Sécurité des réseaux TSN : Vulnérabilités et Exploitations des Mauvaises Configurations

Gabriel Bourgeois*, Mohamed-Aymen Chalouf*, Guillaume Doyen†, et David Espes‡
*SOTERN - IRISA (UMR CNRS 6074), Université de Rennes,
†SOTERN - IRISA (UMR CNRS 6074), IMT Atlantique,
‡IRIS - Lab-STICC (UMR CNRS 6285), Université de Bretagne Occidentale,
*†{prenom.nom}@irisa.fr, ‡david.espes@univ-brest.fr

***Abstract*—Le terme *Time-Sensitive Networking* (TSN) désigne un ensemble de standards développés par le groupe de travail IEEE 802.1 qui vise à rendre Ethernet déterministe. TSN s'est révélé comme un levier vers une convergence des Technologies de l'Information (IT) et des Technologies Opérationnelles (OT), permettant de faciliter le transfert des données dans un contexte critique en utilisant une technologie non propriétaire et peu coûteuse. La sécurité des réseaux TSN est une question peu abordée dans la littérature mais essentielle pour permettre son adoption à grande échelle. Malgré une conception sécuritaire par défaut, certains standards ont des vulnérabilités identifiées et il n'existe pas d'outil de vérification de l'intégrité d'un réseau TSN. La complexité de configuration des standards peut amener à des réseaux TSN opérationnels mais vulnérables à leur exploitation par un attaquant désirant nuire aux performances du système. L'objectif de cet article est d'aborder le thème de la sécurité de TSN à travers le prisme des paramètres de configuration ayant un impact sur la robustesse du réseau.**

***Index Terms*—Time-Sensitive Networking, IEEE 802.1, Latence, Sécurité, Configuration**

I. Introduction

L'émergence de nouveaux champs applicatifs comme l'industrie 4.0 [1] ou les voitures connectées [2] a favorisé le développement des systèmes de communication à latence contrainte. Historiquement, les transferts de données étaient, par exemple, effectués via des bus de données CAN [3] pour les voitures ou des PROFIBUS [4] pour l'industrie. Ethernet a plus récemment été identifié comme une alternative puissante et peu chère pour remplacer ces anciennes solutions propriétaires. Si Ethernet n'offre aucune garantie quant au temps d'acheminement des trames, l'ensemble de standards Time-Sensitive Networking (TSN) étend Ethernet pour offrir des communications déterministes, et ainsi permettre le déploiement d'applications critiques.

La configuration d'un réseau TSN est un enjeu majeur de complexité, notamment pour des réseaux étendus à plusieurs dizaines de nœuds. Le groupe de travail IEEE 802.1 travaille actuellement sur des profils de configuration pour l'industrie et l'automobile, mais ceux-ci ne sont pas encore publiés. La littérature aborde le thème de la configuration en la simplifiant, grâce à divers *frameworks* d'implémentation ou méthodes de configuration. La conception sécuritaire de TSN peut-être impactée par ces méthodes de configuration, notamment lorsque le réseau n'est pas doté de composants de sécurité comme MACsec [5].

Pour répondre au problème de l'exploitation des mauvaises configurations, nous proposons d'analyser et d'identifier les vulnérabilités récurrentes des méthodes de configuration TSN. Cette analyse permettra de chercher des contremesures à ces failles, puis à terme de créer un outil de vérification de configuration pour TSN.

Cet article est structuré comme suit : dans la section II, nous proposons un état de l'art des standards principaux de TSN au travers de ses quatre composantes : synchronisation temporelle, latence faible et bornée, fiabilité et gestion des ressources. La section III est un état de l'art synthétique à la fois sur la sécurité de TSN et sur ses méthodes de configuration. Enfin, la section IV est dédiée aux travaux futurs envisagés sur ce sujet, notamment sur l'identification de l'impact des erreurs de configuration et des contremesures à apporter.

