Approximation of Worst-Case Traversal Times in Real-Time Ethernet Networks: Exploring the Potential of Many-Objective Optimization for Simulation Aggregation

Patrick Keller
Faculty of Science, Technology and Medicine
University of Luxembourg
Esch-sur-Alzette, Luxembourg
patrick.keller@uni.lu

Nicolas Navet
Faculty of Science, Technology and Medicine
University of Luxembourg and Cognifyer
Esch-sur-Alzette, Luxembourg
nicolas.navet@uni.lu

***Abstract*—Simulation is an important tool for the verification of modern complex time-critical communication systems, especially when worst-case schedulability analysis is not available. Evaluating worst-case traversal times via simulation traditionally involves resource-intensive long simulations that are poorly parallelizable. Recent research has demonstrated that aggregating many short simulations with randomized starting conditions yields substantial improvements over this classical approach in terms of likelihood of observing very large communication latencies. In this study, we explore the potential of many-objective optimization to further enhance the efficiency of the aggregation approach.**

To this end we further reduce the length of the aggregated simulations and perform many-objective optimization to set the starting conditions, namely the node start offsets and initial flow scheduling order. Our approach consists in modelling the approximation of worst-case traversal times as a many-objective Pareto optimization problem in the context of real-time Ethernet networks. Performance evaluation, conducted on different industrially relevant use cases from the automotive and aerospace domains, shows up to 46.42% increased end-to-end latencies for a 50 times shorter total simulation time, in comparison to the traditional approach of running single long simulations.

***Index Terms*—Performance evaluation, simulation, timing verification, worst-case traversal times, Ethernet, simulation aggregation.**

I. Introduction

Context of the work: Recent research has explored the potential of aggregated short simulations using a stratified random search approach on node start offsets (NSO) for approximating worst-case latencies in static real-time networks with fixed traffic known at design time [5]. Significant improvements over long simulations with favorable starting conditions were observed, but the full potential of this new verification and validation paradigm remains largely unexplored. In this work, we explore the potential of more advanced techniques, namely multi-objective optimization, that allow leveraging the underlying interference patterns to increase the probability of observing "critical instants" leading to large end-to-end communication delays. More specifically, we model the approximation of the worst-case traversal times (WCTT) as an optimization problem and apply a popular multi-objective algorithm. We chose "Non-dominant Sorting Genetic Algorithm II" (NSGA2, see [1]), as this simple and well understood algorithm has been successfully used for many studies in various domains and applications.

Applications: Approximating Worst-Case Traversal Times (WCTT) in Ethernet networks becomes an increasingly relevant task as the complexity of modern communication architectures and the applications they support grow, and formal timing verification does not cover all the use-cases anymore. For instance, to the best of our knowledge, there is no worst-case analysis for the TCP-based flows, increasingly present in complex real-time systems such as vehicles. However, validation via simulation has some drawbacks that need to be addressed. For instance, it only provides an inaccurate lower-bound on the actual WCTT [11]. One aim of this work is to push the boundaries of how close we can get to the actual WCTT while limiting ourselves to using a reasonable amount of computational resources, as available in most engineering contexts. Another application of this work is design automation, as the results presented allows for a quicker and more reliable preliminary evaluation of candidate solutions compared to standard simulation approaches.

WCTT approximation via aggregation of short simulations: Recent research has found aggregation of short simulations to be an effective alternative to the classical approach of running long simulations with specific starting conditions [3]. A big advantage of the aggregation approach is that it can be parallelized easily and at a massive scale. Another significant advantage of the approach is that it allows for more versatility and, thus, the usage of advanced algorithms to explore the design parameters more optimally, as investigated in this work.

Contributions: The contributions of this paper are threefold. The first contribution is the evaluation of how efficient simulation aggregation performs for WCTT approximation when minimizing the simulation time. The second contribution is a proposal how the worst-case traversal time approximation can be modeled as a Pareto multi-objective optimization problem. Using NSGA2 as a reference algorithm, the potential of many-objective optimization is then evaluated on five diverse industrially relevant network configurations of different complexities. The final contribution is an analysis of the optimization overhead, which is evaluated based on the same experiments.

II. Related prior work

The effectiveness of aggregation of short simulation for approximating worst-case latencies in the context of Ethernet-based real-time networks was first explored in 2022 by Keller and Navet in [3]. Limited but significant improvements could be observed over the traditional approach of running long simulations using synchronized node start offsets (NSO) and randomized node clock drifts (CD). The work investigated how random search could be applied to explore the search space of node start offsets (NSO), which can significantly impact the observed traversal times if chosen well. Many open questions about the aggregation approach for evaluating WCTT in real-time Ethernet traffic remain unanswered, such as how to set the simulation time of short simulations and how to better select node start offsets.

