Online Reconfiguration for TSN in Avionics Using Backup Paths and Integrated Mapping, Scheduling, and Analysis Tools

Daniel Bujosa
Mälardalen University, Västerås, Sweden

Mohammad Ashjaei
Mälardalen University, Västerås, Sweden

Saad Mubeen
Mälardalen University, Västerås, Sweden

Abstract

This work addresses the Euromicro Conference on Real-Time Systems (ECRTS) Industry Challenge on embedded Time-Sensitive Networks (TSN) reconfiguration for avionics systems. We propose a solution that combines previously developed configuration and analysis tools with a theoretical mechanism for stream rerouting and deployment. The proposed solution has been validated on the case study provided in the challenge, achieving minimal performance degradation for existing streams.

2012 ACM Subject Classification Computer systems organization → Embedded systems; Computer systems organization → Real-time systems

Keywords and phrases Time Sensitive Networking, Scheduled Traffic, Audio-Video Bridging, Online Reconfiguration

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.9

1 Introduction

Time-Sensitive Networking (TSN) is gaining momentum as a replacement for legacy avionics networks such as AFDX, offering enhanced flexibility and support for mixed-criticality traffic. However, its deployment in safety-critical environments, such as aircraft, raises new challenges, particularly in fault recovery through online reconfiguration. The Industry Challenge described in [2] focuses on embedded reconfiguration of TSN networks under port faults, with constraints on computation time, resource availability, and service continuity.

In this work, we present a hybrid approach to the challenge. We first propose a theoretical mechanism for stream rerouting and deployment, designed to minimize disruption and support fault containment. Then, we leverage previously published tools: HERMES [4] for recomputing and applying Scheduled Traffic (ST) schedules, and the method in [3] to verify that Audio-Video Bridging (AVB) traffic still meets its timing constraints post-reconfiguration. We validate our solution using the data and failure scenarios described in the challenge case study, demonstrating its effectiveness with minimal impact on unaffected flows.

This contribution aims to bridge the gap between theoretical feasibility and practical applicability in TSN reconfiguration within embedded avionics systems.

2 Proposed Solution

The proposed reconfiguration procedure comprises three sequential stages. Under the Industry Challenge assumption that the Centralized Network Controller (CNC) immediately detects any permanent link failure, the first stage reroutes all streams originally traversing the failed link onto alternative paths. In the second stage, a new end-to-end configuration is computed for every path now carrying relocated streams. Finally, in the third stage, this updated configuration is deployed across the network. Although both the rerouting and deployment phases are described here as theoretical proposals, we have implemented a simplified rerouting mechanism to evaluate our reconfiguration framework, and we also provide our own implementation of the CNC [1]. Each of these steps is described in detail below.

2.1 Rerouting

To enable instantaneous failover, we precompute, for each stream, one or more disjoint backup paths that can be activated immediately upon detection of a primary-path failure. Under the assumption that the interval between successive failures exceeds the time required to compute a new backup path, this approach adds zero extra latency to the reconfiguration process.

The exact number of backup paths per stream depends on the exposure of the primary route to non-critical link failures: backups need only protect against failures on links that are not inherent single points of failure (for example, the end-station–switch links in the case study). Consider the TSN topology in Figure 1, where ES1 and ES2 are connected through four switches (SW1–SW4) and seven links.

- **Single backup suffices.** If the primary path is

  `ES1-SW1-SW2-SW3-ES2`,

  then a single backup path

  `ES1-SW1-SW4-SW3-ES2`

  protects against any failure on `SW1-SW2`, `SW2-SW3`, or `SW3-ES2`.
- **Two backups required.** If the primary path is

  `ES1-SW1-SW2-SW4-SW3-ES2`,

  then two backups are needed:

  `ES1-SW1-SW4-SW2-SW3-ES2`

  (covers failures of `SW1-SW2` and `SW4-SW3`), and

  `ES1-SW1-SW2-SW3-ES2`

  (covers failures of `SW2-SW4`).

