Multi-Partner Project: Resilient Time-Sensitive Networks (ResTSN)

Marc Boyer
DTIS, ONERA, Université de Toulouse
3100 Toulouse, France
marc.boyer@onera.fr

Rafik Henia
cortAIx/Labs, THALES
91767 Palaiseau, France
rafik.henia@thalesgroup.com

***Abstract*—Faults may appears in embedded networks: vibrations, temperature conditions, cybersecurity attacks, etc. may cause a port to stop sending frames, requiring a network reconfiguration. In real-time networks, the new configuration must continue ensuring real-time guarantees. In the French Resilient Time-Sensitive Networks (ResTSN) project, ONERA and THALES are developing a solution to enable the reconfiguration of embedded Time-Sensitive Networks. This paper presents the core ideas driving the development of the architecture and associated algorithms of TSN reconfiguration. The approach will ultimately be applied on a Multi-Role Armoured Vehicle (MRAV) use-case. This paper also presents our cybersecurity risk assessment approach to the use of TSN on this MRAV use-case.**

***Index Terms*—Real-Time Ethernet, TSN, reconfiguration, embedded networks, cybersecurity risk assessment**

I. Introduction

A. Context: embedded real-time networks

Today, real-time communication networks in vehicles are numerous, each designed to a certain shape of traffic. For example, in airliners, the Avionics Full-Duplex Switched Ethernet (AFDX) network transmits critical flight control data, whilst the Aeronautical Radio Incorporated (ARINC) 429 network connects sensors and actuators, and the Ethernet network transmits non-critical video data requiring high bandwidth. This fragmentation creates big challenges in vehicles in terms of cables weight and volume, energy consumption, maintenance costs, and architecture complexity. The rise of new applications increasing the communication demand, like autonomous driving in cars, or predictive maintenance in aircraft, further complicates the network architecture. Merging some of these networks into a single real-time communication network would provide significant benefits.

Time-Sensitive Networking (TSN), an extension of Ethernet technology standardized by IEEE, is a promising technology to unify these networks. One notable advantage of TSN lies in its capability of supporting traffic data of different shapes (e.g. synchronous control-command vs. asynchronous video) and various levels of urgency (e.g. critical flight control vs. best-effort passenger entertainment) within the same network infrastructure.

Transmitting critical and non-critical data over the same physical TSN infrastructure requires maintaining reliable communications. However, in the event of a network failure (due to a hardware breakdown, a software malfunction or a cyberattack), some communications will inevitably be lost, potentially compromising the system's overall availability and/or integrity.

This project has been partially funded by the French National Research Agency (ANR) as project ANR-22-ASTR-0017-01.

B. Project aim and challenge: ensuring a resilient TSN through reconfiguration

The ResTSN project's objective is to maintain uninterrupted network service and uphold the stringent safety and reliability standards of operations, even in the face of unforeseen failures, by allowing the network to automatically and dynamically reconfigure itself in case of failure, ensuring that its most critical functions continue to operate, even with reduced resources.

The project's challenge is threefold: (i) the TSN must be reconfigured to preserve critical streams; (ii) the new network configuration must minimize the impact, both on the flows that are not affected by the failure, to ensure continuity of service, and on the number of new vulnerabilities that could be exploited by an attacker; (iii) the reconfiguration must be calculated and implemented rapidly, ideally within a few seconds, using the onboard computing resources, whereas the initial configuration may have required days of computation on dedicated machines during the design phase.

C. Project's reduced scope and self-imposed constraints

TSN offers several real-time and reliability oriented mechanisms. The scope of the project has been restricted to two main real-time solutions: the *Time Aware Shaper* (TAS) [1] to forward *scheduled traffic* (ST), i.e. traffic where the transmission time of each frame respects a predefined schedule, a.k.a. Time-Triggered (TT) scheduling [2], and the *Credit-Based Shaper* (CBS) [3] to forward asynchronous traffic. TAS is intended to forward traffic with very hard constraints on latency and jitter (sub-millisecond), whereas CBS is intended to forward real-time traffic with no constraints on jitter and latencies constraints in milliseconds. The Frame Replication and Elimination for Reliability (FRER) mechanisms [4], allowing to transparently use several paths, using replication of frames and elimination of duplicated frames, are also considered in the project.