II. État de l'art des Principaux Standards Time-Sensitive Networking

Le terme TSN désigne un ensemble de standards qui peuvent être utilisés individuellement ou en combinaison. Les standards opérent sur la couche liaison de données du modèle OSI, au même niveau qu'Ethernet dont ils sont une extension. Au nombre de 28 publiés au moment de la rédaction de cet article [6], ces standards TSN se concentrent en quatre fonctionnalités.

A. Synchronisation Temporelle

Certain standards comme 802.1Qbv ou 802.1Qci nécessitent que les nœuds du réseau soient synchronisés entre eux. Pour répondre à ce besoin, 802.1AS propose une généralisation du *Precision Time Protocol*, gPTP. gPTP est fondé sur le *Best Master Clock Algorithm* (BMCA), un algorithme qui permet de chercher au sein du réseau local le nœud qui a la meilleure horloge interne. Une fois le Grand Maître[1] de l'horloge élu par le BMCA, celui-ci propage périodiquement son horloge au reste du réseau pour que chaque nœud puisse s'y synchroniser.

Le choix de l'intervalle de synchronisation a un impact sur le réseau : un intervalle trop petit offre une meilleure synchronisation mais une surcharge de communications sur

[1]Le Grand Maître est la machine désignée par le BMCA comme ayant la meilleur horloge interne du réseau.

le réseau, tandis qu'un intervalle trop grand peut engendrer un décalage d'horloge [2].

B. Latence Faible et Bornée

La composante principale de TSN est la latence faible et bornée. On retrouve dans cette deuxième catégorie les standards appartenant à la classe des *shapers* qui mettent en forme le trafic au sein des files d'attente des commutateurs TSN pour un transfert plus efficace [7]. Ces *shapers* visent à gérer les communications grâce à des classes de trafic qui permettent d'atteindre une latence bornée pour les trames critiques.

Le *Credit-Based Shaper* (CBS), introduit dans 802.1Qav et utilisé principalement pour les flux vidéos, fonctionne grâce à un mécanisme de crédits attribués périodiquement à chaque flux de données. Un flux a besoin de crédits disponibles pour transmettre ses trames, sinon il doit attendre le prochain réapprovisionnement. Le CBS permet d'éviter le monopole de transfert par un seul type de trafic [8]. La configuration de CBS prend en compte plusieurs paramètres, notamment l'*idleSlope* qui indique la quantité de crédit attribuée à chaque classe de trafic. Une mauvaise valeur d'*idleSlope* pourrait bloquer le trafic *best-effort*, ou inversement bloquer le trafic des trames critiques.

Le *Time-Aware Shaper* (TAS), introduit dans 802.1Qbv et utilisé majoritairement pour les flux les plus critiques, est fondé sur un mécanisme d'étiquettage de priorité des trames Ethernet. Ces niveaux de priorité sont utilisés dans chaque commutateur pour envoyer les trames de manière cyclique grâce à un mécanisme de portes. Le planning d'ouverture des portes est contrôlé par la *Gate-Control List* (GCL). Une configuration optimale de cette GCL est un problème d'ordonnancement qui implique de pouvoir planifier les plannings d'arrivée et de sortie des flux de données à chaque nœud du réseau. L'établissement des plannings fait ressortir une hyper-période[2], autre paramètre à observer pour optimiser la bande passante [9]. L'implémentation du TAS implique le choix d'un *guardband*, paramètre de configuration qui ajoute un délai entre chaque cycle d'ouverture des portes. Il permet d'éviter le blocage de transmission de nouvelles trames en attente de la transmission d'une trame d'un cycle précédent. Un *guardband* trop long réduit le volume de transfert global, tandis qu'un trop petit crée un risque pour les contraintes de latence.

On retrouve également 802.1Qbu classé dans cette catégorie, un mécanisme de préemption de trames qui vise à découper les trames en morceaux pour pouvoir interrompre le transfert d'une trame de basse priorité au profit d'une trame de haute priorité. La taille des morceaux est un autre paramètre à configurer, permettant notamment de réduire la taille du *guardband*.