The work on simulation aggregation by Keller and Navet was inspired by a comparable approach to maximize observed worst-case response times for a specific process in bus-based networks using CAN and FlexRay as presented in 2008 by Samii et al. in [4]. They apply expert knowledge on bus-systems to reduce the execution-time space of the various processes involved and apply different optimization algorithms, including genetic algorithms, to find the maximal response time for a certain process. The system model applied in this work significantly differs from ours as it focuses on bus-based systems and execution times of processes rather than the traffic interaction in Ethernet networks, which is the focus of our system model.

The open question of how to set the simulation time and node start offsets in the short simulation aggregation approach was addressed in 2023 by Keller and Navet in [5]. To set the simulation time, they proposed a pretest to determine the speedup factors for a range of simulation times and suggest to select a possibly small time that roughly maintains the same speedup as longer simulations. For selecting the node start offsets they suggest to apply an overlapping stratified sampling approach on the node start offset range. It means that the values are drawn in equal parts from multiple ranges, starting from a narrow range around the synchronized case and increasing exponentially, leading to a reasonable trade-off of exploration and exploitation. The results conclude a significant increment of observed maximal traversal times over naively sampling NSO from a single fixed range.

The controlling of node start offsets to generate critical instants was inspired heavily by findings made in 2010 and 2011 by Bauer, Scharbarg and Fraboul in [7] and [8]. They present a trajectory-based analytical approach to generating unfavorable scenarios that yield very high traversal times in Ethernet-based real-time networks limited to sporadic flows using first-in-first-out (FIFO) and fixed priority (FP/FIFO) traffic mechanisms. The approach has shown to be effective in approximating worst-case latencies in [12] by Boyer et al. in 2012.

The conveyed understanding of how high latencies scenarios can be generated helped motivate the stratified approach [5] in that the inter-flow offsets are generally concentrated around multiple hotspots. The work further inspired this current paper as optimization can be seen in this context as a generic approach for exploring and optimizing towards such unfavorable scenarios in the generic case.

As this current work relies on multi-objective optimization, more specifically multi-objective evolutionary algorithms (MOEAs), we want to highlight a 2021 survey [9] by Tian et al. on the topic. They provide an overview of the state-of-the-art MOEA algorithms, discuss their respective strengths and weaknesses and provide an overview on the benchmarking of MOEAs.

For our approach, NSGA2, introduced in 2002 by Deb, Pratap, Agarwal and Meyarivan in [1], was chosen due to its popularity, simplicity and wide availability of implementations. The algorithm is briefly introduced in the second research question section in the context of our application.

III. System Model

The system model is typical for static Ethernet-based real-time networks. It consists of a topology comprising network bridges, end-nodes and communication links. The network bridges forward traffic packets on the network, also known as switches. The end-nodes are networked devices, and are known under different names depending on the application domain, for instance end-systems, stations or ECUs. They typically provide computation, sensing or actuation capabilities, and communicate with other end-nodes on the network. The links in our model are full-duplex Ethernet connections that support speeds of 100, 1000 or higher Mbit/s. The network traffic is known at design time and is referred to as flows in this work, which specify the traffic characteristics. A traffic flow is defined by a transmission type of sporadic, periodic, periodic burst or TFTP using either a fixed or a minimal sending frequency, and a fixed predefined data packet size. The routing of the different flows is also determined at design time and does not change during the execution of the system. Our network model further includes several free parameters, such as node start offsets (NSO), frame offsets (FO) and node clock drifts (CD). The NSO define for each node when it starts initially sending data after the origin of time t_0, at which the system is started. The FO define a time displacement of the initial frame belonging to a flow in relation to the node start offset, when it becomes ready for sending data. CD defines a drift for each end-node compared to the system reference clock. A node with a positive clock drift will process and emit packets faster. In our case, they can amount to up to 200ppm, which is deemed acceptable in automotive scenarios [11]. To retain the genericity of the system model, we assume that node clocks are not synchronized.

A. Terminology

- **Simulation time** is the time elapsed inside the simulated system.
- **Simulation duration** is the time the simulation runs on the computer, also known as wall-clock time.
- **Multi-/Many-objective optimization** is used solve optimization problems with multiple competing objectives, which are referred to as many-objective when the number of objectives exceeds four.
- **Worst-Case Traversal Time (WCTT)** is the highest end-to-end latency (or delay) that can possibly be observed for a traffic flow.
- **Stratified Sampling** refers to sampling a value from different ranges. In this work, the node start offsets are sampled from five overlapping exponential strata (layers) $[0, x^n]$ for $n \in 1..n$, with x^n being the selected maximum range for the NSO.