A TSN switch supports 8 priority levels (from #0 to #7), each one having its own queue. In ResTSN, the highest priority level (#7) is devoted to TAS flows, intermediate priority levels are devoted to CBS, and the lower priority levels are used for best effort traffic. A *utility* integer parameter is added to each flow to

978-3-9826741-0-0/DATE25/© 2025 EDAA

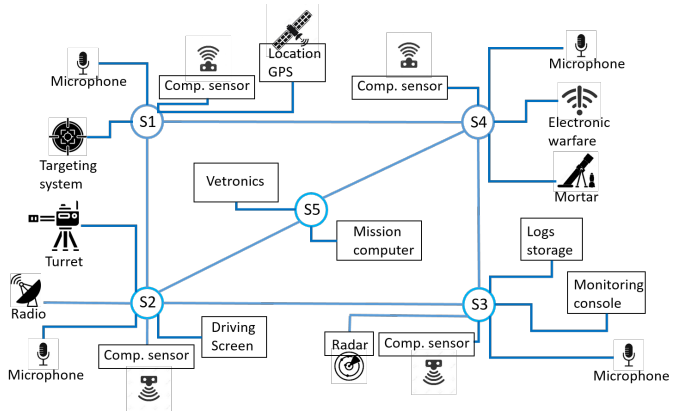

Fig. 1. MRAV network topology

distinguish between flows with the same priority. When one or several ports are considered faulty, they are *excluded* from the network, i.e., the network stops using theses ports to forward data flows.

When some ports are out of order, the reconfiguration problem in a non real-time network consists only in finding new routes. In a real-time network, the difficulty is also to allocate resources to continue guaranteeing real-time performances. In degraded mode, our industrial assumption in the ResTSN project is that it is better to continue guaranteeing real-time performances for high-priority flows, possibly at the expense of low-priority flows, rather than offer best-effort performances to all flows.

II. Industrial use-case: a Multi-Role Armoured Vehicle

For the project, we selected a Multi-Role Armoured Vehicle (MRAV) use-case. An MRAV is a troop transport and combat vehicle whose missions range from reconnaissance to combat support. Due to this multi-mission aspect, its equipment and underlying capacities may be activated or deactivated on short notice. Inherently to the battle hazards, the equipment, including the underlying network infrastructure, may be damaged or destroyed, hence calling for real-time reconfiguration of the TSN flows. The vetronics of an MRAV is the embedded electronics of the vehicle, i.e. the architecture of its navigation, control, communication, observation, and protection systems. We propose here (cf. Fig. 1) a network topology with 5 spatially distributed switches arranged to balance the various connections to the MRAV equipment. The connection topology allows for multiple possible paths for data flows and also provides robustness if the network is partially damaged. Each piece of equipment/sensor is connected to a peripheral switch (S1 to S4). A central switch (S5) connects the mission computer, which receives a large portion of the data streams.

It is impossible to list and describe herein all the equipment connected to the MRAV vetronics. Let us just name a few to provide an insight on the complexity and criticality of the Computerized Command, Control, Communications and Intelligence (C4I) system: (i) a vision system that combines three functions: close hemispheric surveillance, automatic detection of moving objects, and laser alert detection; (ii) software-defined radios, enabling the transfer of voice and data; (iii) a system designed for protection against improvised explosive devices; (iv) multiple mechanical sensors, placed on the main components of the vehicle, such as the suspension, brake pads, and gearbox, enabling predictive maintenance; (v) sensor calculators synthesizing the information coming from the sensors and transmitting them to the mission computer; (vi) mission computer that receives information from various equipment and provides coherent information to the operators, including decision support; (vii) supervision console, with screens to allow the operator to supervise the systems and receive video feedback; (viii) a GPS system, for locating and positioning the vehicle; (ix) microphones, located on the roof, capable of detecting gunfire and thus determining the shooter's position through triangulation with data collected from other connected vehicles, thus allowing for turret automatic rotation and fire; (x) a semi-automatic mortar; (xi) a radar operating at 360° and up to 25 km, capable of detecting, identifying, and tracking ground, maritime, or aerial vehicles, as well as infantry; (xii) a remote-controlled turret, armed with a 7.62 mm caliber on the roof; etc.