[2]L'hyper-période en ordonnancement temps réel désigne le plus petit multiple commun des périodes des différentes tâches.

C. Fiabilité et Disponibilité

Cette troisième catégorie est composée de trois standards qui ciblent la détection rapide des erreurs et proposent les contre-mesures associées. Le premier standard de cette catégorie est 802.1CB *Frame Replication and Elimination* (FRER), qui crée de la redondance dans le réseau pour résister aux défaillances matérielles. FRER a plusieurs leviers de configuration, notamment la définition des chemins de redondance ou le nombre de messages redondants.

802.1Qci *Per-Stream Filtering and Policing* (PSFP) est un standard de disponibilité et de Qualité de Service (QoS) qui filtre les trames entrantes grâce à des tables de règles prédéfinies [10]. Le PSFP est actuellement le seul standard TSN publié dont la fonction est d'abord sécuritaire.

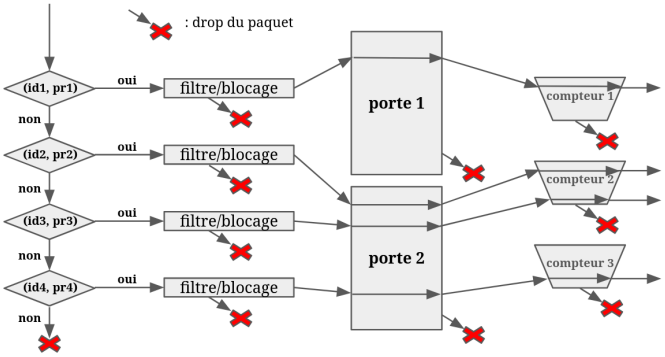

Fig. 1. Architecture de 802.1Qci PSFP (figure inspirée de [10])

Le PSFP est composé de 3 tables : une table de blocs de règles de filtrage, une table de portes, puis une table de compteurs. La Fig. 1 reprend le chemin qu'effectuent les trames dans un composant PSFP. Les tables "filtre/blocage" sont fondées sur un ensemble de règles de filtrage pré-établies. Ensuite, les portes fonctionnent comme dans le TAS. Enfin, les compteurs utilisent le principe de seau à jetons, en supprimant les trames quand la file est pleine. Chacun de ces composants nécessite d'être configuré individuellement, et ce dans chaque nœud d'un réseau TSN. Si le planning de la GCL ou si les seuils de remplissage des compteurs sont mal configurés, les trames seront automatiquement rejetées menant ainsi à l'arrêt du système.

D. Gestion des Ressources et Déploiement

Cette quatrième catégorie aborde la configuration des ressources réseau et le déploiement de TSN pour différentes utilisations. 802.1Qat *Stream Reservation Protocol* (SRP) qui permet l'enregistrement et la réservation de ressources des *switch*. 802.1Qcw traite de l'utilisation de YANG. Le standard 802.1Qcc propose trois modèles de configurations pour TSN : un système décentralisé, hybride ou centralisé, avec des compatibilités diverses avec les autres standards TSN.

III. État de l'art sur la Sécurité et la Configuration de TSN

A. Revue Systématique de la Littérature

Depuis le renommage du groupe en 2012, on retrouve 1693 articles qui abordent le sujet de TSN, dont 1056 qui l'ont comme sujet principal. Les articles concentrés sur TSN peuvent décrire des standards [10], proposer des frameworks d'implémentation [11], questionner l'utilisation de TSN pour différents domaines [12] ou encore présenter des améliorations de certains standards [13]. Afin de s'intéresser aux vulnérabilités connues de TSN mais également aux vulnérabilités des mauvaises configurations, nous avons scindé notre recherche en deux parties : sécurité et configuration.

