatly lower. This is due to the introduction of soft constraints in OSA. Besides, compared with OSA-T and OSA-D, OSA-H greatly reduces the average end-to-end delays at a low integration cost and performs better overall for the upgraded DIMA system scheduling. Fig. 11(d) shows that the running time of

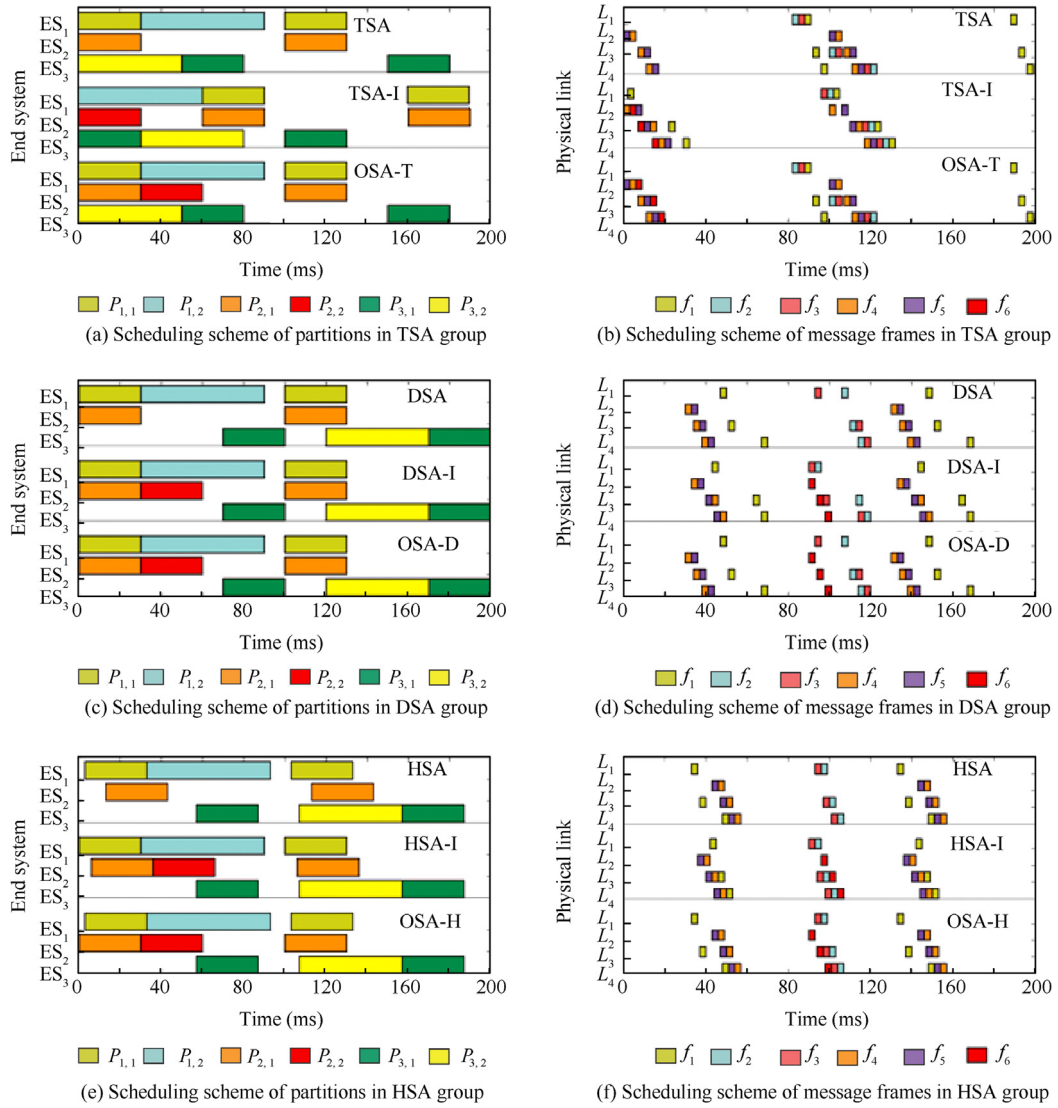


Fig. 10 Scheduling results of the upgraded DIMA system in Case 1 with different algorithms.

