MIP-based Joint Scheduling and Routing with Load Balancing for TSN based In-vehicle Networks

Ammad Ali Syed, Serkan Ayaz, Tim Leinmüller
DENSO AUTOMOTIVE Deutschland GmbH,
Germany
Corporate R&D
{a.syed, s.ayaz, t.leinmueller}@eu.denso.com

Madhu Chandra
Chemnitz University of Technology,
Germany
madhu.chandra@etit.tu-chemnitz.de

2020 IEEE Vehicular Networking Conference (VNC) | 978-1-7281-9221-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/VNC51378.2020.9318350

***Abstract*—As the mobile and automotive industries move towards autonomous vehicles, many advanced driving applications have been developed. These driving applications may require different levels of intelligence, communication capabilities, and processing power from the communication network and processing platform. These advanced driving applications can be grouped into static, which runs all the time as the engine starts and dynamic, which runs for a duration of time depending on the vehicle conditions. After the emergence of IEEE Time-Sensitive Networking (TSN) features for Ethernet technology, the automotive industry started to move towards the usage of TSN for advanced driving applications. However, IEEE TSN poses a challenge in streamlining the schedules and routes of the dynamic traffic since they require swift and fast determination of transmission schedules and routes on-the-fly. In this paper, we mainly focus on static traffic and device a novel static scheduling and routing algorithm that would be conducive for dynamic traffic requirements. In this approach, we have developed Mixed-integer programming (MIP) based joint scheduling and routing of static applications with the aim of load balancing such that more dynamic traffic would be schedulable as the vehicle drives off. We proposed two load-balancing based objective functions and conducted an experimental analysis of objective functions with six different vehicle network configurations in two scales of zonal architecture. Experimental evaluations show the efficacy of our developed algorithm, in which the load is balanced in the egress port of the network, which in turn can schedule more dynamic traffic.**

***Index Terms*—time-sensitive network (TSN), in-vehicle network (IVN), vector bin packing (VBP), area control unit (ACU), gate control list (GCL)**

I. Introduction

Advancements in computer hardware and processing power have led to many advanced driving applications which could make fully automated vehicles possible. These advanced driving applications can include camera-based, lidar-based and radar-based advance driver assistance system (ADAS) applications, vehicle-to-vehicle (V2V) as well as vehicle-to-infrastructure (V2I) based applications (e.g., platooning, sensor data exchange) which will, in turn, be a part of fully automated driving (AD) system. With the increasing number of applications, the demand for electric and electronic (EE) systems is expanding exponentially. Each new application is implemented by using a new electronic control unit (ECU) functionality in the vehicle. As the number of ECUs increase in the vehicle, the communication demand among ECUs in the vehicle increase as well. While some of the above-mentioned applications can be grouped under safety-critical applications, some of the other can be grouped under non-safety-critical applications.

For in-vehicle communication, Ethernet and its Time-Sensitive Networking (TSN) extensions are now being considered as a potential candidate for fully automated vehicles, due to its cost-effectiveness and advanced capabilities supporting different in-vehicle services (i.e., time-triggered, rate constraint, and best effort) with different data rates [3] [4]. As the automation level of vehicle increases, it will require more intelligence supported by a diverse set of sensors and ECUs to fulfil the functional safety requirements. These sensors typically include cameras, lidars, radars, sonars and High-Definition (HD) map that are connected through in-vehicle TSN network on the physical level and interacting with the central controller (i.e., central processing unit (CPU)). These sensors will generate an ample amount of data which will be transferred in the vehicle as well as among vehicles to increase situational awareness of the vehicle. According to Intel, a current available self-driving car can generate 4 Terabytes of data in a single day [5]. Managing the copious amount of data is a formidable task. Among those sensors, some of them generate critical data which varies over time (i.e., lidar, radar and HD-map). It is difficult to design a static scheduling and routing algorithm for variable packet length data. However, designing a static schedule for the worst-case data volume often leads to poor network utilization of available bandwidth [23].

With the growing need of flexibility in the designing of the automotive system, there are some legacy standards (e.g., TTCAN, FTTCAN and Flexray) which allow integrating static as well as dynamic traffic in the network. These standards were motivated by the significant desire for supporting dynamic traffic without jeopardizing time-triggered traffic. The main drawback of these legacy standards is that they were not designed to cover higher bandwidth applications. In future vehicles, the scope of dynamic traffic will be increased as the number of features and functionalities requiring dynamic traffic handling in the vehicle increases. Some of the dynamic applications are V2V, V2X, V2N communication, adaptive cruise control, truck trailer systems and over the air (OTA) software updates.

Authorized licensed use limited to: University of Nantes. Downloaded on February 09,2025 at 16:24:09 UTC from IEEE Xplore. Restrictions apply.