This use-case will be used to experiment our reconfiguration algorithms and as support for our cybersecurity risk assessment. Note that an alternative open use-case has been provided to the community as an ECRTS industrial challenge. The presentation of that use-case, can be found in [5] and the corresponding data set can be found in the associated repository [6].

III. State of the art

A. TSN configuration and reconfiguration

Resilience mechanisms already exist in traditional data networks. In case a failure occurs along a path, new routes can be computed to forward data streams. But in the context of real-time network, computing a path is insufficient: one also has to allocate some resources to ensure timing requirements. With TSN, resource allocation (a.k.a. the *configuration problem*) is highly dependant on the selected kind of mechanism. For TAS, the configuration consists in defining time windows dedicated to each frame all along the path, for each flow, such that the real-time requirements of all flows are guaranteed. The TSN implementation uses a queue *gate*, which is either closed or open, based on a per-port cyclic configuration table, the *Gate Control List* (GCL). For CBS flows, the configuration consists in defining a virtual queue bandwidth, a.k.a. queue *idle slope*.

Both configuration problems are hard. The TAS configuration is a variation of the bin packing problems, and hundreds of papers have proposed some solutions [7] but no clear conclusion can be derived [8], [9]. The problem is considered difficult. When the part of the bandwidth used by TAS is low, several solutions can be used. An open implementation of 17 methods exists: `tsnkit` [8]. The CBS configuration is also an open problem. Increasing the slope of a queue will decrease its own delay but will also increase the delay of lower priority flows in the same port, increase the flow burstiness at

port output, and so increase the delay of downstream queues. There is currently no solution to compute a global optimum. However, some solutions based on per queue budget exist [10], [11]. The reconfiguration problem is even harder than the configuration problem, especially when aiming to limit side-effects on flows not impacted by the event that initially triggered the reconfiguration.

The main project orientations selected to solve these issues are presented in Section IV.

B. TSN cybersecurity

Their are several TSN domain-agnostic threat taxonomies. Ergenç et al. [12] proposes a taxonomy based on an exhaustive approach of STRIDE. Synchronization attacks have been implemented multiple times [13] [14]. Meyer et al. [15] have evaluated the impact of Denial-of-Service (DoS) attacks on a TSN automotive use case, leveraging the Per-Stream Filtering and Policing (PSFP) mechanisms to protect against such attacks. [16] provides a threat analysis and remediation considerations on deterministic networks, including TSN, whilst [17] has a similar aim, but focuses on attacks on synchronization protocols.

However, these papers do not provide a complete risk assessment, with TSN-related risks assessed in terms of likelihood of occurrence and severity of consequences. This is because they do not address the business / operational assets and their security needs. Without these, it is impossible to derive the severity levels of the business / operational impacts if the security needs are violated. Likewise, the aforementioned papers do not precisely list the supporting assets whose vulnerabilities can be exploited by an attacker, or which can fail accidentally. A methodology to perform such as risk assessment is introduced in Section V.

IV. Main project orientations for TSN reconfiguration

A. Architecture considerations

In TSN, two main entities are in charge of the network configuration: the *Centralised User Configuration* (CUC) element, in charge of collecting all data flows requirements (source, destinations and traffic contract), and the *Centralised Network Configuration* (CNC) element, in charge of configuring the network elements. We chose to put both elements on the same physical system to avoid communication problems. Since a faulty port may not declare itself as faulty, this element is in charge of collecting monitoring information and deducing which ports to exclude from the network. This grouping is also used in [18], under the name *Configuration Agent* (CA), and in [19], under the name *Network Management Unit* (NMU). Depending on system resources, a backup CUC/CNC can be set on another host.

FRER is used between the host of the CUC/CNC and all network elements to maximise fault tolerance. This has little impact on the system performance because the configuration traffic is very small by contrast with the global network use.