By tailoring the number and disjointness of backup routes to the primary path's vulnerability, we guarantee immediate, zero-delay rerouting.

More sophisticated but complex solutions have also been proposed. For example, in the context of switched Ethernet networks, Pozo et al. [6] introduce a mechanism for computing backup routes between network switches instead of establishing them for each stream. By activating alternative routes only in the vicinity of a link failure, their scheme confines reconfiguration to the affected region and leaves all other links as well as the streams traversing those links completely unaltered.

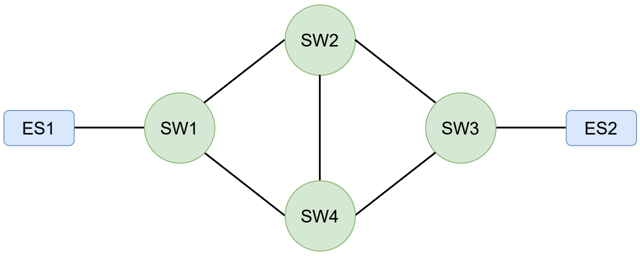


Figure 1 TSN topology for ES1 and ES2 through switches SW1–SW4.

2.2 Configuration

Regarding network configuration and reconfiguration, we employ the HERMES heuristic scheduler for ST [4] together with the AVB Worst-Case Response-Time Analysis (WCRTA) method presented in [3]. To ensure that ST transmission windows are correctly sized and that AVB WCRTs are computed accurately, we base all calculations solely on the maximum frame size of the case-study streams.

First of all, we revised the traffic-to-priority mapping proposed in the Case Study. As noted by Boyer et al. [2], the main goal of priority assignment in real-time networks is to maximize overall schedulability. However, the original mapping exhibited two critical issues:

1. *Deadline-constrained flows in the Best-Effort queue.* Several streams with firm end-to-end deadlines were assigned to Best-Effort (BE) priorities, complicating both schedulability and schedulability analysis.
2. *Unschedulable short-deadline AVB traffic.* A subset of AVB flows with deadlines below 400 000 ns failed the WCRTA, even under the optimistic assumption that all the streams with higher deadlines were confined to the BE queue.

For this reason, we reconfigured the case study using LETRA [5], a mapping tool designed for automated traffic class assignment in TSN. LETRA mapped traffic class TC7 to the ST class and TC0–TC1 to the BE class. Since the case study does not specify the scheduling policy for TC2–TC6, we treat all their streams as Time-Triggered (TT) and therefore eligible for either ST or AVB classes. In a previous pre-analysis we evaluated each TC2–TC6 stream and found that most streams with deadlines under 400 000 ns failed to meet their constraints, therefore, these streams were assigned to the ST class, while the remaining TC2–TC6 streams were assigned to the AVB class.

The traffic classified as ST was scheduled using HERMES, which required three separate queues corresponding to the three highest TSN priorities (priorities 0, 1, and 2). The scheduling process completed in **5453 μs**, and all streams met their respective deadline and jitter constraints by design. Figure 2 illustrates the schedule of ST across all network links. Each colored box represents a frame belonging to a specific stream, with different colors indicating different streams. The portions of time not allocated to ST frame transmissions remain available for other traffic classes, such as AVB and BE.

The procedure for mapping and configuring AVB traffic is as follows. For each link, the available bandwidth remaining after allocating BE traffic is distributed among the AVB queues. The allocation to each AVB class is based on the weighted average of its utilization on that link.

