DETERMINISTIC6G

Report on the Security solutions

D3.2

The DETERMINISTIC6G project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 1010965604.

Report on the Security solutions (Software)

Grant agreement number: Project title: Project acronym: Project website: Programme:	101096504 Deterministic E2E communication with 6G DETERMINISTIC6G Deterministic6g.eu EU JU SNS Phase 1
Deliverable type:	Other (Software)
Deliverable reference number:	D3.2
Contributing workpackages:	WP3
Dissemination level:	Public
Due date:	31-12-2023
Actual submission date:	21-12-2023
Responsible organization:	Montimage
Editor(s):	Edgardo Montes de Oca, Huu Nghia Nguyen
Version number:	1.1
Status:	Final
Short abstract:	The deliverable details the initial security monitoring framework for the DETERMINISTIC6G project, encapsulating the challenges and standards for deterministic network security and outlining the measures for real-time threat detection and response. This framework is pivotal for the project's security monitoring software release, which is in the process of incorporating AI/ML algorithms, in-band networking, and programmable data planes within a dynamic 3GPP Zero-touch Service Management model. It is designed to flexibly address evolving security scenarios, offering precise network telemetry for continuous monitoring and rapid countermeasure deployment, thereby establishing a proactive, adaptable defence for low-latency and deterministic network applications.
Keywords:	TSN, 6G, DetNet, Time Sensitive Networking, network monitoring, security-by-design, Zero-touch Service and network Management (ZSM), In-band Network Telemetry (INT), cybersecurity, Denial of Service (DoS)

Contributor(s):	Montimage (MI): Edgardo Montes de Oca, Huu Nghia Nguyen

Revision History

V0.1 (6/11/2023)	Version ready for internal review
V0.2 (6/12/2023)	Comments addressed
V1.0 (7/12/2023)	Final version for review by Project Management Team (PMT)
V1.1 (20/12/2023)	Final version ready for submission

Disclaimer

Executive summary

This document presents an open-source software framework released as *Deliverable 3.2 - Report on the Security solutions.* It provides a first description of the security monitoring framework established in the DETERMINISTIC6G project, concomitant with the software release, and presents the necessary context. It synthesizes the challenges, requirements, and potential vulnerabilities alongside established standards and security-by-design principles vital for deterministic network security monitoring. Drawing on initial architectural and security concerns outlined in previous work done in the DETERMINISTIC6G project, it introduces the first iteration of the security framework tailored to deterministic networking's demands.

The framework enhances the 6G deterministic communication systems' security, enabling real-time traffic analysis, precise monitoring, and prompt activation of countermeasures. Cyber-security is a vital concern for applications that need communications that are reliable and ensure a user-defined threshold (i.e., latency) for the delivery of the communication packets. Security breaches can easily degrade the latency, and this can be fatal for the quality of many applications and, in safety critical ones, can endanger human lives or compromise critical assets.

The framework leverages the requirements of the DETERMINISTIC6G use cases. It is designed for flexibility, and provides an essential enabler for the 3GPP Zero-touch Service Management (ZSM) model, ensuring the system can dynamically adapt to varying security needs across different scenarios. The ability to automatically adjust defence strategies in real-time, informed by threat intelligence, is crucial for managing the ever-evolving risk landscape.

At its core, the framework will rely on advanced Artificial Intelligence/Machine Learning (AI/ML) algorithms for intelligent detection and anticipatory threat mitigation, supporting a proactive security stance. It currently leverages in-band networking and programmable data planes for real-time policy enforcement, complemented by high-precision telemetry techniques that provide a detailed operational view of the time sensitive aspects of the network, i.e., time synchronisation and latency. This detailed insight is important for maintaining stringent security standards in low (or bounded) latency and deterministic network applications, ensuring robust and adaptable security solution.

The research outlined herein addresses a fundamental enabler for identifying and countering these cyber threats. This represents an initial stride towards establishing and delivering a security-by-design architecture specifically tailored to meet the stringent demands of deterministic networking targeted by the DETERMINISTIC6G project.

Contents

1 Introduction

As highlighted in the DETERMINISTIC6G's recent survey [SPS+23] and in phase with the goals of the DETERMINISTIC6G project (summarized in the next Section 1.1), there is a critical need for scalable and adaptable monitoring of the End-to-End (E2E) deterministic network chain. This requirement is essential for enhancing trust, trustworthiness, liability, resiliency, and reliability. These factors must be taken into account from all perspectives, encompassing the architecture design, component development, deployment, network and application operation, as well as ongoing maintenance and evolution. This comprehensive approach underscores the necessity of implementing security-by-design principles, ensuring that security considerations are integral to each phase of the network lifecycle.