978-1-7281-9221-5/20/$31.00 ©2020 IEEE

Automotive Ethernet with Time-Sensitive Networking (TSN) features is now considered as the replacement of these aforementioned legacy protocols since it provides higher bandwidth with a considerable cost advantage. Making use of IEEE TSN standards, particularly IEEE 802.1 Qbv, one can achieve the highest level of determinism in Ethernet-based network [6]. In time-aware shaper (TAS), all the network devices are synchronized with the global time (IEEE 802.1ASrev). The shaper is deployed at the egress port of network device, and the incoming flows are filtered and placed in one of the eight priority queues. These queues are controlled by Gate Control List (GCL), which determines the opening and closing time of respective queue gates. Ethernet frames can be scheduled in a way that has collision-free access to the egress port by using GCL. Typically, global schedules and routes are determined based on the network topology and flow characteristics. These schedules are designed offline due to the high complexity of the problem (i.e., NP-hard [2]). Afterwards, these schedules are translated to the GCL of each egress ports.

The quest for this research work has been motivated by a desire to support dynamic traffic along with static traffic in TSN based Ethernet network. At this point, no one considers designing static schedules which apposite dynamic traffic efficiently. In this work, we propose MIP-based joint scheduling and routing in the in-vehicle network (IVN). Our algorithm generates GCL by optimizing scheduling and routing simultaneously. In contrast to state-of-the-art adamant solutions, our algorithm generates joint schedules and routes which could maximize the schedulability of dynamic traffic as the vehicle drives off. In this regard, we examine two germane load balancing based objective functions with different network configurations and topologies.

The remainder of the paper is organized as follows. Section II provides a survey of the related work. Section III describes the functional domain architecture of the in-vehicle network. The system model is introduced in Section IV. Section V presents a motivational example, problem formulation and MIP model for routing and scheduling. Section VI analyzes the experimental evaluations. Finally, we conclude the paper in section VII.

II. Related Work

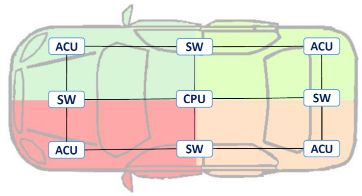
In recent years some seminal work which has been done in designing the static schedule of the time-aware shaper. Among them, some work focuses only on the problem of the schedule [7], while others consider joint optimization of schedules and routes [8]. In [9] authors used satisfiability modulo theories (SMT) based solver (Z3) to compute feasible schedule for IEEE 802.1 Qbv and showed that the computational complexity increase exponentially when the number of flows increases linearly. In [7], the author transformed the problem to flow shop problem and optimize the makespan of schedule, which is based on ILP. To reduce the computational time, the author proposed a tabu search based heuristics. Another ILP-based solver approach was designed by pop et al. [10] to synthesize egress port-specific GCL. Another strategy for solving routing and scheduling problem of TT communication in TSN system was designed by Gavrilut et al. [22]. Their work also considered the impact of AVB traffic in the network and produce schedules in which the timing requirements of TT and AVB flows are met. Schweissguth et al. [11] solved jointly scheduling and routing problem of TSN network. The author evaluates end-to-end delay as performance metrics of different topologies and traffic patterns. In addition to this study, the author in [12] also solved the scheduling and routing problem of time-triggered flows jointly with ILP. In this work, the computation of transmission schedules of TT flows would be completed within seconds for a large network.

All the aforementioned preeminent work that is used for solving scheduling and routing in TSN's based time-aware shaper belongs to the realm of static network configuration. These fixed network configurations are not flexible to use with dynamic scheduling and routing. In [13], the author proposed offline and online scheduling and routing algorithm for virtual machine (VM) migration in TSN's based time-aware shaper framework. The offline schedule is designed based on minimum distance tree (MDT), which consider only the potential dynamic targets. In this work, we focus on developing efficient static schedules and routes which could maximize schedulability of dynamic traffic.

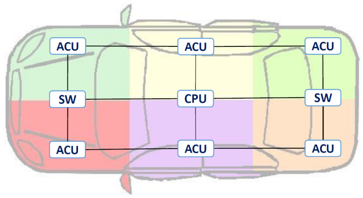
III. IVN FUNCTIONAL DOMAINS ARCHITECTURE