Specifically, the percentage of bandwidth reserved for AVB class X on link l (RBW_l^X),

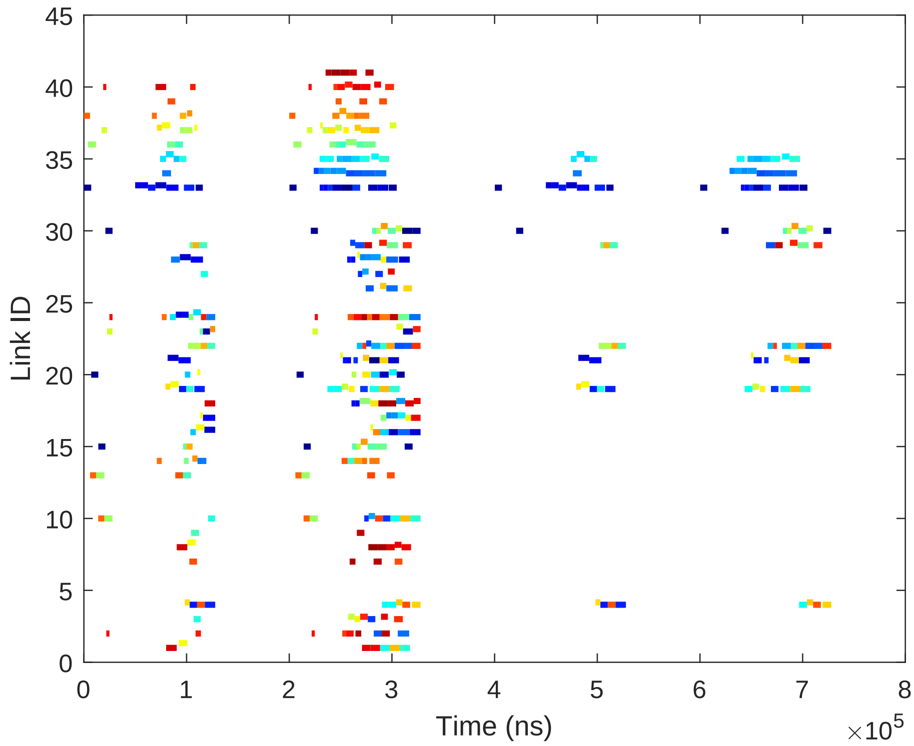


Figure 2 ECRTS Industry Challenge ST schedule.

defined by the *idleSlope* parameter of the standard, is given by:

$$RBW_l^X = \left(1 - U_l^{BE}\right) \times \left(\frac{U_l^X}{U_l^{AVB}}\right) \tag{1}$$

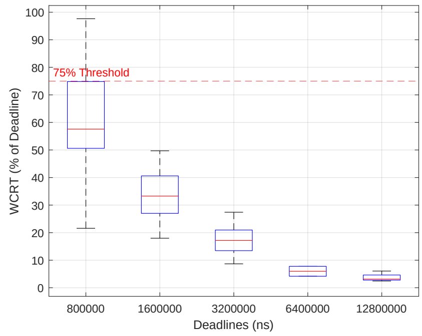
where:

- U_l^{BE} is the utilization of BE traffic on link l,
- U_l^X is the utilization of AVB class X on link l, and
- U_l^{AVB} is the total utilization of all AVB traffic on link l.

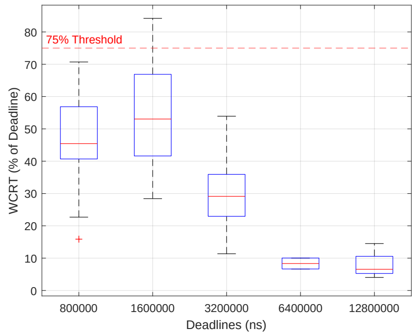
We now proceed to determine the priority mapping of the AVB streams through several analysis iterations, progressively refining the assignment. To do so, we first assign all streams classified as AVB, that is, all streams in TC2–TC6 with deadlines greater than 400 000 ns, to the highest AVB priority, and the traffic is then analyzed using the WCRTA.

For all the AVB streams, if there are two streams that share a link along their path and have the same priority, and one of them exhibits a WCRT greater than the threshold (in our case, 75% of the deadline), while the other has a WCRT below the threshold, then the second stream is moved to a lower priority, and a new traffic mapping iteration is analyzed. This process is repeated until there are no such pairs of streams sharing a link and priority, where one exceeds the threshold and the other does not.