The network monitoring framework can be seen as a fundamental enabler of an E2E security architecture considering the requirements of deterministic networking in wireless, virtualized infrastructures, and edge-driven heterogeneous networks. It leverages the latest technological paradigms, aiming to fortify the integrity and reliability of the data communications of low-latency and deterministic applications.

This deliverable builds upon the results obtained from the INSPIRE-5Gplus project [Inspire-5Gplus], particularly its closed-loop security architecture and monitoring framework. These elements are being tailored and expanded to meet the specific needs of deterministic networking as outlined by the DETERMINISTIC6G project and its associated use cases. Key to this adaptation is the employment of emerging technologies such as In-band Network Telemetry (INT), Programming Protocol-independent Packet Processors (P4) programming, and AI/ML. These technologies are crucial for achieving the detailed security analytics and management necessary to guarantee deterministic communication. By integrating these technologies into the security monitoring and management framework, we are developing an innovative solution for safeguarding E2E deterministic communications. The novelty of this work lies in the ongoing development of a security-by-design architecture and methodologies specifically designed to tackle the unique challenges posed by the DETERMINISTIC6G project.

Applications that depend on deterministic networking are especially vulnerable to cyber threats, which can disrupt time synchronization, compromise the reliability of communications, and breach the required latency thresholds critical their network traffic.

From the perspective of a deterministic and reliable 6G network, there are specific challenges and exigent real-time requirements, such as the deployment of Time-Sensitive Networking (TSN) capabilities. An example is the time-aware shaping feature defined in the IEEE 802.1Qbv [IEEE-8021Qci] standard, which is highly susceptible to traffic overloads that can derail the intended scheduling. Additionally, protocols for clock synchronization are also at risk. These vulnerabilities underscore the need for new, more fine-grained security measures capable of detecting and mitigating attacks on these critical functionalities. The adoption of P4 programmable data planes and INT is emerging as a promising approach that is implemented in this first version of the monitoring framework for the DETERMINISTIC6G project. These techniques offer the means to deliver the precise security metrics needed and, at the same time, the required adaptability and minimal interference in network operations.

The proposed architecture must be intrinsically designed to be versatile, ensuring compatibility and adaptability within multi-domain contexts that are often characterized by complex operational landscapes, i.e., highly complex network structures, including massive IoT networks and dense urban deployments with small cells, all integrated into a seamless network. The report emphasizes the importance of a security framework that can align with, and facilitate, the various business requirements that a multi-provider ecosystem entails, ensuring that all parties can maintain their own security standards while operating within a cohesive, secure network.

The security management component of the architecture is envisioned to be both dynamic and automated, featuring state-of-the-art mechanisms that can swiftly adapt to a continuously evolving landscape of risks and threats. This agility is crucial in an era where threat vectors are not only multiplying but also becoming more sophisticated. The security framework is, therefore, tasked with the ability to automatically update its defence measures in real-time, achieving zero-touch security management, guided by the latest threat intelligence, and adapting its response protocols to mitigate risks proactively.

Central to this adaptive security management is the implementation of robust AI/ML algorithms capable of intelligent detection and pattern recognition, which informs the identification of potential vulnerabilities and emerging threats. By integrating these algorithms, the system is endowed with a predictive capability, anticipating security incidents before they manifest, and thereby allowing the network to pre-emptively fortify its defences.

Furthermore, the security framework encompasses a comprehensive process pipeline for the collection and generation of network traffic data. This is a critical component for the continuous learning and evolution of the AI/ML models during both training and testing phases. Such data are integral in refining the models to accurately detect, and respond to, security and performance issues, particularly those that may compromise the stringent requirements of low-latency or deterministic network applications. This deliverable focuses on signature-based attack detection. AI/ML-based attack detection will be detailed in a future DETERMINISTIC6G project deliverable D3.5: Multi-domain end-to-end schedules [DET25-D3.5].

The report also underscores the significance of leveraging the latest advancements in In-Band Networking and P4-based programmable data planes. These technologies are instrumental in enabling the proposed security architecture to embed real-time security policy enforcement within the data plane itself, thus allowing for rapid and flexible security measures that can be adjusted on the fly in response to detected anomalies or policy changes.