OSA is also closely related to the message count, and generally shorter than that of the corresponding basic algorithm in each comparison group. However, when more scheduling offsets for partitions and message frames need modification, causing larger integration cost, the running time of OSA is significantly longer.

Notably, although HSA aims to minimize the end-to-end delay of messages frames at the partition-level, its average delay is even larger than OSA for experiments with configuration 8–12, as shown in Fig. 11(a). This is due to the custom termination conditions of Gurobi described earlier. In order to clearly explain the reason, we change the minimal number of exploring nodes in the termination condition to 3×10^4 and 5×10^4 , and respectively repeat all experiments. Since the advantages of HSA over TSA and DSA have been explained above, we emphasize the comparison of HSA and OSA and their scheduling results are shown in Fig. 12. The minimal numbers of exploring nodes are marked in the parenthesis of every legend. The HSA results are plotted on the left ordinate axis, and the OSA results are presented by the ratio to the cor-

responding HSA results on the right ordinate axis. Fig. 12(a) demonstrates that as the minimal number of exploring nodes increases, the average end-to-end delay at the partition-level in HSA decreases, and when this number reaches 5×10^4 , the average delays in HSA with all 12 configurations are less than those in OSA. However, the average delay of OSA is also limited within 6ms in Fig. 12(a), and both algorithms have similar capability for the average end-to-end delay at the network-level in Fig. 12(b). Furthermore, OSA still has prominent advantages in terms of integration cost and running time, respectively shown in Figs. 12(c) and (d). Meanwhile, through analyzing the scheduling design of HSA with different minimal exploring nodes, i.e. HSA(1), HSA(3) and HSA(5), the variances of partition-level delay and network-level delay are on average 0.042 and 0.015, respectively. These statistical results illustrate the robustness of our custom termination from a certain perspective.

In addition, Fig. 12(d) also shows that although OSA program runs up to 5.51 h for the experiment with configuration 11 when exploring at least 1×10^4 nodes, its running time is

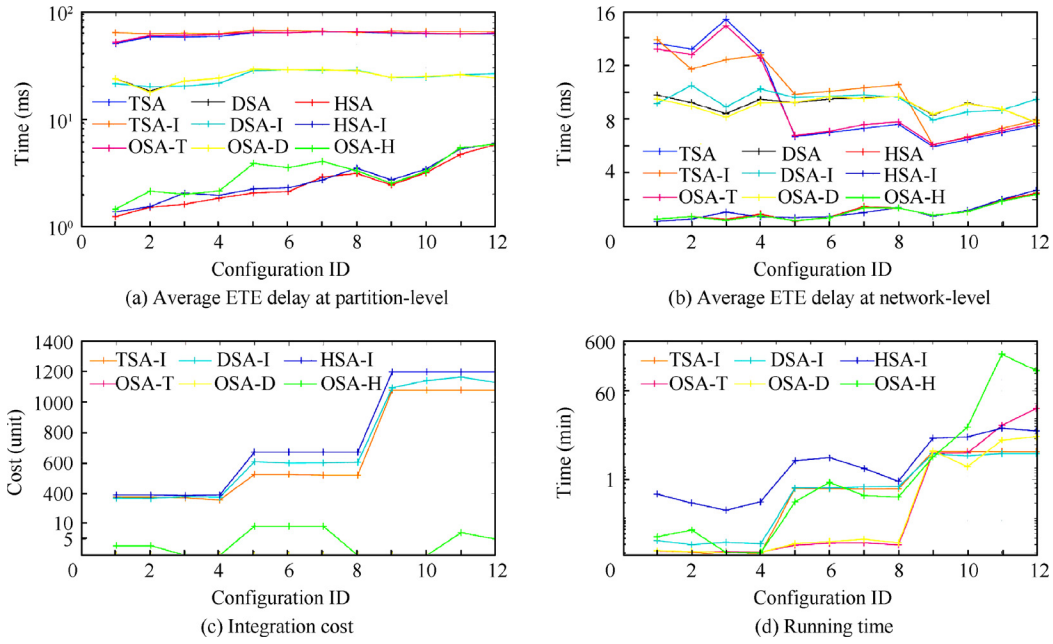


Fig. 11 Scheduling results of the upgraded DIMA system in Case 2 with different algorithms.