B. Traversal Times

Traversal times, or more generally known as end-to-end latencies, are a property of traffic flows resulting from delays that the flow packet faces while transmitted from the sender to the receiver node along its predefined routing. Traversal times can be influenced by many factors. Some of these factors are free and can be adjusted for simulation and some are fixed and predetermined by the network configuration. Fixed factors that must not be changed, include the network topology, the link speeds, and the traffic characteristics, such as flow type, packet size, routing, etc.. Free factors, which can be adjusted at each simulation run without changing the validity concerning the modeled system, include the following:

- **Node Start Offsets:** They define the offset of when an end-node initially starts sending data after the system was started at the origin of time t_0. The node start offsets significantly impact when packets of different data flows interfere with each other at the beginning of the simulation.
- **Flow Scheduling Order:** In our model, flows are initially scheduled simultaneously after the node start offset has elapsed. The sending order is thus typically randomly selected based on the simulation random seed. As our simulation software does not support explicitly setting the flow scheduling order at the time of writing, we achieve this by means of setting very minor frame offsets (FO). Frame Offsets define a delay that is added to the node start offset of the sender node of a flow. It impacts when the first packet of a flow is scheduled on the sender node. This parameter is normally used in synchronized systems like, for instance, when using Time-Aware Shaping. Adjusting them would typically invalidate our simulation as the traffic characteristics would change. To avoid this invalidation, we select and optimize frame offsets in the low nanosecond range, which in the context of our experiments only impact the flow scheduling order without measurably impacting the traffic characteristics.
- **Clock Drifts:** They define how the clock of an end-node changes over time relative to a reference system clock. Randomized clock drifts are a crucial part of the traditional solution of running long simulations, as they enable gradual exploration of the simulation state space. In this study, clock drifts are uniformly sampled from the $[0, 200]$ppm range for every node in each simulation run.
- **Random seed of the simulator:** It can be set to make random factors in the system deterministic to achieve reproducible simulation results. Random factors that are influenced by this seed include, for instance, switching delays, delay between packets of sporadic flows and the flow scheduling order in case two flows try to schedule a packet at the same time. Simultaneously scheduled flows occur, for instance, when all flows originating from the same node are initially scheduled once the NSO time has elapsed.

C. Implementation details

We use the Java based software RTaW-Pegase [10] in version 4.3.8.4 to run our simulations. To implement our experiments, we use Python in version 3.10.5 and use the optimization algorithms provided by the pymoo library [6] in version 0.6.0.1. The NSGA2 implementation provided by this library has shown to be applicable even to our largest use case, which consists of over 3000 objectives. The Python implementation of our experiments communicates via sockets with a custom simulation service implemented using the Java library provided by RTaW-Pegase to execute the simulations.

D. Use cases

Our use cases comprise five configurations based on three different industrially relevant network topologies from the aerospace and automotive domains. Each configuration features unique traffic properties and different QoS mechanisms. The automotive topology we apply is depicted in Figure 1, the avionics AFDX topology in Figure 2, and Figure 3 depicts a space launcher topology. For each topology, a configuration using fixed priorities (denoted as "FP/FIFO") for traffic management is considered, and for the automotive and avionics topologies, additionally a configuration based on first-in-first-out (denoted as "FIFO"). Further traffic characteristics of the configurations, together with some statistics on the topologies, are summarized in Table I.

TABLE I
CHARACTERISTICS OF THE NETWORK CONFIGURATIONS (DERIVED FROM [3]).

Topology	# nodes	# switches	# links	# flows	# flow receivers	QoS mecha.	Flow types
Space	18	18	24	100	985	4 Priorities	21 Command and Control (periodic) 78 Telemetry (periodic) 1 Video (periodic burst)
Avionics	52	4	57	453	3214	5 Priorities	453 Uncategorized (sporadic)
						FIFO	
Automotive	14	5	18	58	70	4 Priorities	19 Command and Control (periodic) 10 Audio (periodic) 11 Video (periodic burst) 6 Best Effort (periodic) 4 TFTP (each ACK+DAT+RRQ)
				46	58	FIFO	46 (sporadic, no TFTP traffic)

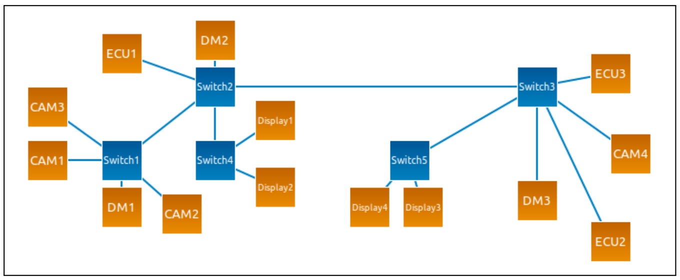

Fig. 1. Automotive topology. Illustration from [3].