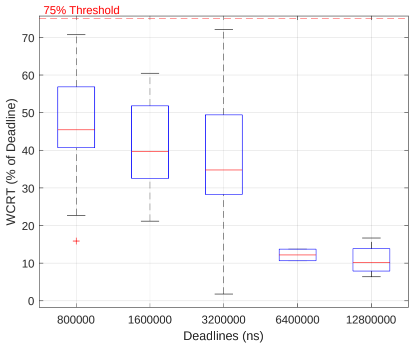
However, in this work we use a simplified version of this method. We group all streams with the same deadline. If one stream in the group exceeds the threshold and another stream from a different deadline group, sharing the same priority, has a WCRT below the threshold, then the entire second group is moved to a lower priority.



(a) 1^{st} iteration



(b) 2^{nd} iteration



(c) 3^{rd} iteration

Figure 3 WCRT progression across iterations for determining the priority of AVB streams

In this case, the process required three iterations. In the first iteration, all AVB streams had priority 3 (the highest available for AVB, since ST occupies priorities 0-2). In the second iteration, streams with deadlines greater than or equal to 1 600 000 ns were moved to priority 4. Finally, in the third iteration, streams with deadlines greater than or equal to 3 200 000 ns were moved to priority 5.

Figures 3a, 3b, and 3c show the progression of the WCRT, expressed as a percentage of the deadline, for iterations 1, 2, and 3, respectively. The graphs display box plots of WCRT values expressed as a percentage of each stream's deadline, grouped by deadline. The dashed horizontal line represents the threshold used.

Finally, all BE streams are assigned to priority 6, resulting in a schedulable configuration. This setup ensures a 25% margin for AVB traffic and leaves one queue/priority unused, which will be leveraged during the deployment phase. The entire configuration process, typically performed offline, required a total of **1579.037 ms**, including both ST scheduling and AVB analysis/configuration.

Table 1 provides a detailed breakdown of the scheduling and analysis times.

Table 1 Breakdown of scheduling and analysis times

HERMES	**WCRTA 1^{st} Iteration**	**WCRTA 2^{nd} Iteration**	**WCRTA 3^{rd} Iteration**	**Total**
5.453 ms	486.279 ms	565.788 ms	521.517 ms	1579.037 ms

2.3 Reconfiguration

Once the link failure has been detected and the affected streams have been redirected to their backup paths, the reconfiguration process begins. In this preliminary version, the reconfiguration consists of rescheduling all ST traffic using HERMES, reallocating the required bandwidth according to Eq. (1), and analyzing the AVB traffic to ensure that it continues to meet its timing requirements.

To evaluate the effectiveness of this reconfiguration process, we simulate the failure of each link that does not represent a single point of failure, specifically the eight inter-switch links. As a simplified implementation of the rerouting strategy described in Section 2.1, for each potential link failure we assume that the backup path for the affected streams is the shortest unaffected path between the stream's source and destination.

The evaluation results indicate that 6 out of the 8 link failures resulted in schedulable configurations. The average rescheduling time and AVB analysis time for these cases were **2.507 ms** and **502.541 ms**, respectively. For the two unschedulable failures, the issue arose from HERMES not supporting circular dependencies, which were introduced by the simplified rerouting mechanism.

In conclusion, the 25% margin between the WCRT and the deadline provided by the original mapping ensures the schedulability of AVB traffic in the presence of at least one link failure. However, the circular dependency constraint imposed by HERMES must be considered when defining routing and backup paths, particularly for ST.

2.4 Deployment

Although the ST traffic scheduling time is relatively low and the analysis time has considerable room for improvement (potentially being reduced by up to an order of magnitude), these reconfiguration times are still relatively high compared to the periods handled in the case study and in similar networks. Communication interruptions on the order of milliseconds could be critical.

This is why the deployment mechanism is fundamental to guarantee the continuous exchange of information, even if some performance degradation occurs. Additionally, there are deployment challenges, such as the one identified in [2], that must be considered during deployment.