Table 4 Statistics of scheduling results with different algorithms in Case 1.

Algorithms	Average ETE delay at partition-level (ms)	Average ETE delay at network-level (ms)	Integration cost (unit)
TSA	196	20	
TSA-I	140	20.5	29
OSA-T	215	18.5	0
DSA	36.0	13.6	
DSA-I	35.0	17.0	14
OSA-D	35.0	12.67	0
HSA	16.0	12.4	
HSA-I	19.0	11.0	22
OSA-H	20.5	9.17	3

significantly less than HSA in general and limited within 1.51h when exploring at least 3×10^4 nodes or more. However, the runtime of HSA is strongly related to the count of exploring nodes and message frames. In order to clearly present this correlation, we select the experiments with configurations 1, 5 and 9, which have the same length range of message frames, and then record the optimality GAP and running time for different exploring nodes as shown in Fig. 13. For example, C1_{N5} represents the results of experiment with configuration 1 when exploring at least 5×10^4 nodes. Each point in the figure represents a feasible solution for the experiments. It can be found that as the number of exploring nodes or message frames increases, the optimality GAP of HSA decreases, while the running time increases apparently and reaches 4h for 200 message frames when exploring at least 5×10^4 nodes.

We also notice there is an abnormal phenomenon in Fig. 12 (d), that the runtime of HSA for some experiments (e.g. experiments with configuration 8 and 12) is apparently less than that for other experiments in the same group (i.e. Group I, Group

II or Group III). This is actually due to different densities of feasible solutions under different configurations. We take experiments with configuration 11 and configuration 12, which have the same number of message frames, as examples to explain and depict the optimality GAP and running time of HSA for different minimal exploring nodes. The comparison results are shown in Fig. 14. From Fig. 14(a), it can be found that the density of feasible solutions is usually higher after the turning point of optimality GAP. In the range of 5×10^4 exploring nodes, the experiment with configuration 12 only has a turning point at the 11609th node, while the experiment with configuration 11 has at least five turning points, respectively at 9767, 10514, 14516, 33396 and 46647. Since the density of feasible solution nodes increases after the turning points, the running time of Gurobi rises rapidly. Therefore, in Fig. 14(b), when exploring at least 1×10^4 , 1×10^4 des, the running time for experiments with configuration 11 and configuration 12 is similar; when 3×10^4 , 3×10^4 , the running time with configuration 11 is 22 min longer; and when 5×10^4 , enormously 5 h longer. Besides, this rule, that the larger feasible solution density, the faster the runtime increases, also applies to Fig. 13.

Based on the above comparison and analysis, OSA is more suitable for upgraded DIMA system scheduling design with balanced consideration of various factors, including average end-to-end delays at the partition-level and the network-level, integration cost when integrating new functions and running time of algorithm program.

6. Conclusions

This paper focuses on the hybrid partition- and network-level scheduling for DIMA systems. We firstly establish a general DIMA system model, which coordinates partitioned processing defined in ARINC 653 and unified network typically implemented through Time-Triggered Ethernet. Based on the