Automotive electronics and mechanical systems are grouped into different sub-groups or domains. Each domain works independently and cooperates with other domains if needed [1]. The domains include: 1) Autonomous Driving domain, in which components are responsible for driving vehicle autonomously. Lidar, radar, camera, ultrasonic sensors are the main components of this domain. 2) Power Train domain is mainly responsible for controlling the engine as requested by the driver. Controlling includes optimization of certain parameters, i.e. ignition timing, fuel consumption, driver comfort etc. This domain exhibit a stringent delay requirement for the network. 3) Chassis domain is mainly responsible for ensuring the safety and comfort of the driver and passengers (suspension). It controls the interaction of the vehicle with road [1] (wheel, suspension, etc.). Antilock braking system (ABS), electronic stability program (ESP), automatic stability control (ASC), and four-wheel drive (4WD) are some of the components that lie in this domain. 4) Body domain includes components which are not responsible for vehicle running. These include air-bag, wiper controller, window rolling, seat adjustment etc. 5) Telematics is responsible for data exchange between other vehicles and infrastructure. It includes V2X, V2I and V2N applications. 6) Infotainment is responsible for the interaction of human with different component in the car. Today, these automotive domains consider a hierarchical in-vehicle network architecture. In this hierarchical network design, every functional domain has its network segment. Each domain is connected to its domain controller (brain) via bus. The cross-domain functionalities are distributed over different Domain Controllers which leads to complexity. The wiring harness would also be a problem as the network gets

Authorized licensed use limited to: University of Nantes. Downloaded on February 09,2025 at 16:24:09 UTC from IEEE Xplore. Restrictions apply.



(a) Four zones



(b) Six zones

Fig. 1: Vehicle partitioning into different zones

complex when more and more ECUs are added. Statistics show that by 2030, the electronics content in cars accounted for 50% of an automobile's total production cost [24] and the third-highest cost is the wiring. Therefore, the architectural changes are required to cope with these problems. The idea of new zonal architecture is to consolidate many different functionalities while keeping physical domain secluded. The conceptual architecture of area control unit (ACU) in which vehicle area is partitioned into four and six zones, as shown in Fig. 1.

IV. SYSTEM MODEL

In this work, we model the network topology and flow mapping with two separate entities. The network topology is modelled as unweighted directed graph $G < \alpha, \beta >$, where α is the set of network nodes, having switching capabilities, each node deems as the set of ACUs, CPUs and ECUs and β is the set of links connecting ACUs, CPUs and ECUs. For example, a link $[\alpha_{acu1}, \alpha_{acu2}]$ of β transmit flows from source node ACU1 to the destination node ACU2 and similar a link $[\alpha_{acu2}, \alpha_{acu1}]$ of β transmit flows in opposite direction. Note that the flow is specified as an end-to-end flow which is composed of one or more links. The set of flows is denoted by F. Each flow $f_i \in F$ is characterized by a tuple $< src_i, dst_i, p_i, w_i >$, where src_i is the source node, dst_i is the destination node, p_i is the periodicity, w_i is the size of the flow. There is a minimum time period that the network support which is called as base-period. The maximum period in our network is also known, which is called as max-period. The set of egress ports of switching devices and end systems is denoted by E e.g. $E = e_1, e_2, e_3, e_4, \ldots, e_l$, wherein an egress port and respective ingress port constitute a link. A flow f_i from sender node α_a to the receiver node α_b routed through the egress ports of the intermediate nodes. The set of possible paths from source to the destination node for all flows is denoted by P. This set consists of possible shortest paths from source to destination, thereby reducing the search space by eliminating other arbitrary paths. This reduction in the search space results in saving the computational time of the optimizer.

The flow assignment is modeled as a vector bin packing (VBP), in which a set J of n items are given, where each item J_i, $i \in \{1, 2, 3, \ldots, n\}$ composes of a d-dimensional demand vector as $J_i^1, J_i^2, J_i^3, \ldots, J_i^d$. The goal is to partition the item set J into minimum number of feasible sets $B_1, B_2, B_3, \ldots, B_m$, where B_a is feasible if sum of vectors in it is less than the total capacity of the bin in any dimension. If there is only one dimension $(d = 1)$ then d-VBP is a classical bin packing problem. Both d-VBP and classical bin packing problem are NP-hard problems [14] [15]. Related to classical bin packing and its extension, rich seminal work has been published in literature. The published work targeting numerous applications few of them named as virtual machine placement in cloud computing [16], network design [17], logistic problem [18], layout design [19], and machine reassignment problem [20].

Each egress port of a switching device and end system is considered as a single resource element. The capacity of a single resource element is partitioned into bins, and each bin is composed of dimensions that represent the egress port. Each bin's dimension has a capacity which is equal to base-period time, as shown in Fig. 2b. The load on the egress port is defined as the summation of transmission times of all flows passing through the port. Different flows traversing the switch egress port accrued the load on the egress port additively and shaved the capacity of the resource element off by its size. Some of the flows are utilizing multiple resource elements, while some are requiring at least single. There could be multiple tangible transmission paths from source to destination node that implicitly depend on the network topology. The load on the network will be larger if there are more hops in the transmission path as compared to fewer hops. More hops mean more egress ports are required to transfer the data from source to destination, resulting in large load over the network.