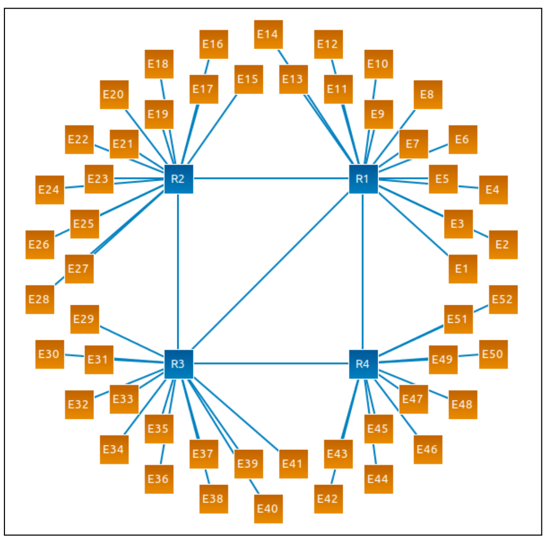

Fig. 2. Avionics topology. Illustration from [3].

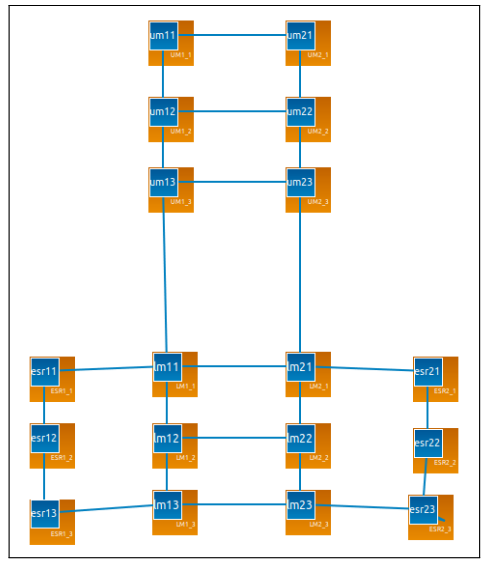

Fig. 3. Space launcher topology. Illustration from [3].

IV. RESEARCH QUESTIONS

We designed research questions to individually investigate the potential of minimizing the simulation time in simulation aggregation, the addition of optimization of node start offsets with and without scheduling order, and the overhead of the multi-objective optimization with respect to the resources spent on simulation. For all experiments, stratified sampling is used to determine the NSO, as it is a cost-free method to boost the observed WCTT. For optimization, the initial population is determined by stratified sampling.

We formulate the research questions as follows:

- RQ1. Can we leverage the exploratory power of shorter simulations by minimizing the simulation time beyond a threshold that would maintain the speedup factor?
- RQ2. Can multi-objective Pareto optimization algorithms improve the exploration of the node start offset search space in the approach of aggregating short simulations for approximating worst-case traversal times via simulation?
- RQ3. What is the overhead cost of performing the optimization compared to resources invested in simulations?

Evaluation Metric: The metric we apply for evaluating the WCTT approximation is the sum of the cumulative maximum per flow. As discussed in the previous section, no single solution typically exists that maximizes the WCTT for all flows simultaneously. For that reason, the cumulative maximum is used in our metric to reflect the action of retaining the best solutions per flow during an experiment. Technically, the goal is maximizing the WCTT for every flow, but this is impractical for evaluation and visualization as the number of flows can reach many thousands. This means we retain the highest observed end-to-end latency value per flow across all simulation runs up to a certain point of a given experiment and compute the sum of those maximal latencies as a metric. We will refer to this metric as "aggregated maximal traversal times" (AMTT). The higher this value, the better our approximation of the WCTT.

Experiment Types: Five types of experiments are conducted for each configuration. A single long simulation with 100 hours of simulation time (denoted by "single long"), two aggregation experiments with different simulation times ST and stratified sampling for the NSO (denoted by "random short" and "random minimized"), and two aggregation experiments with minimized simulation times and optimization of NSOs (denoted by "opti. NSO"), and, optionally, frame scheduling order ("opti. NSO+FO").

The simulation time is minimized to a value such that the simulation still produces an end-to-end delay for every flow when applying the maximal value to be considered for the NSO. The maximal NSO range for the stratification sampling is set to be half of the simulation time.

Result representation: The results of the different experiments for RQ1 and RQ2 are combined in Figure 4 and in Table II for simplified comparison. The results for RQ3 are summarized in Table III. All results reported represent the average over at least ten experiment repetitions to increase statistical significance.

Figure 4 depicts the experiment results for the space launcher configuration. The figures for the other use cases show similar trends but are omitted for brevity, and the key information is summarized in Table II instead. The figure shows the evolution of the aggregated maximal traversal times metric (AMTT) on the y-axis with respect to the simulation duration (wall-clock time) on the x-axis. The long simulation is drawn as a dashed horizontal line as no intermediate results are retained, and only the final flow latencies are reported. Table II summarizes the results for all configurations. The column denoted "ST" shows the simulation time per simulation, "EST" shows the total simulation time for a single experiment repetition, "SD" denotes the average simulation duration for a single experiment repetition, "speedup" denotes the simulation speedup factor resulting from the EST/SD columns and "AMTT" denotes our evaluation metric, the aggregated maximal traversal times.