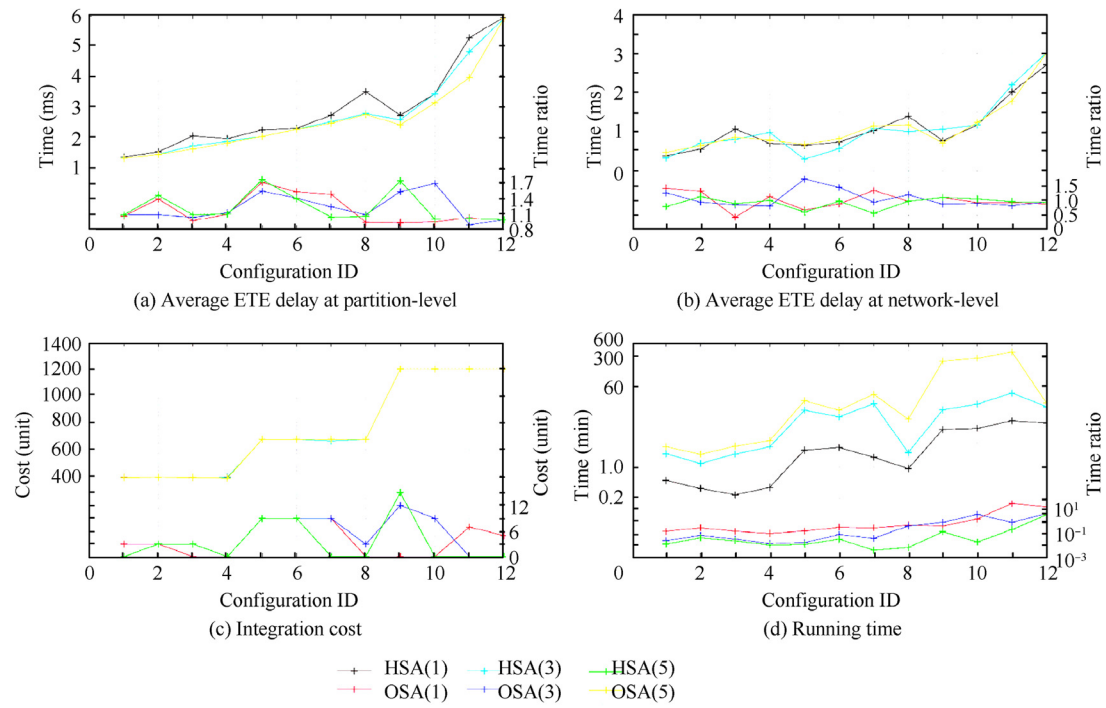


Fig. 12 Scheduling results of the upgraded DIMA system in Case 2 with HSA and OSA under different minimal number of exploring nodes.

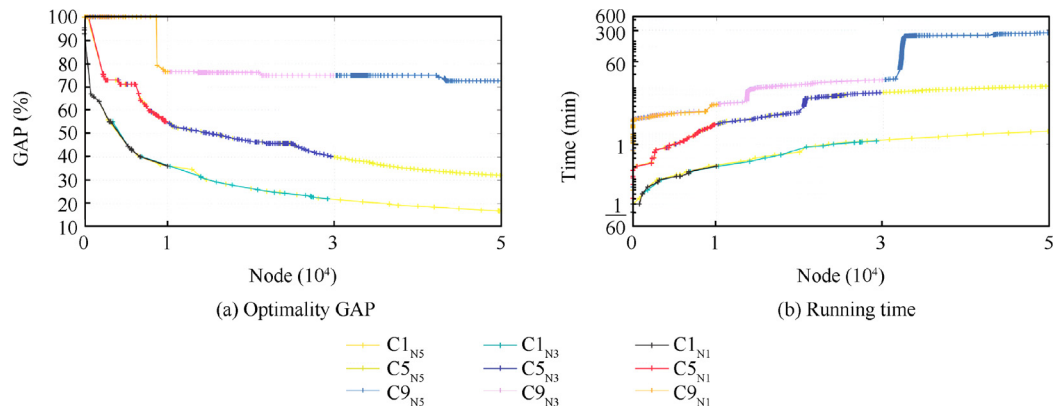


Fig. 13 Optimality GAP and running time for different number of message frames in HSA with different minimal number of exploring node.