High-precision telemetry techniques are highlighted as a cornerstone for the proposed framework, furnishing the system with granular data necessary for an in-depth understanding of the network's operational state. This high-resolution view is paramount for ensuring that security measures are based on the most current and detailed network performance metrics, allowing for a nuanced and highly effective security posture.

1.1 DETERMINISTIC6G approach

Digital transformation of industries and society is resulting in the emergence of a larger family of time-critical services with needs for high availability and which present unique requirements distinct from traditional Internet applications like video streaming or web browsing. Time-critical services are already known in industrial automation; for example, an industrial control application that might require an end-to-end "over the loop" (i.e., from the sensor to the controller back to the actuator) latency of 2 ms and with a communication service requirement of 99.9999% [3GPP16-22261]. But with the increasing digitalization similar requirements are appearing in a growing number of new application domains, such as extended reality, autonomous vehicles and adaptive manufacturing. The general long-term trend of digitalization leads towards a *Cyber-Physical Continuum* where the monitoring, control and maintenance functionality is moved from physical objects (like a robot, a machine or a tablet device) to a compute platform at some other location, where a digital representation – or digital twin – of the object is operated. Such Cyber Physical System (CPS) applications need a frequent and consistent information exchange between the digital and physical twins. Several technology developments in the ICT-sector drive this transition. The proliferation of (edge-) cloud compute paradigms provide new cost-efficient and scalable computing capabilities, that are often more efficient to maintain and evolve compared to embedded compute solutions integrated into the physical objects. It also enables the creation of digital twins as a tool for advanced monitoring, prediction and automation of system components and improved coordination of systems of systems. New techniques based on Machine Learning can be applied in application design, that can operate over large data sets and profit from scalable compute infrastructure. Offloading compute functionality can also reduce spatial footprint, weight, cost and energy consumption of physical objects, which is in particular important for mobile components, like vehicles, mobile robots, or wearable devices. This approach leads to an increasing need for communication between physical and digital objects, and this communication can span over multiple communication and computational domains. Communication in this cyber-physical world often includes closed-loop control interactions which can have stringent end-to-end KPI (e.g., minimum and maximum packet delay) requirements over the entire loop. In addition, many operations may have high criticality, such as business-critical tasks or even safety relevant operations. Therefore, it is required to provide *dependable time-critical communication* which provides communication service-assurance to achieve the agreed service requirements.

Time-critical communication has in the past been mainly prevalent in industrial automation scenarios with special compute hardware like Programmable Logic Controller (PLC), and is based on a wired communication system, such as EtherCat and Powerlink, which is limited to local and isolated network domains which is configured to the specific purpose of the local applications. With the standardization of Time-Sensitive Networking (TSN), and Deterministic Networking (DetNet), similar capabilities are being introduced into the Ethernet and IP networking technologies, which thereby provide a converged multi-service network allowing time critical applications in a managed network infrastructure allowing for consistent performance with zero packet loss and guaranteed low and bounded latency. The underlying principles are that the network elements (i.e. bridges or routers) and the PLCs can provide a consistent and known performance with negligible stochastic variation, which allows to manage the network configuration to the needs of time-critical applications with known traffic characteristics and requirements.

It turns out that several elements in the digitalization journey introduce characteristics that deviate from the assumptions that are considered as baseline in the planning of deterministic networks. There is often an assumption for compute and communication elements, and also applications, that any stochastic behavior can be minimized such that the time characteristics of the element can be clearly associated with tight minimum/maximum bounds. Cloud computing provides efficient scalable compute, but introduces uncertainty in execution times; wireless communications provides flexibility and simplicity, but with inherently stochastic components that lead to packet delay variations exceeding significantly those found in wired counterparts; and applications embrace novel technologies (e.g. ML-based or machine-vision-based control) where the traffic characteristics deviate from the strictly deterministic behavior of old-school control. In addition, there will be an increase in dynamic behavior where characteristics of applications, and network or compute elements may change over time in contrast to a static behavior that does not change during runtime. It turns out that these deviations of *stochastic characteristics* make traditional approaches to planning and configuration of end-to-end time-critical communication networks such as TSN or DetNet, fall short in their performance regarding service performance, scalability and efficiency. Instead, a revolutionary approach to the design, planning and operation of time-critical networks is needed that fully embraces the variability but also dynamic changes that come at the side of introducing wireless connectivity, cloud compute and application innovation. DETERMINISTIC6G has as objective to address these challenges, including the planning of resource allocation for diverse time-critical services end-to-end over multiple domains, providing efficient resource usage and a scalable solution [SPS+23].