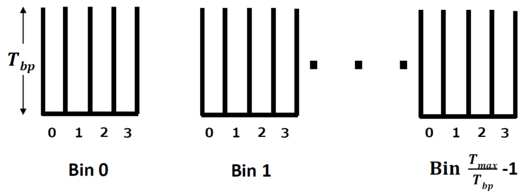
In our problem, the items are viewed as flows with d-dimensional resource demand vector. The resource demand vector tells the required resource elements which define a specific flow composed of a set of egress ports. For a single flow, there are multiple resource demand vectors which are because of multiple potential paths. The size of an item is the flow size which remains the same in all dimensions. As mentioned above, the base-period T_{bp} partitions the max-period T_{max} into bins $[i.T_{bp}, (i + 1).T_{bp})$. As there are T_{max}/T_{bp} number of bins which is static and known at design time. Taking Fig. 2a into account, it can be observed that there are four egress ports in the network. These egress ports in the network are modelled as the dimensions of the bin, i.e. dimensions from 0 to 3, as shown in Fig. 2b.

V. PROBLEM FORMULATION

In this section, we first discuss the motivational example that underpins the development of efficient static schedules and routes. We then state and formulate the problem.

Authorized licensed use limited to: University of Nantes. Downloaded on February 09,2025 at 16:24:09 UTC from IEEE Xplore. Restrictions apply.

(a) Network topology



(b) Vector bin packing

Fig. 2: Network topology and flow assignment model

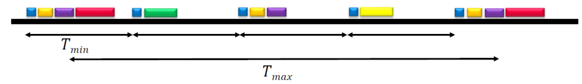
A. MOTIVATION

Load balancing has been a widely studied technique, and it is used to improve resource utilization in distributed embedded system efficiently [21]. With the proposed load balancing strategy, we will have static schedules and routes as the vehicle engine starts after ignition. These static schedules and routes are designed in a way such that more dynamic flows are schedulable later as the vehicle operates. The rationale behind developing a load-balanced static schedule and route is explained in the example. Consider single sender, switch and receiver as shown in Fig. 3a. The flows are transmitted through the egress port of the switch and are repeated according to their periodicity. For the sake of simplicity, packet transmission from the switch's egress port is shown. The flow with base-period (blue flow) partition the available bandwidth into bins (number of bins=4) of fixed size. The flow which has a maximum period (red flow) has only one instance within the transmission cycle. The number of instances of the flow j within the transmission cycle can be determined by maximum period and period of flow j, i.e. T_{max}/T_j. Consider the schedule in Fig. 3b, in which the static scheduler is designed without considering the network-wide load balancing then there might be some dynamic schedule request which is not schedulable due to some bins are fully loaded, as shown in Fig. 3c. The main idea of this work is to design static schedules in a way that prevents bin from being overwhelmed too early. In Fig. 3d, if the bins are load-balanced, then the more dynamic request is schedulable.

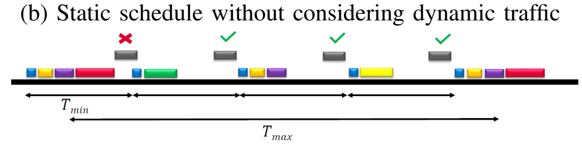
B. PROBLEM STATEMENT

In this work, joint scheduling and routing of flows with network-wide load balancing for TSN's proposed time-aware shaper in zonal architecture is studied and MIP formulation is proposed. Finally, CPLEX is used to solve the MIP formulation. In this work, our primary focus is on the static network. To the best of our knowledge, this would be the first work that takes into account the impact of the static schedules and routes on to the dynamics flows and proposes load-balanced static scheduling and routing for TSN-based in-vehicle networks.

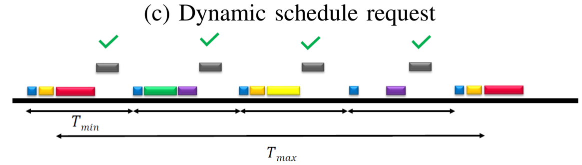
(a) Single egress port switch



(b) Static schedule without considering dynamic traffic



(c) Dynamic schedule request



(d) Load-balanced schedule

Fig. 3: Motivational example

C. MIP FORMULATION

In this section, we present the definition of variables, objective functions and the constraint sets that help in the formulation of MIP. The list of MIP inputs and variables are shown in Table I and Table II respectively.

TABLE I: List of MIP inputs.

