Anomaly Diagnosis and Self-Protection in Time-Sensitive Networks

Gioacchino Sterlicchio, postdoctoral researcher at IMT-Atlantique
gioacchino.sterlicchio@imt-atlantique.fr

1 Scientific Context and Motivation

Time-Sensitive Networking (TSN) [3] defines a suite of IEEE 802.1 standards designed to guarantee bounded latency, zero congestion loss, and deterministic delivery for mission-critical applications such as automotive, avionics, and industrial automation. However, because time itself is a critical vector in deterministic networks, microsecond delays or subtle flow rate anomalies induced by zero-day attacks (e.g., Reduction-of-Quality, stream injection, or gate-timing manipulation) can lead to catastrophic system failures [1].

Traditional intrusion detection systems (IDS) rely on supervised machine learning or fixed attack signatures. While unsupervised anomaly detectors can flag unseen zero-day anomalies, they function as "black boxes" and fail to explain which security requirement or Quality of Service (QoS) invariant has been breached. Inspired by recent advances in smart home security [5], this project addresses the diagnostic gap by combining unsupervised anomaly detection with Answer Set Programming (ASP) [2] and abductive reasoning [4] by refutation to enable explainable, real-time diagnosis and automated self-protection in TSN architectures. ASP is a declarative logic programming paradigm and unlike imperative or procedural languages where control flow is explicitly programmed (*how to solve*), ASP allows engineers to specify domain knowledge, operational assumptions, and security invariants declaratively (*what the system and its requirements are*). ASP is particularly well-suited for cyber-physical architectures because of its non-monotonic reasoning capabilities, which enable reasoning over partial, evolving system specifications without requiring exhaustive, monolithic system models. Instead, abductive reasoning infers the most plausible causes or explanations for an observed effect. In runtime security monitoring, raw anomaly detectors identify *that* a network deviation has occurred, but cannot explain *why*. To bridge runtime network traces with the system's high-level threat model, the framework employs abduction by refutation. When an anomalous trace is injected into the ASP system model, it creates a logical contradiction (unsatisfiability) against the security requirements. Abductive refutation sys-

tematically evaluates which security requirement or invariant must be relaxed for the system model to regain satisfiability, thereby precisely pinpointing the violated security requirement and the corresponding class of attack.

2 Research Problem and Core Methodology

The proposed framework establishes a continuous, four-stage self-protection cycle designed to detect, diagnose, and mitigate anomalies in TSN.

2.1 System and Requirement Modelling in ASP

- Formalize the TSN network topology (talkers, listeners, bridges), stream descriptors, and traffic control mechanisms (IEEE 802.1Qci gate states, burst sizes, rate limits) as a logic program in ASP.
- Encode high-level safety, QoS, and security requirements (e.g., maximum stream bandwidth, strict gate time windows, non-interfering priority queues) as ASP integrity constraints (`:- ...`).

2.2 Runtime Anomaly Detection and Contextual Fact Translation

- Process TSN execution traces using unsupervised anomaly detection (e.g., Isolation Forest) trained strictly on benign traffic.
- Automatically translate flagged network deviations and metrological contextual facts (e.g., stream ID, measured jitter variance, packet rate exceedance, gate state) into declarative ASP predicates (`communicate(...)`, `stream_rate(...)`).

2.3 Abductive Diagnosis by Refutation

- Perform abduction by refutation by inserting the anomalous trace into the ASP model, check for model unsatisfiability, and systematically evaluate which security constraint must be relaxed for the system to remain satisfiable.
- The output directly identifies the violated security requirement and the class of attack/misconfiguration (e.g., bandwidth violation, unauthorized stream spoofing, TAS timing misalignment).

2.4 Self-Protection and Rule Updating

Map the logical diagnosis into actionable mitigation policies e.g., dynamically reconfiguring IEEE 802.1Qci Per-Stream Filtering and Policing (PSFP) flow tables to drop or shape the anomalous stream at line rate.

3 Expected Deliverables

1. ASP System Model: documented ASP logic programs representing TSN assets, operational rules, and security constraints.
2. Experimental Dataset: repository containing dataset traces, the anomaly detector model, the ASP trace converter, and evaluation scripts.
3. Internship Report: comprehensive report document detailing the state of the art, methodology, experimental results, and future directions.

4 Desired Profile

- Education: student in Cybersecurity, Computer Science, or Formal Methods.
- Technical qualification:
  - Declarative logic programming or constraint solving.
  - Understanding of Ethernet/IP networking and deterministic Time-Sensitive Networking (TSN).
  - Knowledge of Machine Learning and related python frameworks (scikit-learn, pandas, ...).
- Ability to analyse scientific publications, delivering oral presentations, and writing reports.
- Autonomy, thoroughness, and analytical skills.

5 Working Language and Environment

- The internship will be conducted entirely in English.
- The final report, code documentation, presentations, and possibly a scientific paper draft, resulting from the results, must be written in English.

6 Practical Information and Application Procedure

- Location: IMT Atlantique (Rennes Campus) - SRCD Department - SOTERN Team.
- How to Apply: Interested candidates should send their application package in English via email to Gioacchino Sterlicchio (gioacchino.sterlicchio@imt-atlantique.fr). The application must include:

1. CV and academic results.
2. Cover letter covering experience and motivations.

- Start date: February/March 2027
- Duration: 6 months
- Application deadline: 18/12/2026

References

[1] Doğanalp Ergenç, Cornelia Brülhart, Jens Neumann, Leo Krüger, and Mathias Fischer. On the Security of IEEE 802.1 Time-Sensitive Networking. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6.

[2] V Lifschitz. Answer sets and the language of answer set programming. *AI Magazine*, 37(3):7–12, 2016.

[3] Lucia Lo Bello and Wilfried Steiner. A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems. *Proceedings of the IEEE*, 107(6):1094–1120, June 2019.

[4] Gabriele Paul. Approaches to abductive reasoning: an overview. *Artif. Intell. Rev.*, 7(2):109–152, 1993.

[5] Kushal Ramkumar, Wanling Cai, John McCarthy, Gavin Doherty, Bashar Nuseibeh, and Liliana Pasquale. Diagnosing unknown attacks in smart homes using abductive reasoning. *IEEE Trans. Software Eng.*, 51(11):3117–3137, 2025.