Context: In this study, we primarily aim to investigate the exploratory power of simulations with very short simulation time. However, reducing the simulation time (ST, the time elapsed inside the simulation) causes the initialization cost to gain increasing importance with respect to the overall computational resources. This can be observed in Table II from the reduced speedup factors. Therefore, for RQ1 and RQ2, we focus on pure simulation time, which includes the time to initialize and run the simulation.

The optimization overhead cost, which highly depends on the choice of the optimization algorithm and on the configuration complexity, is separately investigated in RQ3. As shorter simulations become less efficient as the simulation time decreases, due to the increased relevance of the initialization cost, we chose two hours of total experiment simulation time for the experiments with minimized simulation time. This results in a total simulation duration (SD, wall-clock time) comparable to the long simulation in most cases, as can be seen in Table II.

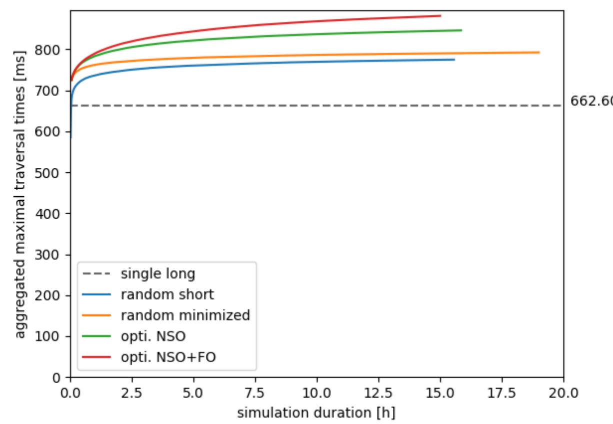

Fig. 4. Space Launcher - Results of the different experiments for RQ1 and RQ2 (x-axis, in ms, shows up to 20 h simulation duration) for the space launcher configuration with fixed priorities. The y-axis represents our "aggregated maximal traversal times" metric (AMTT) in milliseconds.

A. Research Question #1 - Minimized simulation time

In this research question, we investigate how aggregated simulations with minimized simulation time perform compared to single long simulations and aggregated short simulations. The aim is to investigate the effect of the increased exploratory potential on the observed end-to-end latencies. By reducing the simulation time (ST) to the low millisecond range, more simulations can be run for the same simulation time budget, which increases the opportunity for exploration by a factor of up to a few thousand compared to simulations of multiple seconds, as in [3], for instance.

Results: We observe that both aggregation of simulations with short (denoted as "random short") and minimized (denoted as "random minimized") simulation time (ST) yield, on average, better results on our metric (AMTT) than long simulation (denoted as "single long") on all use cases and up to $+46.42\%$. Additionally, minimized ST performs better than short ST in all cases except for the automotive configuration with FP/FIFO, where the metric AMTT of minimized ST is 0.52% lower. It should be noted that the automotive FP/FIFO also has the highest ST and, thus, less opportunity for exploration compared to the other cases.

B. Research Question #2 - Optimization

This research question describes how the WCTT approximation can be modeled as a many-objective Pareto optimization problem and aims to analyze the performance of optimizing

TABLE II
EXPERIMENT RESULTS SUMMARY FOR RQ1 AND RQ2.

Configuration	experiment	ST[a]	EST[b][h]	SD[a][h]	speedup	AMTT[c][ms]	wrt. single long[d]
Automotive FIFO	single long	100 h	100.0	5.29	18.869	24.41	
	random short	225 s	100.0	5.61	17.821	28.44	+16.51%
	random minimized	8 ms	2.0	5.65	0.354	31.58	+29.37%
	opti. NSO	8 ms	2.0	5.18	0.386	34.61	+41.79%
	opti. NSO+FO	8 ms	2.0	5.34	0.375	35.74	+46.42%
Automotive FP/FIFO	single long	100 h	100.0	6.83	14.655	257.18	
	random short	56.3 s	100.0	7.88	12.682	261.62	+1.73%
	random minimized	30 ms	2.0	0.91	2.197	260.35	+1.23%
	opti. NSO	30 ms	2.0	0.85	2.353	267.78	+4.12%
	opti. NSO+FO	30 ms	2.0	0.84	2.380	268.82	+4.53%
Space Launcher	single long	100 h	100.0	11.4	8.809	662.60	
	random short	450 s	100.0	15.6	6.425	774.89	+16.95%
	random minimized	18 ms	2.0	19.0	0.105	792.56	+19.61%
	opti. NSO	18 ms	2.0	15.9	0.126	846.37	+27.73%
	opti. NSO+FO	18 ms	2.0	15.0	0.133	881.79	+33.08%
Avionics FIFO	single long	100 h	100.0	34.0	2.942	4999.08	
	random short	450 s	100.0	43.9	2.277	6116.37	+22.35%
	random minimized	14 ms	2.0	34.6	0.058	6289.82	+25.82%
	opti. NSO	14 ms	2.0	29.0	0.069	6670.35	+33.43%
	opti. NSO+FO	14 ms	2.0	29.0	0.069	7118.70	+42.40%
Avionics FP/FIFO	single long	100 h	100.0	34.0	2.940	4073.57	
	random short	900 s	100.0	46.9	2.130	4660.83	+14.42%
	random minimized	16 ms	2.0	32.7	0.061	4780.18	+17.35%
	opti. NSO	16 ms	2.0	28.4	0.070	5079.91	+24.70%
	opti. NSO+FO	16 ms	2.0	28.4	0.070	5220.15	+28.15%