Symbol	Definition
$\mathcal{F} \equiv \{f_j\}$	Set of flows to be scheduled. J is the total number of flows. $j \in [1, \ldots, J]$
$\mathcal{P} \equiv \{p_k\}$	Set of possible path through which flow may be routed. This set contains only possible shortest paths from source to destination node for all flows. K is the total number of paths. $k \in [1, \ldots, K]$
$\mathcal{SP} \equiv \{sp_{jk}\}$	Mapping between flows and paths. $sp_{jk} = 1$, if flow j can traverse through path k, else 0.
$\mathcal{PD} \equiv \{pd_k^l\}$	Mapping between path and dimension (switch egress port). $pd_k^l = 1$, if path k includes bin dimension l, else 0. L is the total number of dimensions. $l \in [1, \ldots, L]$
$C^l[i]$	Capacity of dimension l in bin i. I is the total number of bins. $i \in [1, \ldots, I]$
w_j	Size of flow j. The size of flow is calculated as $\frac{paylaod(bits)}{Bandwidth}$
T_j	Periodicity of flow j.

1) Objective function: The first objective is to minimize the absolute deviation between bins ($\mathscr{L}1$-norm) and can be expressed as follows.

$$\text{minimize} \quad \sum_{i=1}^{I} \sum_{l=1}^{L} \mid W^l[i] - \bar{W} \mid \tag{1}$$

Authorized licensed use limited to: University of Nantes. Downloaded on February 09,2025 at 16:24:09 UTC from IEEE Xplore. Restrictions apply.

TABLE II: List of MIP Variables

Symbol	Definition
$x^l_{jk}[i]$	This decision variable is 1, if flow j is traversing through path k and is placed in dimension l of bin i. Otherwise 0.
$y_{jk}[i]$	This decision variable is 1, if flow j is traversing through path k and its all required dimensions are placed in bin i. Otherwise 0
z_{jk}	This decision variable is 1, if flow j is traversing through path k. Otherwise 0.
v^l_{jk}	This decision variable is 1, if flow j is traversing through path k and is placed in dimension l. Otherwise 0.

Where:

- $W^l[i] = \sum_{j=1}^{J}\sum_{k=1}^{K} w_j * x^l_{jk}[i]$ is the load of bin's i in dimension l.
- $\bar{W} = \sum_{i=1}^{I}\sum_{l=1}^{L} \frac{W^l[i]}{I*L}$ is the average load of the bins.

The second objective function is to minimize the maximum load (size of flow) of all bins. That can be expressed as follows.

$$\text{minimize} \quad \max.\ W^l[i], \qquad \forall i \in I, \forall l \in L \tag{2}$$

2) Capacity Constraint: This constraint makes sure that the size of all flows that are placed in dimension l of Bin i and passes through path k do not exceed the capacity of bin's dimension.

$$\sum_{j=1}^{J} w_j * [sp_{jk} * pd^l_k] * x^l_{jk}[i] \leq C^l[i], \qquad \forall i \in I, \forall k \in K, \forall l \in L \tag{3}$$

3) Dimension Constraint: With the aid of this constraint, all required dimensions (Egress ports) corresponding to path k of a flow j is placed in same bin due to no-wait scheduling assumptions [7].

$$\sum_{l=1}^{L} x^l_{jk}[i] = \sum_{l=1}^{L} [sp_{jk} * pd^l_k] * y_{jk}[i], \qquad \forall i \in I, \forall j \in J, \forall k \in K \tag{4}$$

4) Instance Constraint: This constraint makes sure that the number of occurrences of a flow j within the maximum transmission cycle with all required dimensions (Egress Ports) corresponds to path k are met.

$$\sum_{i=1}^{I} x^l_{jk}[i] = [sp_{jk} * pd^l_k] * z_{jk} * \lceil \frac{T_{max}}{T_j} \rceil, \qquad \forall j \in J, \forall k \in K, \forall l \in L \tag{5}$$

5) Path Constraint: This is the routing constraint, and it selects one candidate path of a flow j based on objective function among all possible provided paths.

$$\sum_{k=1}^{K} z_{jk} * sp_{jk} = 1, \qquad \forall j \in J \tag{6}$$

6) Allocation gap Constraint: Place all number of occurrences of a flow j in bins w.r.t its periodicity considering all required dimensions (Egress ports) corresponding to path k.

$$x^l_{jk}[i + b_j] = x^l_{jk}[i], \quad \text{if } i * b_j + 1 < I, \forall i \in I, j \in J, \forall k \in K, \forall l \in L \tag{7}$$

Where, $b_j = \frac{T_j * I}{T_{max}}$ determine the allocation gap between two consecutive instance of a flow.