Once the link failure is detected and the affected streams are redirected to their backup paths, the deployment process begins alongside the reconfiguration process. After recalculating the AVB bandwidth reservations according to Eq. (1), a step that requires negligible computation time, the transmission of AVB traffic and the repositioned BE traffic can commence, as no additional configuration is required.

Although the WCRTA must still be completed to ensure compliance with timing requirements, the margin between the WCRT and the deadline suggests that traffic will likely remain schedulable or, at worst, operate with minimal performance degradation.

In contrast, deploying ST traffic is more complex. The schedule cannot be modified mid-hyperperiod without causing issues, such as ST frames occupying incorrect transmission windows, among other inconsistencies. Therefore, the new schedule must be deployed only at the beginning of a new GCL hyperperiod, ensuring all switches are synchronized at the start of their GCLs.

However, in the worst-case scenario, this would result in a disruption in the transmission of ST streams equal to $2.507 + 0.8 = 3.307$ ms, where 2.507 ms is the ST rescheduling time and 0.8 ms the duration of the GCL hyperperiod. In the case of the Industry Challenge scenario, such a delay is unacceptable.

To address this, we leverage the queue intentionally left unused during the configuration process described in Section 2.2. This queue is reassigned a priority level higher than non-ST traffic but lower than ST traffic. Once the failure is detected and ST traffic is rerouted, it can immediately begin transmission using this intermediate queue.

To the best of our knowledge, there are no existing WCRTA that account for this type of TSN configuration. However, its characteristics suggest high performance. Since this temporary queue is shaped solely by the TAS and strict priority arbitration, it can only be interrupted by ST traffic, same-priority flows (due to FIFO ordering), and at most one lower-priority frame. Consequently, although this temporal configuration may introduce higher jitter and latency until the new schedule is fully deployed, the overall performance degradation is expected to be minimal.

Finally, after a short transition period corresponding to the reconfiguration time, during which affected traffic may experience slight performance degradation, the new ST schedule will be applied. This guarantees ST traffic schedulability by construction, while WCRTA analysis will validate the schedulability of AVB traffic. In case AVB traffic becomes unschedulable, alternative priority-mapping mechanisms, such as the one presented in Section 2.2, should be applied. In such cases, it may also be necessary to evaluate the utility variable to selectively degrade lower-utility streams in order to preserve the schedulability of higher-priority ones.

Figure 4 shows a diagram of the reconfiguration process. It presents four timelines: the reconfiguration procedures executed by the CNC and the transmission phases of each traffic class as described previously.

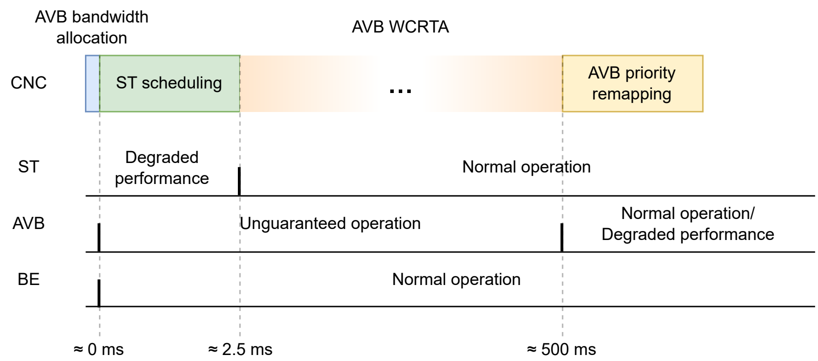


Figure 4 Diagram of the reconfiguration and deployment process.

3 Implementation and Effort Evaluation

Both the tools and the experimental results are available at `https://github.com/DanielBujosa/ECRTS_Industry_Challenge.git`. The repository contains two main folders:

- **Configuration**: This folder includes the tools used for the initial network configuration and the results of each analysis and iteration required. It contains:
  - Seven subfolders:
    - `Experiment0.X` (4 folders): contain the analysis of the configuration proposed in the Industry Challenge.
    - `Experiment1.X` (3 folders): contain the three iterations required to map AVB traffic priorities.