B. Sécurité des Réseaux TSN

Pour se pencher sur les vulnérabilités connues de TSN et les différents composants de sécurité associés, on effectue la recherche scopus suivante :

- `TITLE-ABS-KEY("time-sensitive network*") AND PUBYEAR > 2012 AND TITLE("attack" OR "security" OR "vulnerabilit*" OR "threat*" OR "safety")`

Avec seulement 55 résultats, il apparaît que le sujet de la sécurité de TSN est peu abordé dans la littérature en comparaison du nombre total d'articles sur TSN. L'article sur la sécurité de TSN le plus cité est [14] et parle de la sécurité globale de TSN. Il reprend plus de 30 vulnérabilités potentielles issues de la conception de TSN grâce au modèle STRIDE [15]. On note cependant l'absence de *testbed* afin de tester l'impact des différentes vulnérabilités potentielles, ainsi que des scénarios qui présentent souvent comme prérequis l'élévation de privilèges sur une machine infectée du réseau.

Dans [16], la sécurité d'un réseau TSN a été prise en considération lors de la configuration du routage et de la planification afin d'éviter les interférences malveillantes d'autres flux tout en garantissant la QoS requise par les flux critiques légitimes. Les fonctions de sécurité reposent sur des clés dont la gestion a été automatisée dans [17] afin de simplifier le rôle de l'administrateur qui n'a plus besoin de connaître la topologie du réseau sous-jacent.

Plusieurs vulnérabilités sont liées à la confidentialité et à l'intégrité, notamment dans FRER et gPTP. Pour y répondre, on peut utiliser le mécanisme de sécurité MACsec qui permet de chiffrer les communications grâce à un échange de clés. Dans la littérature, certains articles étudient l'impact de MACsec sur le trafic et la latence dans TSN [18], [19]. Ces articles montrent que les algorithmes d'authentification et chiffrement de MACSec sont parallélisables et peuvent n'avoir qu'un impact très faible, voire négligeable sur la latence totale, rendant donc MACsec compatible avec le caractère faible latence de TSN.

C. Configuration de TSN

La configuration des standards pris individuellement est un enjeu de complexité, avec par exemple l'ordonnancement du TAS qui est un problème NP-complet [20]. Une partie de l'état de l'art de TSN est donc dédié à la configuration de ces standards. Une deuxième recherche Scopus renvoie 46 articles qui abordent le thème de la configuration.

Les auteurs de [21] proposent un environnement logiciel de configuration pour l'industrie fondé sur Sysrepo, utilisant le modèle de données YANG. NeSTiNg [22] est un environnement TSN fondé sur OMNeT++ qui est repris dans de nombreux articles qui traitent de la configuration de TSN. Il permet de simuler un réseau TSN et intègre des standards essentiels comme Qbv ou Qbu. La question du matériel physique disponible compatible avec les standards TSN est traitée dans [12].

[23], [24] proposent une implémentation de TSN avec la technologie OPC-UA, avec [24] qui la combine également avec du *Software-Defined Networking*. OPC-UA est un protocole de communication industriel qui est ici utilisée comme méthode de reconfiguration de la topologie réseau. [25], [26] posent la question d'un réseau TSN dynamique avec des fonctions d'auto-configuration qui réagissent aux changements de topologie du réseau. [26] utilise également OPC-UA et NETCONF pour la configuration dynamique des composants TSN aux switch. La configuration des GCL peut être synthétisée via du *Deep Reinforcement Learning* [27], avec une planification efficace pour 85% des flux sur une expérience à 150 flux périodiques.

D. Limites de l'État de l'art

La jonction des articles sur la sécurité et des articles sur la configuration ne renvoie pas de résultats. En effet, c'est une zone non abordée de la littérature. Pourtant, l'impact d'une mauvaise configuration ou de l'utilisation d'un environnement d'automatisation de la configuration sur la sécurité du système peut être majeur, comme on a pu le voir en section II-C.

IV. Objectifs de Recherche et Contributions

L'objectif de recherche à terme pour la sécurité de TSN est la génération de configurations robustes. Pour répondre à cette question, nous allons pouvoir nous concentrer sur trois élèments. D'abord, l'identification de l'impact sur la sécurité des erreurs de configuration dans un système TSN. Ensuite, nous proposerons des contremesures pour réduire ou négliger cet impact sur la sécurité du système. Enfin, nous pourrons proposer un outil de vérification de la robustesse d'une configuration.