[a] ST: simulation time (time elapsed inside the simulation); SD: simulation duration (wall-clock time)
[b] EST: total experiment simulation time
[c] AMTT: "aggregated maximum traversal time" metric
[d] wrt. single long: performance in comparison to the single long reference simulation; higher is better

the node start offsets (NSO) in isolation, and in combination with optimizing flow order (NSO+FO). We briefly describe NSGA2, the optimization algorithm of our choice, without going into much detail as it is not the focus of the study. We further describe the structure of the problem and the essential chosen optimization parameters.

To be able to compare the optimization results to the random search approach, we report the results in terms of total experiment simulation time (EST), excluding the optimization overhead cost. The optimization overhead will be analyzed and discussed separately in the third research question.

Modelling as an optimization problem: Approximating the worst-case traversal times can be modeled as a many-objective Pareto optimization problem. The goal of the problem is to maximize the observed traversal time of every individual traffic flow, hence each flow traversal time is an objective of the problem. The design variables can include any free variables that influence the traversal times in the simulation. In our case, we focus on the node start offsets (NSO) as they are highly important in generating high traversal times, as discussed before. We further conduct experiments (NSO+FO) that additionally optimize the flow scheduling order by the means of optimizing minor frame offsets in the nanosecond range. This allows us to impact the flow scheduling order without invalidation of the model. Finding a distinct set of node start offsets might be necessary to maximize the worst-case traversal time for every flow, rendering it a Pareto problem.

To model this problem, our input variables are defined by the number of NSO we can set for a simulation, which is the number of nodes in our network architecture. It is increased by the number of flows when additionally optimizing for the flow scheduling order. The number of objectives to optimize for is the number of traversal times that a simulation produces. It is equivalent to the number of flow receptions of our configuration. As our system model supports multi-cast flows, it means that the number of traversal times can be significantly higher than the number of flows in the configuration.

As an optimization algorithm, we chose NSGA2 [1], which is simple to understand and has proven useful in many scientific studies of various fields. It combines the classic genetic algorithm with a non-dominated sorting and survivor selection via crowding-distance to maintain the diversity of solutions. We apply mostly standard parameters appropriate for our application.

Results: For this research question, we specifically compare the results of the random search using the minimized simulation time (denoted as "random minimized" in the figures and tables), the optimization of NSO (denoted as "opti. NSO"), and additionally optimizing the initial flow scheduling order (denoted as "opti. NSO+FO").

As we can observe, the optimization approach ("opti. NSO" and "opti. NSO+FO") shows advantages over the random search approach ("random short") across all of our use cases.

TABLE III
RESULTS OF OPTIMIZATION OVERHEAD EVALUATION OF RQ3.

Configuration	experiment	exp. duration ED [h]	PSD[a][h]	SD[a][h]	overhead[b]
Automotive FIFO	opti. NSO	0.706	0.432	5.18	+63.55%
	opti. NSO+FO	0.729	0.445	5.34	+63.82%
Automotive FP/FIFO	opti. NSO	0.093	0.071	0.85	+31.29%
	opti. NSO+FO	0.092	0.070	0.84	+31.43%
Space Launcher	opti. NSO	1.903	1.325	15.9	+43.62%
	opti. NSO+FO	1.830	1.250	15.0	+46.40%
Avionics FIFO	opti. NSO	8.121	2.418	29.0	+235.93%
	opti. NSO+FO	7.983	2.418	29.0	+230.22%
Avionics FP/FIFO	opti. NSO	7.364	2.367	28.4	+211.15%
	opti. NSO+FO	7.255	2.367	28.4	+206.55%

[a] PSD: parallel simulation duration (SD); time to run in parallel on 12 cores
[b] overhead: time spent on optimization rather than simulation := $(ED - PSD)/PSD$

We observe an improvement of up to an additional 46.42% over long simulations (”single long”), and thus, an additional improvement of $+29.91\%$ over aggregation of short simulations (”random short”). Further, we can observe that optimizing the initial flow scheduling order in addition to the node start offsets (”opti. NSO+FO”) yields further relative improvements in most use cases, up to an additional 8.97% increase over only optimizing NSOs (”opti. NSO”).