DETERMINISTIC6G takes a novel approach towards converged future infrastructures for scalable cyber-physical systems deployment. With respect to networked infrastructures, DETERMINISTIC6G advocates (I) the acceptance and integration of stochastic elements (like wireless links and computational elements) with respect to their stochastic behavior captured through either short-term or longer-term envelopes. Monitoring and prediction of KPIs, for instance latency or reliability, can be leveraged to make individual elements plannable despite a remaining stochastic variance. Nevertheless, system enhancements to mitigate stochastic variances in communication and compute elements are also developed. (II) Next, DETERMINISTIC6G attempts the management of the entire end-to-end interaction loop (e.g. the control loop) with the underlying stochastic characteristics, especially embracing the integration of compute elements. (III) Finally, due to unavoidable stochastic degradations of individual elements, DETERMINISTIC6G advocates allowing for adaptation between applications running on top such converged and managed network infrastructures. The idea is to introduce flexibility in the application operation such that its requirements can be adjusted at runtime based on prevailing system conditions. This encompasses a larger set of application requirements that (a) can also accept stochastic end-to-end KPIs, and (b) that possibly can adapt end-to-end KPI requirements at run-time in harmonization with the networked infrastructure. DETERMINISTIC6G builds on a notion of time-awareness, by ensuring accurate and reliable time synchronicity while also ensuring security-by-design for such dependable time-critical communications. Generally, we extend a notion of deterministic communication (where all behavior of network and compute nodes and applications is pre-determined) towards dependable time-critical communication, where the focus is on ensuring that the communication (and compute) characteristics are managed in order to provide the KPIs and reliability levels that are required by the application. DETERMINISTIC6G facilitates architectures and algorithms for scalable and converged future network infrastructures that enable dependable time-critical communication end-to-end, across domains and including 6G.

1.2 Relation to other work packages

The work presented here concerns the first version of the security monitoring and management framework adapted to the specific requirements of deterministic networking. The requirements from the application domains and the overall security-by-design architecture, developed in WP1, serve as input for the design and implementation of the security framework. The project's deliverableD1.1: DETERMINISTIC6G use cases and architecture principles [DET23-D1.1] provided the initial requirements and architecture. The deliverable D2.2: First report on time synchronization for E2E time awareness [DET23-D2.2] provided a description of the security concerns and resilience requirements of the time synchronisation mechanisms.

The security framework developed here will be provided for the planned DETERMINISTIC6G's evaluation of the security mechanisms to demonstrate improved resiliency, privacy, and determine their efficiency in different E2E multi-party scenarios.

1.3 Objective of the document

This deliverable serves as a comprehensive first report on the security monitoring framework formulated and deployed within the DETERMINISTIC6G project. Its primary purpose is to augment the accompanying software release, furnishing the contextual background that encapsulates a succinct overview of the inherent challenges, prerequisites, potential vulnerabilities, prevailing standards, security-by-design architecture, and foundational techniques pertinent to the security monitoring of deterministic networking.

The essential contribution of the deliverable is the software solution and the artefacts (software documentation and instructions: user guide, configuration files, readme file), which can be found at the project's public Github repository and as a snapshot of this repository at the Zenodo platform. Links to the software and data sets are provided in Table 1.

Component name	License	Link
Security and performance monitoring framework	Apache 2.0	https://github.com/DETERMINISTIC6G/deterministic6g_security-solutions/ https://zenodo.org/communities/deterministic6g (https://zenodo.org/records/10401698)

Table 1: Software solution of this deliverable.

The proposed security framework is seen as an essential enabler to secure the system by design within the DETERMINISTIC6G project and applicable to future 6G deterministic networking in general. It allows real-time traffic monitoring, improved security data collection to provide high precision

monitoring, real-time analysis and reasoning, and the triggering of remediation mechanisms. The framework is designed to be flexible so that it can adapt to different security requirements, deployments, and scenarios. It constitutes part of the 3GPP ZSM model proposed in the Inspire-5Gplus project [Inspire-5Gplus], providing real-time monitoring to report the security incidents in a timely manner with the goal of re-regulating system parameters to mitigate ongoing incidents or avoid such incidents in the future.