A. Identification de l'impact des erreurs de configuration

La première étape de cette identification est la classification des erreurs de configuration pour analyser leurs verrous. On distingue trois catégories d'erreurs de configuration :

1) *Erreur de configuration sur un composant TSN* : Quelles erreurs sur des composants comme Qbv ou Qav impacteraient la sécurité du système TSN ? Comment réduire la possibilité d'occurence de ces erreurs ? Quel compromis est acceptable entre simplification et propriétés de sécurité ?

2) *Erreur de configuration de la sécurité de TSN* : Quel impact sur la sécurité aurait une erreur de configuration d'un composant de sécurité comme MACsec ? Quel compromis peut-on faire entre fonctionnalité et sécurité de ces composants ?
3) *Erreur de configuration globale inter-composants* : Quels impacts mutuels ont les composants entre eux ? MACsec crée-t-il trop de latence pour le reste du réseau ? Les implémentations de fonctions de sécurité supplémentaires rendent-elles TSN inutilisable ?

L'identification sera effectuée d'abord via une recherche systématique de la littérature sur les modèles de configuration et sur les composants de sécurité de TSN, puis via l'implémentation d'un *testbed* afin de quantifier l'impact des erreurs de configuration. Pour simplifier l'analyse, la mise en place du *testbed* impliquera d'abord de choisir un cas d'étude et de définir une topologie associée, avant de pouvoir être exhaustif sur tous les cas d'usage par la suite. Dans le cas d'un *testbed* physique et non-simulé, il nous faudra également faire un choix parmis les différents appareils TSN-compatibles.

B. Contremesures

Dans un deuxième temps, nous souhaitons traiter les problèmes de configuration identifiés en proposant des contremesures dans un objectif de réduction des risques de chaque vulnérabilité identifiée. La recherche des contremesures pourra passer par l'expérimentation grâce au *testbed*, notamment pour éviter les attaques à la latence qui dénaturent TSN.

C. Outils de vérification et génération

L'analyse manuelle des vulnérabilités et la recherche de contremesures nous permettra de proposer d'abord un outil automatisé de vérification de l'intégrité d'un système TSN, qui pourra vérifier une liste de vulnérabilités identifiées. À notre connaissance, il n'existe pas de tel outil dans la littérature, cependant on retrouve des outils de vérification de configuration des *shapers* qui s'assurent du respect des délais pour un ensemble de flux de différentes priorités. [28] propose un tel outil de vérification des plannings via une méthode hybride fondée sur le *supervised learning* et sur des méthodes traditionnelles d'analyse d'ordonnancement. Finalement, on pourra créer un outil de génération de configurations robustes pour TSN.

V. Conclusion

Le *Time-Sensitive Networking* (TSN) est une extension d'Ethernet qui vise à le rendre déterministe. Par sa nature sécuritaire par défaut, il ne persiste que peu de vulnérabilités dans sa conception. En revanche, TSN est complexe à configurer et de mauvaises configurations peuvent rendre le réseau vulnérable aux attaques. Des méthodes de simplification de configuration existent mais ne couvrent pas tous les protocoles de TSN et peuvent impacter la sécurité du système. Nous proposons d'identifier ces vulnérabilités qui apparaissent dans de mauvaises configurations, de chercher des contremesures et de créer un outil de vérification de la robustesse d'un réseau TSN.