7) Periodic Constraint: This constraint makes sure that only one instance of a flow j is placed in dimension l of any bins within its periodicity.

$$\sum_{i=1}^{b_j} x^l_{jk}[i + s - 1] = 1 * v^l_{jk}, \quad \text{if } b_j < I, \text{if } s \leq I - b_j + 1, \forall j \in J, \forall k \in K, \forall l \in L, \forall s \in I \tag{8}$$

VI. Experimental Results

In this section, we present the evaluation of MIP-based joint scheduling and routing with the objective of network-wide load balancing for time-aware shaper in zonal architecture. For the experimental evaluation, we use CPLEX v12.0.1, which is the commercial state-of-the-art optimization tool [25]. The experiments were run on 64-bit 56 cores 2.6GHz Intel(R) Xeon(R) CPU with 504GB memory. We examine the output of two objective functions as described in the MIP formulation. The analysis includes the impact of two objective functions on run time performance of the scheduling and routing algorithm and bin utilization of the four ingress ports of the CPU. The analysis of the ingress port of CPU is of vital importance since the CPU performs most of the compute-intensive tasks, i.e. sensor fusion, V2X, V2N, therefore, it is essential not to overload the bins of any ingress ports of CPU.

We run six synthetic configurations in two different scales of zonal architecture. In four zones architecture, we run configurations $1, 2, 3$ and in six zones configurations $4, 5$ and 6 are executed. The minimum period in our network is $5ms$ and the maximum period is $50ms$, that corresponds to 10 bins. Each bin composes of 24 dimensions (i.e., the total number of egress ports in the topology), and each dimension has a capacity of $5ms$. Each ACU transmits and receives integer multiple of 5 flows wherein the periodicities of the five flows are $5ms$, $10ms$, $25ms$, $25ms$, and $50ms$. For example, if we consider configuration 3 (60 flows) wherein four ACU are considered, each ACU transmits $2 \times 5 = 10$ flows to CPU and receives $1 \times 5 = 5$ from CPU. Within these five periodicities, we consider 50 KB of data transmission. The bandwidth of every

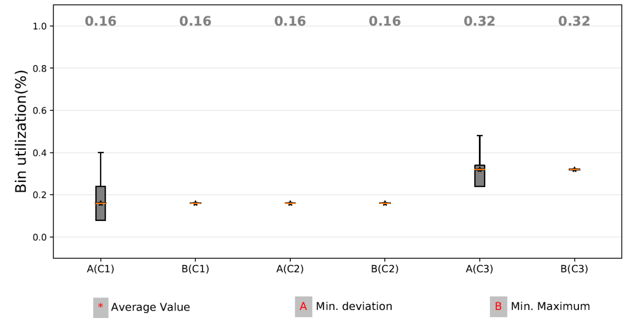
Authorized licensed use limited to: University of Nantes. Downloaded on February 09,2025 at 16:24:09 UTC from IEEE Xplore. Restrictions apply.

link is set as 1 Gbps. From source to destination, we consider only the paths that contain no more than four hops. These limitations in the path set help in reducing the search space as well as bound the delay of flows. The maximum bounded delay of a flow j which traverse through 4 hops is calculated as $\frac{w_j(bytes)}{Bandwidth} \times 4 + T_{sw} + T_{que}$, T_{sw} and T_{que} are switching and queuing delay respectively. For a 50 KB flow, the maximum bounded delay is $\frac{50 \times 10^3 \times 8}{1 \times 10^9} \times 4 \approx 1.6ms$, as switching delay is negligibly small compared to transmission delay and queuing delay is zero due to no-wait scheduling assumption [7]. The details of the configurations are summarized in Table III.

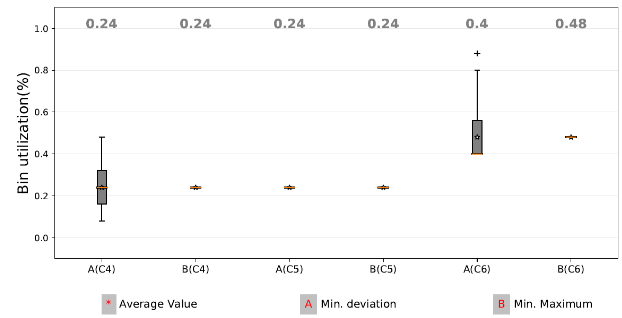
TABLE III: Configuration sets.

Configurations	Total number of flows	Flow direction
Four zones (4 ACUs) architecture		
C1	20	5 flows per ACU send to CPU.
C2	40	5 flows per ACU send to CPU and each ACU receives 5 flows from CPU.
C3	60	10 flows per ACU send to CPU and each ACU receives 5 flows from CPU.
Six zones (6 ACUs) architecture		
C4	30	5 flows per ACU send to CPU.
C5	60	5 flows per ACU send to CPU and each ACU receives 5 flows from CPU.
C6	90	10 flows per ACU send to CPU and each ACU receives 5 flows from CPU.