1.4 Structure and scope of the document

After the Introduction, the deliverable continues in Section 2 with the definition of the relevant terms. The major content of the document starts in Section 3, which provides the context concerning the network monitoring framework, where a first description of the high-level security architecture and framework is provided. This section includes an introduction, the main security challenges, a brief state of the art, the initial security architecture and enablers, and two main security scenarios.

Section 4 and Section 5 then introduce a description of the main results of the work that includes the software solution for monitoring the latency and performance, and the software solution for attack generation and detection. In these sections, the mechanisms used by the framework that include non-disruptive monitoring, in-band network monitoring and signature-based attack detection are described.

Finally, Section 6 provides a conclusion of the document, and presents the future work planned during the lifetime of the project.

2 Terms and Definitions

Term	Definition	Notes
Determinism	All events in future can be determined completely based on past events and the laws of nature.	Also, referred as causal determinism in philosophy.
Dependable communication	Given a message, the correct transmission is guaranteed to be performed in a specified period (not faster, not slower).	
Latency	Latency describes the required time to send a packet from a given sender to a given receiver over a given network.	Sometimes also described as "*packet delay*" or "*network delay*".
Packet Delay Variation (PDV)	PDV describes the amount of variation of the latencies perceived when a series of messages are transmitted from a given sender to a given receiver over a given network.	Sometimes also described as "*jitter*".

P4 programming	A high-level programming language designed for controlling and forwarding data in network devices.	It enables programmers to define how network packets are processed, offering flexibility and customization in the management of network traffic.
In-band Network Telemetry (INT)	Network monitoring technology that embeds telemetry data about the packet's journey directly within the packet itself as it travels through the network.	This allows for real-time monitoring and diagnosis of network performance issues without the need for additional probing traffic.
Rule-based network security analysis	Involves using predefined criteria or rules to evaluate network traffic and detect security threats.	This method relies on specific conditions set by network administrators to identify and respond to potential security risks, such as unauthorized access or malicious activities.
Behaviour-based network security analysis	Involves monitoring network traffic for unusual or anomalous behaviour that deviates from established patterns.	Unlike rule-based analysis, which relies on predefined criteria, behaviour-based analysis learns normal network activity over time and flags deviations as potential security threats. This approach is effective in detecting unknown or emerging threats that do not match known attack signatures.

3 High-level Security Architecture and Framework

In this section, we specify and describe the design of a comprehensive E2E security architecture and framework for deterministic networking in virtualized and edge-based heterogeneous networking environments. The architecture needs to be adapted to multi-domain, multi-stakeholder, and multi-provider business requirements, and offers dynamic and automated security management that is adaptable to diverse needs and changing risks and threats.
In the following sub-sections, we first describe the security challenges, and then present details of the security architecture.

3.1 Security challenges in E2E deterministic networks

First, E2E deterministic communications are envisioned to support ultra-low-latency communication for critical applications such as industrial automation, which necessitates extremely precise and fast time synchronization mechanisms. This requires E2E security mechanisms based on high precision monitoring and fast reaction to quickly perform countermeasures. The network monitoring usually introduces additional latency in the communication. The additional latency should be short enough to avoid any compromise to timing guarantees of the communication.

Second, E2E deterministic networks rely on TSN or Deterministic Networking (DetNet) standards which are usually deployed in closed and controlled environments that avoid exposing them to external attack vectors. Furthermore, the 5G/6G systems rely on open and heterogenous networking technologies and verticals. Introducing these wireless systems, as proposed in the DETERMINISTIC6G

project, exposes the deterministic applications to external attack vectors (e.g., jamming, device impersonation, compromised devices, fake gNodeBs) that can impact time synchronization, time determinism, packet delivery and consistency.

Finally, the security mechanisms proposed in DetNet or TSN, such as zones, conduits and per-stream filtering and policing (PSFP), only address local security requirements, not E2E security requirements as targeted by DETERMINISTIC6G.

We address these challenges via a security by design approach, which consists of ZSM to automate security E2E management.

3.2 State of the art

In this section we present a the state-of-the-art of security for 5G/6G focusing on deterministic networking security tools and techniques.