B. Approach for TAS flows management

To find a new route and allocate time windows for flows whose path is broken, we plan either to reuse an existing TAS configuration algorithm (some are building a configuration by adding flow by flow iteratively) or to develop a new one to find time windows for flows whose path is broken. But it may be impossible to allocate resources to all TAS flows (Here, impossible may mean that the problem has no solution, but also that the embedded algorithm was not able to find a solution in admissible time.). TAS flows without a route are allocated by increasing utility, and lower utility flows may be removed to free resources. After these steps, some TAS flows may have no time window, either because they were initially on a path with a faulty port or link, or because they have been removed to free resources. The queue #6 is reserved to host all these remaining TAS flows. This queue must not be used in the initial TSN configuration. No shaping mechanisms is used for this queue. Since queue #6 has the highest priority after the TAS queue #7, these flows will use all the unused bandwidth.

C. Approach for CBS flows management

The handling of the CBS flow will be done by priority level, once that the TAS flows have been handled. We will use a CBS configuration based on delay budget: each CBS queue q_i has a delay budget D_i, and the slope of this queue is computed to provide this delay. New flows will be routed using a shortest path algorithm with some per queue load weighting, to avoid to putting all flows on the same path. If the sum of the delay budgets along the path is insufficient to ensure the latency requirement, the delay budgets are reduced in an homogenous way all along the path. The slope of all queues where some flow have been added is recomputed to ensure the delay budget. This slope configuration has to take into account the impact of the higher priority queues: the TAS queue and the degraded TAS queue. The algorithms in [10], [11] only consider TSNs where a port has only CBS queues, and have to be adapted to take into account the TAS queues with higher priority.

Like for the TAS case, some queues may be too loaded to be able to ensure the expected latency. We plan to test a few alternative routes, and if none is found, to degrade the priority of the flows. This degradation can be done either locally (the PSFP mechanisms allows to locally modify the priority of a flow) or globally, all along the path.

D. Approach for reconfiguration flow management

The reconfiguration itself will require some management messages, to detect the faulty ports, and deploy the new configuration. In order to shorten the reconfiguration time, we plan to give a high priority to theses flows. Since they are not used in the nominal case, these flows have a traffic impact only during the reconfiguration phase. Either a dedicated queue is used (queue #5), or the queue #6 is used, mixing configuration deployment and degraded TAS flows. If a dedicated queue is used, one may add a CBS shaping mechanisms to avoid long blocking of the lower priority flows during the deployment of a new configuration.

V. Risk assessment methodology

Performing a cybersecurity risk assessment of a Time-Sensitive Network equipping an MRAV is a challenging task. This section presents a high-level view of our approach to deal with this risk assessment.

The level of a risk is generally assessed as the combination of the consequences of an event (i.e., the *severity* of the impacts) and the associated *likelihood* of occurrence.

To assess the severity of the impacts of a risk, we first need to identify the key business / operational assets. For our MRAV use-case, the transmission of data to support the driving of the vehicle, and the transmission of data and commands to and from weapon systems are two typical examples of operational assets. We then need to assess the cybersecurity needs of those assets. In the case of our MRAV, we can state that both assets need to be available. The need can be precisely specified in terms of traffic shape (i.e., message size, period or burst, offset...) and requirements (i.e., tolerable latency and jitter). Next, we can study what happens when those requirements are not met. For our MRAV use-case, a loss of availability of transmissions to/from the weapon systems may disallow fire, possibly leading to the loss of the engaged vehicle and its occupants. These consequences are assessed as critical in terms of severity.

To assess the likelihood of such an event occurring, we need to map the abstract operational asset to physical supporting assets. In the case of our MRAV, this mapping involves multiple steps. For example, we can first state that the availability of the transmission to/from the weapon systems depends, at least partially, on the availability and integrity of the time-synchronisation protocol of TSN, which in turn depends on the generalized Precision Time Protocol (gPTP). This supporting asset mapping is performed until the description level reaches the level at which a cybersecurity attack can be performed, e.g., the *systemIdentity* attribute stored in the Management Information Base (MIB). The attack scenario likelihood is assessed considering close to real life conditions. In our MRAV case study, we will for example check how likely it is that the attacker has remote access and control over at least one peripheral equipment.

This global approach is necessary to realistically assess risk levels. Indeed, some attacks may be technically easy to prepare in laboratory conditions, when one has a physical access to a switch, but accessing a switch of an MRAV located in a miliary Forward Operating Base (FOB) will dramatically reduce the risk likelihood.