References

[1] L. Lo Bello and W. Steiner, "A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1094–1120, Jun. 2019, conference Name: Proceedings of the IEEE. [Online]. Available: https://ieeexplore.ieee.org/document/8695835/?arnumber=8695835

[2] H.-T. Lim, D. Herrscher, L. Völker, and M. J. Waltl, "IEEE 802.1AS time synchronization in a switched Ethernet based in-car network," in *2011 IEEE Vehicular Networking Conference (VNC)*, Nov. 2011, pp. 147–154, iSSN: 2157-9865. [Online]. Available: https://ieeexplore.ieee.org/document/6117136/?arnumber=6117136

[3] T. Nolte, H. Hansson, and L. Bello, "Automotive communications-past, current and future," in *2005 IEEE Conference on Emerging Technologies and Factory Automation*, vol. 1, Sep. 2005, pp. 8 pp.–992, iSSN: 1946-0759. [Online]. Available: https://ieeexplore.ieee.org/document/1612631/?arnumber=1612631

[4] E. Tovar and F. Vasques, "Real-time fieldbus communications using Profibus networks," *IEEE Transactions on Industrial Electronics*, vol. 46, no. 6, pp. 1241–1251, Dec. 1999, conference Name: IEEE Transactions on Industrial Electronics. [Online]. Available: https://ieeexplore.ieee.org/document/808018/?arnumber=808018

[5] "IEEE Standard for Local and metropolitan area networks-Media Access Control (MAC) Security," *IEEE Std 802.1AE-2018 (Revision of IEEE Std 802.1AE-2006)*, pp. 1–239, Dec. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8585421

[6] "Time-Sensitive Networking (TSN) Task Group |." [Online]. Available: https://1.ieee802.org/tsn/

[7] N. Finn, "Introduction to Time-Sensitive Networking," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 22–28, Jun. 2018, conference Name: IEEE Communications Standards Magazine. [Online]. Available: https://ieeexplore.ieee.org/document/8412458/?arnumber=8412458

[8] J. Pei, Y. Hu, and L. Tian, "A Review on Key Mechanisms of Time-Sensitive Networking," in *2021 International Conference on Advanced Computing and Endogenous Security*, Apr. 2022, pp. 01–07. [Online]. Available: https://ieeexplore.ieee.org/document/10013335/?arnumber=10013335

[9] M. Vlk, K. Brejchová, Z. Hanzálek, and S. Tang, "Large-scale periodic scheduling in time-sensitive networks," *Computers & Operations Research*, vol. 137, p. 105512, Jan. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054821002549

[10] M. Boyer, "Usage of TSN Per-Stream Filtering and Policing," 2023. [Online]. Available: https://hal.science/hal-04159172

[11] B. Houtan, A. Bergström, M. Ashjaei, M. Daneshtalab, M. Sjödin, and S. Mubeen, "An Automated Configuration Framework for TSN Networks," in *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, vol. 1, Mar. 2021, pp. 771–778. [Online]. Available: https://ieeexplore.ieee.org/document/9453628/?arnumber=9453628

[12] D. Hallmans, M. Ashjaei, and T. Nolte, "Analysis of the TSN Standards for Utilization in Long-life Industrial Distributed Control Systems," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, Sep. 2020, pp. 190–197, iSSN: 1946-0759. [Online]. Available: https://ieeexplore.ieee.org/document/9212162

[13] L. Murselovic, "PERFORMANCE ANALYSIS OF THE PREEMPTION MECHANISM IN TSN."

[14] D. Ergenç, C. Brülhart, J. Neumann, L. Krüger, and M. Fischer, "On the Security of IEEE 802.1 Time-Sensitive Networking," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, Jun. 2021, pp. 1–6, iSSN: 2694-2941. [Online]. Available: https://ieeexplore.ieee.org/document/9473542/?arnumber=9473542

[15] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Threat modeling-uncover security design flaws using the stride approach," *MSDN Magazine*, pp. 68–75, Jan. 2006.

[16] M. Letourneau, G. Doyen, R. Cogranne, and B. Mathieu, "A Comprehensive Characterization of Threats Targeting Low-Latency Services: The Case of L4S," *Journal of Network & Systems Management*, vol. 31, no. 1, pp. 1–33, Mar. 2023, publisher: Springer Nature.