C. Research Question #3 - Overhead cost of optimization

In this final research question, we aim to evaluate the overhead cost of the optimization algorithm. Previously, we excluded the optimization overhead to compare the results to the traditional approach. Evaluating the overhead, however, is tricky as it depends on many factors and is not paid at every simulation run but mainly after each epoch and further depends on the specific optimization algorithm.

Each experiment’s total duration (ED) and the total simulation duration (SD) are collected to evaluate this cost. Since our implementation uses parallelism, the total per-core simulation duration (SD) must be divided by the parallelization factor of 12, yielding the parallel simulation duration (PSD), as shown in Table III. The PSD represents the part of the experiment duration spent on running simulations. Since perfect parallelism is assumed, the overhead cost we compute is an upper bound and thus slightly pessimistic concerning the actual overhead, as some of it stems from minor synchronization needs.

Results: To evaluate the optimization overhead cost of our approach, we report the total simulation duration, the total experiment duration and the proportions of simulation and optimization in Table III. The values reported include the total experiment duration (ED) per repetition, the parallel simulation duration (PSD), the sequential simulation duration (SD), and the overhead of the computed optimization as a percentage in relation to the PSD.

As we can see from the results, the overhead for the avionics configurations is significantly larger, up to $+207\%$ to $+236\%$ of the duration used on simulation. The main reasons for this are the increased complexity in the size of the population and the number of objectives. These factors increase the complexity of generating new solutions and performing the non-dominated sorting, which grows linearly in the number of objectives and quadratically in the population size for NSGA2 [1]. The population size is typically chosen proportional to the number of objectives and thus increases with the number of end-to-end latencies considered. The increased number of variables, however, seems less important as optimizing flow orders does not significantly impact the experiment duration.

For the automotive and space use cases, we still observe a significant overhead of 31.29% up to 63.55% of the resources spent on the simulation on average, but even then, the benefits outweigh the overhead cost as observing higher end-to-end latencies typically becomes exponentially less likely the closer the true WCTT is approximated.

V. CONCLUSION

In this work, we have explored the efficiency of simulation aggregation when applying a minimized simulation time that is still viable, which is in the milliseconds for our use cases. Despite yielding a lower speedup factor compared to longer simulations, we conclude that aggregation of simulations with minimal simulation time is competitive to the traditional approach of long simulations and to the aggregation of longer short simulations.

Further, we describe how the WCTT approximation can be modeled as a many-objective optimization problem, using a biased generation of the initial population inspired by the stratified sampling of the NSO. The optimization approach is evaluated on our use cases using NSGA-II as a sample algorithm. In our experiments, we observed up to 46.42% increased end-to-end latencies compared to single long simulations with a total simulation time reduced by a factor of 50. Achieving higher traversal times in simulation becomes exponentially harder the closer we approach the true WCTT. Thus, we conclude that the optimization approach is superior in terms of resource efficiency and/or lower-bound tightness compared to previously explored simulation-based approaches to approximating WCTTs. Improving the efficiency of extremely short simulations to maintain a reasonable speedup factor remains a technical challenge to be solved.

We end the paper with an analysis of the overhead of the optimization approach, which has shown to be rather high, up to $+236\%$ of the simulation duration, for more complex network configurations due to the nature of the chosen optimization algorithm. Despite this overhead cost, we conclude that the approach is promising and could potentially be leveraged further by technical advances such as, for instance, simulators that better support shorter simulations and thus maintain higher speedup factors, or application of optimization algorithms that are better suited for high numbers of variables and objectives.

A. Parallelizability

The simulation aggregation approach, in general, is highly parallelizable as simulations are independent of each other. The introduction of optimization slightly reduces the parallelizablilty but can still be realized in different ways. All simulations performed during the same epoch can be run in parallel, thus larger population sizes could be beneficial, given a scalable optimization algorithm. Beyond that, co-evolutionary techniques would allow for an increase of this parallelization factor, for instance, an ”island model” as suggested in [2].

B. Perspectives

Further research and technical developments may enhance the efficiency and the generality of the proposed approach. In this section, we outline two shortcomings of our current implementation and suggest potential solutions for them. Finally, we suggest potential research directions on how the problem could be approached alternatively using Machine Learning.

A main bottleneck of the optimization approach presented in this work is the ability to scale to higher objective counts due to exponential overhead with higher objective counts. This could, for example, be done by leveraging the property that some objectives in our problem are independent of each other, or by applying more scalable optimization algorithms. This means that we could, for instance, explicitly keep a certain number of solutions per traffic flow rather than generating the costly non-dominant sorting for NSGA2.