  - TSN_Streams.txt: defines the streams of the case study.
  - ECRTSICreader.m: a MATLAB script that reads TSN_Streams.txt, classifies the traffic, and creates the input file for the ST scheduler.
  - TSN_HeuristicScheduler.exe: executable for the ST scheduler.
  - AVB_analysis_input_generator.m: MATLAB script that takes the ST schedule and the AVB/BE traffic and generates the input file for WCRTA.
  - AVB_analysis.exe: WCRTA analysis executable.
  - test0.bat: a Windows batch script that executes the complete configuration workflow.
- **Reconfiguration**: This folder contains the tools and results related to the evaluation of network reconfiguration. It includes:
  - Eight subfolders:
    - Experiment1 to Experiment8: each folder corresponds to a different link failure scenario.
  - TSN_Streams.txt: defines the case study streams.
  - ECRTSICreaderLoop.m: a MATLAB script that reads TSN_Streams.txt, creates the eight link failure scenarios with their backup paths, classifies the traffic into TSN classes, and generates the corresponding ST scheduler input file for each scenario.
  - TSN_HeuristicScheduler.exe, AVB_analysis_input_generator.m, and AVB_analysis.exe: the same tools used for ST scheduling and AVB analysis.
  - test.bat: a Windows batch script that executes all reconfiguration tests.

Regarding the evaluation effort, understanding the problem was not particularly difficult, thanks to the clarity of the white paper. However, a few inconsistencies in the file TSN_Streams.txt did pose some challenges. For instance, the stream STR_ES14_ES7_B has source = ES15, but its defined path is ES14 SW5 SW4 SW3 ES7, which is inconsistent. Similarly, the stream STR_ES6_ES14_B includes the path ES6 SW3 SW5 ES14, which is invalid because there is no direct link between SW3 and SW5.

On the other hand, most of the tools used in the solution had already been developed prior to this challenge. Although their development took several years, this effort is not considered part of the time dedicated to solving the Industry Challenge.

Specifically, we devoted approximately 2 to 3 weeks of full-time work to the challenge itself.

Acknowledgements

This work is supported by the Swedish Knowledge Foundation (KKS) via the SEINE and MARC projects and by the Swedish Governmental Agency for Innovation Systems (VINNOVA) via the INTERCONNECT, FLEXATION and iSecure projects.

References

1 Inés Álvarez, Andreu Servera, Julián Proenza, Mohammad Ashjaei, and Saad Mubeen. Implementing a first CNC for scheduling and configuring TSN networks. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4. IEEE, 2022.

2 Marc Boyer and Rafik Henia. Industrial challenge: Embedded reconfiguration of TSN. `https://hal.science/hal-04630862`, 2024. HAL Id: hal-04630862.

3 Daniel Bujosa, Julian Proenza, Alessandro V Papadopoulos, Thomas Nolte, and Mohammad Ashjaei. An improved worst-case response time analysis for avb traffic in time-sensitive networks. In *2024 IEEE Real-Time Systems Symposium (RTSS)*, pages 135–147. IEEE, 2024.

4 Bujosa, Daniel and Ashjaei, Mohammad and Papadopoulos, Alessandro V and Nolte, Thomas and Proenza, Julián. HERMES: Heuristic multi-queue scheduler for TSN time-triggered traffic with zero reception jitter capabilities. In *Proceedings of the 30th International Conference on Real-Time Networks and Systems*, pages 70–80, 2022.

5 Mateu, Daniel Bujosa and Ashjaei, Mohammad and Papadopoulos, Alessandro V and Proenza, Julian and Nolte, Thomas. LETRA: Mapping legacy ethernet-based traffic into TSN traffic classes. In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8. IEEE, 2021.

6 Francisco Pozo, Guillermo Rodriguez-Navas, and Hans Hansson. Self-healing protocol: Repairing schedules online after link failures in time-triggered networks. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 129–140. IEEE, 2021.