[17] R. Mahfouzi, A. Aminifar, S. Samii, P. Eles, and Z. Peng, "Security-aware Routing and Scheduling for Control Applications on Ethernet TSN Networks," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 25, no. 1, pp. 1:1–1:26, Nov. 2019. [Online]. Available: https://dl.acm.org/doi/10.1145/3358604

[18] R. A. Peña, M. Pascual, A. Astarloa, D. Uribe, and J. Inchausti, "Impact of MACsec security on TSN traffic," in *2022 37th Conference on Design of Circuits and Integrated Circuits (DCIS)*, Nov. 2022, pp. 01–06, iSSN: 2640-5563. [Online]. Available: https://ieeexplore.ieee.org/document/9970155/?arnumber=9970155

[19] D. Dik, I. Larsen, and M. Stübert Berger, "MACsec and AES-GCM Hardware Architecture with Frame Preemption Support for Transport Security in Time Sensitive Networking," in *2023 International Conference on Computer, Information and Telecommunication Systems (CITS)*, Jul. 2023, pp. 01–07. [Online]. Available: https://ieeexplore.ieee.org/document/10188711/?arnumber=10188711

[20] B. J. Mackenzie, F. Bruns, and W. Nebel, "Model-based Automation of TSN Configuration for Industrial Distributed Systems," in *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*, Jul. 2023, pp. 1–6, iSSN: 2378-363X. [Online]. Available: https://ieeexplore.ieee.org/document/10218085/?arnumber=10218085

[21] S. Oechsle, F. Frick, A. Lechler, and A. Verl, "A modular configuration and management framework for distributed real-time applications based on converged networks using TSN," *Procedia CIRP*, vol. 118, pp. 38–43, 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2212827123002305

[22] J. Falk, D. Hellmanns, B. Carabelli, N. Nayak, F. Dürr, S. Kehrer, and K. Rothermel, "NeSTiNg: Simulating IEEE Time-sensitive Networking (TSN) in OMNeT++," in *2019 International Conference on Networked Systems (NetSys)*, Mar. 2019, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8854500

[23] Z. Zhou and G. Shou, "An Efficient Configuration Scheme of OPC UA TSN in Industrial Internet," in *2019 Chinese Automation Congress (CAC)*, Nov. 2019, pp. 1548–1551, iSSN: 2688-0938. [Online]. Available: https://ieeexplore.ieee.org/document/8996369/?arnumber=8996369

[24] T. Kobzan, I. Blöcher, M. Hendel, S. Althoff, A. Gerhard, S. Schriegel, and J. Jasperneite, "Configuration Solution for TSN-based Industrial Networks utilizing SDN and OPC UA," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, Sep. 2020, pp. 1629–1636, iSSN: 1946-0759. [Online]. Available: https://ieeexplore.ieee.org/document/9211897/?arnumber=9211897

[25] M. Pahlevan, J. Schmeck, and R. Obermaisser, "Evaluation of TSN Dynamic Configuration Model for Safety-Critical Applications," in *2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, Dec. 2019, pp. 566–571. [Online]. Available: https://ieeexplore.ieee.org/document/9047392/?arnumber=9047392

[26] M. Gutiérrez, A. Ademaj, W. Steiner, R. Dobrin, and S. Punnekkat, "Self-configuration of IEEE 802.1 TSN networks," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2017, pp. 1–8, iSSN: 1946-0759. [Online]. Available: https://ieeexplore.ieee.org/document/8247597/?arnumber=8247597

[27] M. Guo, G. Shou, Y. Liu, and Y. Hu, "Adaptive Configuration with Deep Reinforcement Learning in Software-Defined Time-Sensitive Networking," in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, May 2024, pp. 1–7, iSSN: 2374-9709. [Online]. Available: https://ieeexplore.ieee.org/document/10575223/?arnumber=10575223

[28] T. L. Mai, N. Navet, and J. Migge, "A Hybrid Machine Learning and Schedulability Analysis Method for the Verification of TSN Networks," in *2019 15th IEEE International Workshop on Factory Communication Systems (WFCS)*, May 2019, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8757948