The ”Polynomial Mutation” and ”Simulated Binary Crossover” operators used in this work are standard in the genetic algorithm domain. They are employed by the genetic algorithm to recombine the parent solutions and mutate the resulting offsprings to explore the parameter search space. Designing specialized operators that more explicitly exploit the properties and expert knowledge about how the offsets we applied influence traffic congestion on the network could be beneficial. Specifically, it could be useful to have a mutation operator that more quickly offsets multiple flows into several hot spots, which may benefit flow interference at switches.

In recent years, Reinforcement Learning (RL) has proven effective in many domains and optimization problems. An interesting research avenue could be to model the optimization problem presented in this work as an RL problem, potentially in combination with Graph Neural Networks (GNN) as introduced by Scarselli et al. in [13]. GNNs have proven effective in Ethernet-based real-time networks before, for instance, in predicting the schedulability as done by Mai and Navet in [14]. This approach could be leveraged further by applying Transfer Learning to create an optimization agent that can adjust to new use cases more quickly by exploiting previously attained networking knowledge.

References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 2. Institute of Electrical and Electronics Engineers (IEEE), pp. 182–197, Apr. 2002. doi: 10.1109/4235.996017.

[2] M. Märtens and D. Izzo, “The asynchronous island model and NSGA-II,” Proceedings of the 15th annual conference on Genetic and evolutionary computation. ACM, Jul. 06, 2013. doi: 10.1145/2463372.2463516.

[3] P. Keller and N. Navet, “Approximating WCRT through the aggregation of short simulations with different initial conditions: application to TSN,” Proceedings of the 30th International Conference on Real-Time Networks and Systems. ACM, Jun. 07, 2022. doi: 10.1145/3534879.3534886.

[4] S. Samii, S. Rafiliu, P. Eles, and Z. Peng, “A Simulation Methodology for Worst-Case Response Time Estimation of Distributed Real-Time Systems,” 2008 Design, Automation and Test in Europe. IEEE, Mar. 2008. doi: 10.1109/date.2008.4484735.

[5] P. Keller and N. Navet, ”Approximation of worst-case latencies in real-time Ethernet networks via aggregated short simulations: how to select the relevant parameters” under submission, 2023.

[6] J. Blank and K. Deb, “Pymoo: Multi-Objective Optimization in Python,” IEEE Access, vol. 8. Institute of Electrical and Electronics Engineers (IEEE), pp. 89497–89509, 2020. doi: 10.1109/access.2020.2990567.

[7] H. Bauer, J.-L. Scharbarg, and C. Fraboul, “Improving the Worst-Case Delay Analysis of an AFDX Network Using an Optimized Trajectory Approach,” IEEE Transactions on Industrial Informatics, vol. 6, no. 4. Institute of Electrical and Electronics Engineers (IEEE), pp. 521–533, Nov. 2010. doi: 10.1109/tii.2010.2055877.

[8] H. Bauer, J.-L. Scharbarg, and C. Fraboul, “Applying Trajectory approach with static priority queuing for improving the use of available AFDX resources,” Real-Time Systems, vol. 48, no. 1. Springer Science and Business Media LLC, pp. 101–133, Dec. 01, 2011. doi: 10.1007/s11241-011-9142-9.

[9] Y. Tian et al., “Evolutionary Large-Scale Multi-Objective Optimization: A Survey,” ACM Computing Surveys, vol. 54, no. 8. Association for Computing Machinery (ACM), pp. 1–34, Oct. 04, 2021. doi: 10.1145/3470971.

[10] M. Boyer, N. Navet, X. Olive, and E. Thierry, “The PEGASE Project: Precise and Scalable Temporal Analysis for Aerospace Communication Systems with Network Calculus,” Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 122–136, 2010. doi: 10.1007/978-3-642-16558-0_13.

[11] N. Navet, J. Seyler, J. Migge, ”Timing verification of realtime automotive Ethernet networks: what can we expect from simulation?”, Embedded Real-Time Software and Systems (ERTS 2016), Toulouse, France, January 27-29, 2016.

[12] M. Boyer, N. Navet and M. Fumey, ”Experimental assessment of timing verification techniques for AFDX”, Embedded Real-Time Software and Systems (ERTS 2012), Toulouse, France, February 1-3, 2012.

[13] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini, “The Graph Neural Network Model,” IEEE Transactions on Neural Networks, vol. 20, no. 1. Institute of Electrical and Electronics Engineers (IEEE), pp. 61–80, Jan. 2009. doi: 10.1109/tnn.2008.2005605.

[14] T. Long Mai and N. Navet, “Improvements to Deep-Learning-based Feasibility Prediction of Switched Ethernet Network Configurations,” 29th International Conference on Real-Time Networks and Systems. ACM, Apr. 07, 2021. doi: 10.1145/3453417.3453429.