We use the box plot to show the bin utilization of four aggregated ingress ports of CPU with two objective functions, namely minimizing maximum load (MinMax) and minimizing deviation (MinDev). For the scalability evaluation, we measure the run time of CPLEX while optimizing the schedules and routes jointly. Experimental results show that MinMax performs better than MinDev in terms of load distribution, as shown in Fig. 4. In all of the configurations, MinMax distributed the load equally among all the bins of four ingress ports of the CPU. On the other hand, in MinDev the load is scattered, and some of the bins are loaded less while others are loaded more, as shown in minimum and maximum value of configuration 4 in Fig. 4b. To give more detail about the results, e.g. in configuration 6, MinDev provides worse results when the load is increased, in Fig. 4b, in which one of the bin utilizes 88% (see outlier) of total capacity. In contrast, all of the bins in MinMax utilize 48% of the capacity. The results of MinDev and MinMax in configuration 2 (Fig. 4a) and configuration 5 (Fig. 4b) are similar this is due to the fact that in that configurations the flows are distributed equally i.e. 50% flows to CPU and 50% from CPU. From the run-time aspect, MinDev provides better results as compare to MinMax since it takes lesser time to find the optimal solution for all configurations, as shown in Fig. 5.

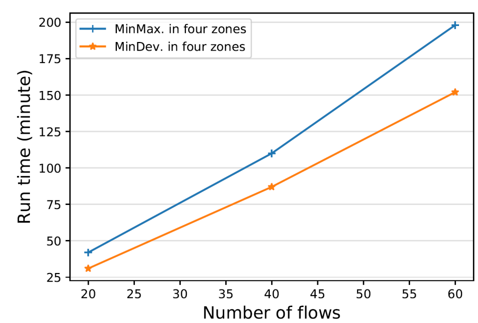


(a) Four zones

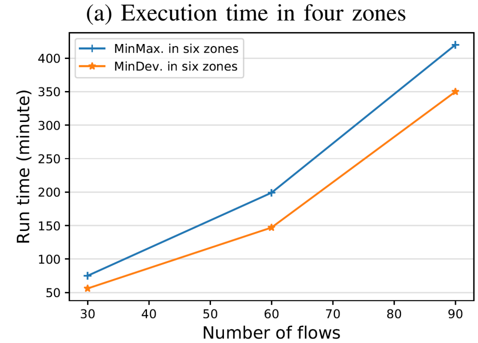


(b) Six zones

Fig. 4: Bin utilization of two objective functions



(a) Execution time in four zones



(b) Execution time in six zones

Fig. 5: Execution time

VII. Conclusion and Future Work

In this paper, we have designed MIP-based joint static scheduling and routing for TSN's time-aware shaper in two scales of zonal architecture. We transformed the problem

Authorized licensed use limited to: University of Nantes. Downloaded on February 09,2025 at 16:24:09 UTC from IEEE Xplore. Restrictions apply.

into vector bin packing and proposed that with the aid of load balancing more dynamic traffic will be schedulable. In this regard, we have examined two load balancing based objective functions for joint scheduling and routing. Our results show that by taking into account the load balancing during optimization of scheduling and routing, we can cater more dynamic flows efficiently. The objective function (MinMax) outperforms other objective function (MinDev) in terms of balanced bin utilization despite requiring more time to optimize. As this work is mainly designed for static applications, run-time performance is of secondary importance for our work. In addition, we will further continue to investigate scheduling and routing algorithms for dynamic applications that reuse the results of MinMax objective function of static applications.

Acknowledgment

This work was partially funded by the Federal Ministry of Education and Research (BMBF) under KI-PRO project with funding code FKZ:16ES1004.

References

[1] Nicolas Navet, Francoise Simonot-Lion, "Automotive Architectures" in Automotive Embedded Systems Handbook. United States of America: CRC Press Taylor and Francis Group, 20 Dec. 2008, ch. 1, pp. 7–15.

[2] Maryam Pahlevan, Nadra Tabassam and Roman Obermaisser, "Heuristic List Scheduler for Time Triggered Traffic in Time Sensitive Networks," ACM SIGBED Vol. 16, issue 1, February 2019.

[3] "IEEE 802.3 Greater than 10 Gb/s Automotive Ethernet Electrical PHYs Study Group," 2019. [Online]. Available: http://www.ieee802.org/3/B10GAUTO.

[4] "Multi-Gig Automotive Ethernet PHY Task Force," IEEE, 2018. [Online]. Available: http://www.ieee802.org/3/ch/.

[5] Intel Newsroom, "Data is the New Oil in the Future of Automated Driving," 2016 [Online]. Available: https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving.