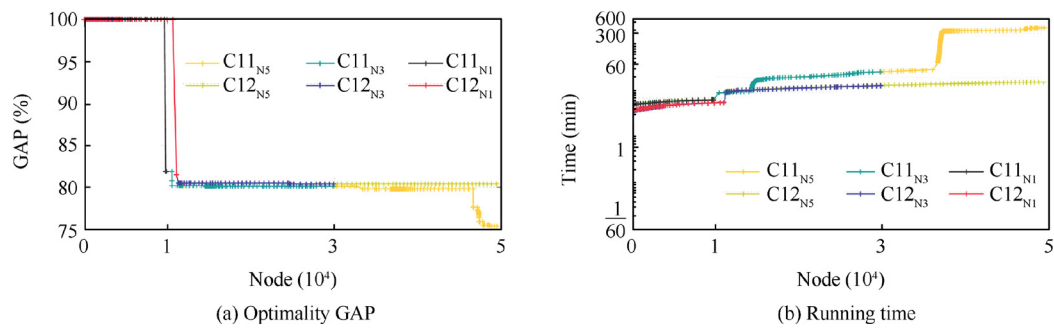


Fig. 14 Optimality GAP and running time for different range of frame length in HSA with different minimal number of exploring nodes.

model, we apply MIP to design a scheduling algorithm with the minimal end-to-end delay at the partition-level as the optimization goal. We further propose an MAXSAT-based optimal algorithm specifically for upgraded DIMA system scheduling, which reduces the integration cost caused by the upgrade or reconfiguration of DIMA systems when incremental integrating new functions. Compared with traditional scheduling algorithm, the experiment results of two typical cases, including a simple fully connected DIMA system case and an industrial DIMA system case, demonstrate that our proposed MIP-based hybrid scheduling algorithm manages to design a feasible scheme for independent DIMA systems with a more compact end-to-end arrangement in an acceptable time. Its modification algorithm for upgraded DIMA systems, i.e. MAXSAT-based optimization scheduling algorithm, inherit its compactness and effectively limits the integration cost.

In future work, the hybrid partition- and network-level scheduling design will be investigated with other forms of performance indicator, such as the worst case end-to-end transmission delay for Rate-Constraint traffic flows.

Acknowledgements

This study was co-supported by the National Natural Science Foundation of China (No. 71701020), the Defense Research Field Foundation of China (No. 61403120404) and the Civil Aircraft Airworthiness and Maintenance Key Laboratory Fund of Civil Aviation University of China (No. 2017SW02).