VI. Current status and way forward

The project started in January 2023. During the first two years, we essentially gathered the requirements and constraints, and established a state of the art of TSN configuration and reconfiguration. Recently, the main architecture principles of the system have been defined, as shortly presented in this paper. We also plan to develop a TAS configuration algorithm devoted to ease TAS reconfiguration. In parallel, the cybersecurity risk assessment of the TSN on the MRAV is ongoing.

Acknowledgement

The authors wish to thank Stéphane Paul and Olivier Gilles of cortAIx/Labs for their contributions to the use-case description and risk assessment methodology in this paper.

References

[1] "IEEE standard for local and metropolitan area networks–bridges and bridged networks–amendment 25: Enhancements for scheduled traffic," IEEE, IEEE Standard 802.1Qbv, 2015.

[2] H. Kopetz, "Event-triggered versus time-triggered real-time systems," in *Operating Systems of the 90s and Beyond*, A. Karshmer and J. Nehmer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 86–101.

[3] "Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams," IEEE, Tech. Rep. IEEE 802.1Qav, 2010.

[4] "IEEE standard for local and metropolitan area networks – frame replication and elimination for reliability," IEEE, Tech. Rep. 802.1CB, September 2017.

[5] M. Boyer and R. Henia, "Industrial challenge: Embedded reconfiguration of TSN," Jul. 2024, working paper or preprint. [Online]. Available: https://hal.science/hal-04630862

[6] "Industrial challenge at ecrts: repository," https://github.com/ecrtsorg/.

[7] T. Stüber, L. Osswald, S. Lindner, and M. Menth, "A survey of scheduling algorithms for the time-aware shaper in time-sensitive networking (tsn)," *IEEE Access*, vol. 11, pp. 61 192–61 233, 2023.

[8] C. Xue, T. Zhang, Y. Zhou, M. Nixon, A. Loveless, and S. Han, "Real-time scheduling for 802.1Qbv time-sensitive networking (TSN): A systematic review and experimental study," in *IEEE 30th Real-Time and Embedded Technology and Applications Symp. (RTAS)*, 2024.

[9] T. Stüber, M. Eppler, L. Osswald, and M. Menth, "Performance comparison of offline scheduling algorithms for the time-aware shaper (TAS)," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 7, 2024.

[10] L. Maile, K.-S. J. Hielscher, and R. German, "Delay-guaranteeing admission control for time-sensitive networking using the credit-based shaper," *IEEE Open Journal of the Communications Society*, vol. 3, 2022.

[11] L. Zhao, Y. Yan, and X. Zhou, "Minimum bandwidth reservation for CBS in TSN with real-time QoS guarantees," *IEEE Transactions on Industrial Informatics*, 2023.

[12] D. Ergenç, C. Brülhart, J. Neumann, L. Krüger, and M. Fischer, "On the security of ieee 802.1 time-sensitive networking," in *2021 IEEE Int. Conf. on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.

[13] A. Finkenzeller, T. Wakim, M. Hamad, and S. Steinhorst, "Feasible time delay attacks against the precision time protocol," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 3375–3380.

[14] M. Fotouhi, A. Buscemi, A. Boualouache, F. Jomrich, C. Koebel, and T. Engel, "Assessing the impact of attacks on an automotive ethernet time synchronization testbed," in *IEEE Vehicular Networking Conf. (VNC)*, 2023.

[15] P. Meyer, T. Häckel, F. Korf, and T. C. Schmidt, "Dos protection through credit based metering – simulation-based evaluation for time-sensitive networking in cars," 2019. [Online]. Available: https://arxiv.org/abs/1908.09646

[16] E. Grossman, T. Mizrahi, and A. J. Hacker, "Deterministic Networking (DetNet) Security Considerations," RFC 9055, Jun. 2021.

[17] T. Mizrahi, "Security Requirements of Time Protocols in Packet Switched Networks," RFC 7384, Oct. 2014.

[18] P. Pop, M. L. Raagaard, M. Gutierrez, and W. Steiner, "Enabling fog computing for industrial automation through time-sensitive networking (TSN)," *IEEE Communications Standards Magazine*, vol. 2, no. 2, 2018.

[19] A. Kostrzewa and R. Ernst, "Achieving safety and performance with reconfiguration protocol for Ethernet TSN in automotive systems," *Journal of Systems Architecture*, vol. 118, p. 102208, 2021.