[6] Ahmed Nasrallah, Akhilesh S. Thyagaturu, Ziyad Alharbi, Cuixiang Wang, Xing Shao, Martin Reisslein, Hesham Elbakoury, "Performance Comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS)," IEEE Access Vol. 7, pp. 44165-44181, April 3, 2019.

[7] Frank Dürr, Naresh Ganesh Nayak, "No-wait Packet Scheduling for IEEE Time-sensitive Networks (TSN)," in Proceedings of the 24th International Conference on Real-Time Networks and Systems, pp. 203–212, October 2016.

[8] Fedor Smirnov, Michael Glaß, Felix Reimann, Jürgen Teich, "Optimizing message routing and scheduling in automotive mixed-criticality time triggered networks," in 54th Design Automation Conference (DAC), pp.1-6, June 2017.

[9] Silviu S. Craciunas Ramon Serna Oliver Martin Chmelík Wilfried Steiner, "Scheduling Real-Time Communication in IEEE 802.1 Qbv Time Sensitive Networks," in 24th International Conference on Real-Time Networks and Systems (RTNS), October 2016.

[10] Paul Pop, Michael Lander Raagaard, Silviu S. Craciunas, Wilfried Steiner, "Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks," IET Cyber-Physical Systems: Theory and Applications vol. 1, December 2016.

[11] Eike Schweissguth, Peter Danielis, Dirk Timmermann, Helge Parzyjegla, G. Mühl, "ILP-based joint routing and scheduling for time-triggered networks," in Proceedings of the 25th International Conference on Real-Time Networks and Systems, pp. 8–17, October 2017.

[12] Naresh Ganesh Nayak, "Scheduling and Routing Time-triggered Traffic in Time-sensitive Networks," Ph.D. dissertation, Graduate School of Excellence advanced Manufacturing Engineering, University of stuttgart, November 2018.

[13] Qinghan Yu, Hai Wan, Xibin Zhao, Yue Gao, and Ming Gu, "Online Scheduling for Dynamic VM Migration in Multicast Time-Sensitive Networks," IEEE Transactions on Industrial Informatics, vol. 16, no. 6, June 2020.

[14] M.R Garey, R.L Graham, D.S Johnson, Andrew Chi-Chih Yao, "Resource constrained scheduling as generalized bin packing," Journal of Combinatorial Theory, Series A Vol. 21, Issue 3, November 1976, Pages 257-298.

[15] Panigrahy R, Talwar K, Uyeda L, Wieder U, "Heuristics for vector bin packing," Technical report, Microsoft Research, 2011.

[16] Sadiq M. Sait, Abubakar Bala, Aiman H El-Maleh, "Cuckoo search based resource optimization of datacenters," Applied Intelligence vol. 4 no.3 , April 2016.

[17] Lawrence T. Kou, George Markowsky, "Multidimensional Bin Packing Algorithms," IBM Journal of Research and Development vol. 21, issue 5, pp. 443 - 448, October 1977.

[18] S.Y. Chang, H. Hwang, Sanghyuck Park, "A two-dimensional vector packing model for the efficient use of coil cassettes," Computers and Operations Research Vol. 32, Issue 8, pp. 2051-2058, August 2005.

[19] H. Walder, C. Steiger, M. Platzner, "Fast Online Task Placement on FPGAs: Free Space Partitioning and 2D-Hashing," in Proceedings International Parallel and Distributed Processing Symposium, April 2003.

[20] Michaël Gabay, Sofia Zaourar, "Vector bin packing with heterogeneous bins: application to the machine reassignment problem," Annals of Operations Research vol. 242, pp. 161–194, 2016.

[21] E. Arzuaga, D. Kaeli, "Quantifying load imbalance on virtualized enterprise servers," in Proceedings of the first joint WOSP/SIPEW International Conference on Performance Engineering, San Jose, California, USA, January 28-30, 2010.

[22] Voica Gavrilut, Luxi Zhao, Michael l. Raagaard, and Paul Pop, "AVB-Aware Routing and Scheduling of Time-Triggered Traffic for TSN," IEEE Access vol. 6, pp. 75229-75243, November 2018.

[23] Johannes Specht, Soheil Samii, "Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks," 28th Euromicro Conference on Real-Time Systems, 2016.

[24] Statista, "Automotive electronics cost as a percentage of total car cost worldwide from 1970 to 2030," 2020 [Online]. Available: https://www.statista.com/statistics/277931/automotive-electronics-cost-as-a-share-of-total-car-cost-worldwide/.

[25] CPLEX Optimizer. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

Authorized licensed use limited to: University of Nantes. Downloaded on February 09,2025 at 16:24:09 UTC from IEEE Xplore. Restrictions apply.