References

- Robati T, Gherbi A, Kouhen AE, Mullins J. Design and simulation of distributed IMA architectures using TTEthernet: a model-driven approach. *J Ambient Intell Hum Comput* 2017;**8**(3):345–55.
- Wolfig R, Jakovljevic M. Distributed IMA and DO-297: Architectural, communication and certification attributes. *Proceedings of AIAA/IEEE digital avionics systems conference*; 2008 Oct 29–30; St. Paul, United States. Piscataway: IEEE Press; 2008. p. 1–10.
- Steiner W. An evaluation of SMT-based schedule synthesis for time-triggered multi-hop networks. *Proceedings of real-time systems symposium*; 2010 Nov 30–Dec 3; San Diego, United States. Piscataway: IEEE Press; 2010. p. 375–84.
- Steiner W. Synthesis of static communication schedules for mixed-criticality systems. *Proceedings of 2011 14th IEEE international symposium on object/component/service-oriented real-time distributed computing workshops*; 2011 Mar 28–31; Newport Beach, United States. Piscataway: IEEE Press; 2010. p. 11–8.
- Zhang YJ, He F, Lu GS, Xiong HG. An impositivity message scheduling based on modified generic algorithm for time-triggered Ethernet. *Sci China Inf Sci* 2018;**61**(1) 019102.
- Zhang LC, Goswami D, Schneider R, Chakraborty S. Task-and network-level schedule co-synthesis of Ethernet-based time-triggered systems. *Proceedings of the Asia and South Pacific design automation conference*; 2014 Jan 20–23; Suntec, Singapore. Piscataway: IEEE Press; 2014. p. 119–24.
- Craciunas SS, Craciunas RS. Combined task-and network-level scheduling for distributed time-triggered systems. *Real-Time Syst* 2016;**52**(2):161–200.
- Tamas-Selicean D, Pop P, Steiner W. Synthesis of communication schedules for TTEthernet-based mixed-criticality systems. *CODES + ISSS'12 – Proceedings of the 10th ACM International conference on hardware/software-codesign and system synthesis, co-located with ESWEAK*; 2012 Oct 7–12; Tampere, Finland. New York: ACM Press; 2012. p. 473–82.
- Tamas-Selicean D, Pop P, Steiner W. Design optimization of TTEthernet-based distributed real-time systems. *Real-Time Syst* 2015;**51**(1):1–35.
- Suethanuwong E. Scheduling time-triggered traffic in TTEthernet systems. *IEEE symposium on emerging technologies and factory automation*; 2012 Sep 17–21; Krakow, Poland. Piscataway: IEEE Press; 2012. p. 1–4.
- Zheng Z, He F, Xiong Y. The research of scheduling algorithm for time-triggered Ethernet based on path-hop. *Proceedings of AIAA/IEEE digital avionics systems conferences*; 2016 Sep 25–29; Sacramento, United States. Piscataway: IEEE Press; 2012. p. 1–6.
- Lauer M, Mullins J, Yeddes M. Cost optimization strategy for iterative integration of multi-critical functions in IMA and TTEthernet architecture. *Proceedings of international computer software and applications conference*; 2013 Jul 22–26; Kyoto, Japan. Piscataway: IEEE Press; 2013. p. 139–44.
- Beji S, Hamadou S, Gherbi A, Mullins J. SMT-based cost optimization approach for the integration of avionic functions in IMA and TTEthernet architectures. *Proceedings of IEEE international symposium on distributed simulation and real-time applications*; 2014 Oct 1–3; Toulouse, France. Piscataway: IEEE Press; 2014. p. 165–74.
- Pozo F, Steiner W, Rodriguez-Navas G, Hansson H. A decomposition approach for SMT-based schedule synthesis for time-triggered networks. *IEEE international conference on emerging technologies and factory automation*; 2015 Sep 8–11; Luxembourg, Luxembourg. Piscataway: IEEE Press; 2015. p. 1–8.
- Pozo F, Rodriguez-Navas G, Steiner W, Hansson H. Period-aware segmented synthesis of schedules for multi-hop time-triggered networks. *Proceedings of 2016 IEEE 22nd international conference on embedded and real-time computing systems and applications*; 2016 Aug 17–19; Daegu, Korea. Piscataway: IEEE Press; 2016. p. 170–5.
- Pozo F, Rodriguez-Navas G, Hansson H, Steiner W. Schedule synthesis for next generation time-triggered networks. *DiVA* 2017.
- Beji S, Gherbi A, Mullins J, Hladik PE. Model-driven approach to the optimal configuration of time-triggered flows in a TTEthernet network. *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*; 2016 Oct 3–4; Saint-Melo, France. Berlin: Springer; 2016. p. 164–79.
- Xu YY, Ge HJ, Yang JZ. Research on the scheduling method of AFDX terminal system based on time triggered and event triggered. *J Comput Sci* 2017;**27**:445–53.
- An DH, Kim KH, Kim KI. Optimal configuration of virtual links for avionics network systems. *Int J Aerosp Eng* 2015;**2015**:1–9.
- Ashjaei M, Patti G, Behnam M, Nolte T, Alderisi G, Bello LL. Schedulability analysis of Ethernet Audio Video Bridging networks with scheduled traffic support. *Real-Time Syst* 2017;**53**(4):526–77.
- Pop P, Raagaard ML, Craciunas SS, Steiner W. Design optimization of cyber-physical distributed systems using IEEE time-sensitive networks. *IET Cyber-Phys Syst: Theor Appl* 2016;**1**(1):86–94.
- Lu H, Zhou Q, Fei Z, Zhou R. Scheduling based on interruption analysis and PSO for strictly periodic and preemptive partitions in Integrated Modular Avionics. *IEEE Access* 2018;**6**:13523–40.
- Sheikh A, Brun O, Hladik PE, Prabhu BJ. Strictly periodic scheduling in IMA-based architectures. *Real-Time Syst* 2012;**48**(4):359–86.
- Garcia-Valls M, Domínguez-Poblete J, Touahria IE, Lu CY. Integration of data distribution service and distributed partitioned systems. *J Syst Archit* 2018;**83**:23–31.
- Deroche E, Scharbarg JL, Fraboul C. Mapping real-time communicating tasks on a distributed IMA architecture. *IEEE*

- international conference on emerging technologies and factory automation*; 2016 Sep 6-9; Berlin, Germany. Piscataway: IEEE Press; p. 1-8.
26. Giannopoulou G, Stoimenov N, Huang PC, Thiele L, Dinechin BD. Mixed-criticality scheduling on cluster-based manycores with shared communication and storage resources. *Real-Time Syst* 2016;**52**(4):399-449.
 27. Chen JC, Du CL, Han PC. Scheduling independent partitions in integrated modular avionics systems. *PLoS ONE* 2016;**11**(12) e0168064.
 28. Rouxel B, Derrien S, Puaut I. Tightening contention delays while scheduling parallel applications on multi-core architectures. *ACM Trans Embedded Comput Syst* 2017;**16**(5):164.
 29. García-Valls M, Perez-Palacin D, Mirandola R. Pragmatic cyber-physical systems design based on parametric models. *J Syst Softw* 2018;**144**:559-72.
 30. Bersani MM, García-Valls M. Online verification in cyber-physical systems: Practical bounds for meaningful temporal costs. *J Software: Evolut Process* 2018;**30**(3) e1880.
 31. García-Valls M, Rodríguez-López I, Fernández-Villar L. iLAND: An enhanced middleware for real-time reconfiguration of service oriented distributed real-time systems. *IEEE Trans Ind Inf* 2013;**9**(1):228-36.
 32. Hu ML, Luo J, Wang Y, Lukasiewicz M, Zeng Z. Holistic scheduling of real-time applications in time-triggered in-vehicle networks. *IEEE Trans Ind Inf* 2014;**10**(3):1817-28.
 33. Craciunas SS, Oliver RS. SMT-based task-and network-level static schedule generation for time-triggered networked systems. *Proceedings of the 22nd international conference on real-time networks and systems*; 2014 Oct 8-10; Versailles, France. New York: ACM Press; 2014. p. 45-54.
 34. Blikstad M, Karlsson E, Loow T, Ronnberg E. An optimisation approach for pre-runtime scheduling of tasks and communication in an integrated modular avionic system. *Optimiz Eng* 2018;**19**(4):977-1004.
 35. Annighofer B, Thielecke F. Multi-objective mapping optimization for distributed Integrated Modular Avionics. *Proceedings of AIAA/IEEE digital avionics systems conference*, 2012 Oct 14-18; Williamsburg, United states. Piscataway: IEEE Press; 2012. p. 1-13.
 36. Zhang C, Xiao JL. Modeling and optimization in distributed integrated modular avionics. *Proceedings of AIAA/IEEE digital avionics systems conference*; 2013 Oct 5-10; East Syracuse, United States. Piscataway: IEEE Press; 2013. p. 1-12.
 37. Wang T, Gu QF. Research on distributed integrated modular avionics system architecture design and implementation. *Proceedings of AIAA/IEEE digital avionics systems conference*; 2013 Oct 5-10; East Syracuse, United states. Piscataway: IEEE Press; 2013. p. 1-10.
 38. Bjoern A, Reif C, Thieleck F. Network topology optimization for distributed integrated modular avionics. *Proceedings of AIAA/IEEE digital avionics systems conference*; 2014 Oct 5-9; Colorado Springs, United States. Piscataway: IEEE Press; 2014. p. 1-12.
 39. Xu JR, Li Q, Xiong HG, Zhao LX. Real-time communication flow latency analysis of DIMA systems. *Comput Eng Des* 2015;**36**(4):879-85 [Chinese].
 40. He F, Li ES. Deterministic bound for avionics switched networks according to networking features using network calculus. *Chin J Aeronaut* 2017;**30**(6):1941-57.
 41. De Moura L, Bjorner N. Z3: An efficient SMT solver. *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*; 2008 Mar 29-Apr 6; Budapest, Hungary. Berlin: Springer; 2008. p. 337-40.
 42. Gurobi Optimization [Internet]. Gurobi optimizer reference manual, Version 7.5 [cited 2018 Jun 19]. Available from: <http://www.gurobi.com/>.
 43. Bjorner N, Phan AD. vZ-Maximal Satisfaction with Z3. *SCSS* 2014;**30**:1-9.