As explained in DETERMINISTIC6G project's published paper [SPS+23], numerous research endeavours, such as INSPIRE-5Gplus [Inspire-5Gplus] and HEXA-X [HexaX], have placed a strong focus on security within the context of 5G/6G networks. However, there is a noticeable gap in these projects regarding a comprehensive E2E perspective for deterministic networking. There is also a limited number of studies aimed at creating security measures specifically for TSN and DetNet standards.

3.2.1 Publications

Recent published work dealing with specific topics includes:

- [MHK+19] employing the Credit-Based Shaping (CBS) algorithm, defined in IEEE 802.1Qav [IEEE-8021Qav], to defend TSN-equipped vehicular systems against Denial of Service (DoS) attacks by permitting only legitimate traffic patterns, as verified through E2E latency and frame counts analysis;
- [TCL+23] evaluated the potential of 802.1Qci's [IEEE-8021Qci] Per-Stream Filtering and Policing (PSFP) mechanism that consist of three instance tables, Stream Filters, Stream Gates, and Flow Meters, to change the priority of flows based on a time schedule and adopts several stream gates to enable deadline-based frame priority;
- [AS20] studied IEEE 1588 PTP [IEEE-1588PTP] vulnerabilities and proposed a monitoring function that compares clock offsets/delay measurements provided by numerous secondary devices.[BL21] Focuses on Time-Aware Shaper (TAS) that utilizes Gate Control Lists (GCLs). GCLs contain information about gate statuses to control the transmission of frames of different priorities. To mitigate the occurrence of misbehaviours (not necessarily due to cyber-attacks), the authors use reinforcement learning, Deep Deterministic Policy Gradient (DDPG), to model the uncertainty caused by the transmission-influencing factors such as time-synchronization errors, and then modify the GCLs used in the network to find more suitable queuing of output ports.
- [ZC23] focuses on the detection and mitigation of scheduling violations in each port of TSN switches due to time-synchronisation errors, and frame transmission and scheduling

dynamics. The detection is done by mapping the Time Triggered traffic frames with their scheduled windows stored in the memory of the switches. The mitigation is done by dynamically and periodically reconfiguring the schedule at each TSN switch.

- [MCP+23] proposed rethinking the security design bottom up, starting at the physical layer to overcome security hurdles in massive Machine-Type Communications (mMTC), Ultra Reliable Low-Latency Communications (URLLC) and autonomous cyber-physical systems. The authors identify the need to align physical layer security metrics to semantic security metrics and defining the security levels based on, for instance, the criticality of information and the performance constraints. A key advantage is seen for developing light-weight security solutions for low-latency and massive Internet of Things (IoT) use cases.

For very recent published works presenting monitoring solutions for TSN, we have:

- [ESF23] extends a monitoring tool to be able to parse TSN protocols and detect certain security breaches. The authors focus on the threats against two TSN control layer protocols, IEEE 802.1CB Frame Replication and Elimination for Reliability (FRER) and IEEE 802.1Qcc Stream Reservation Protocol (SRP). They implement rules to detect attacks, such as: Excessive resource request, Deviating resource request, Too many requests, Changing existing allocation, Dangling resources, Excessive member streams, Terminated member streams, Unusual SRP request, Flooding SRP requests, Forging fake sequence numbers, Malicious rerouting, and Triggering timeout. These rules will also be implemented in the MMT monitoring framework that will also be able to detect anomalies in the time synchronisation and data latency.
- [BY+19] proposes a TSN network monitor for the analysis and display of changes in the time synchronization accuracy of the ePTP protocol. It uses INT to detect clock deviation offset between the MASTER and SLAVE clock nodes. A similar technique is implemented in the MMT monitoring framework but is more adaptable to different time synchronisation mechanisms and use cases, providing reaction mechanisms, and further enabling the detection of data latency anomalies.
- [CB+22] proposes an architecture design and data collection mechanism that enable timely identification and collection of packet-forwarding performance-related misbehaviour in TSN. The authors use probe packets to collect misbehaviour information. They mirror the traffic to avoid disrupting the Time Triggered flows. Each switch needs to be equipped with an identifier for storing misbehaviour data. The probe packets collect the stored misbehaviour using INT. The updating of GCLs to mitigate the misbehaviours is planned as future work.

Despite these and similar focused efforts, there is a distinct absence of a holistic security-by-design framework that fully addresses 1) both the data and control planes; 2) the multi-layered, adaptive, and differentiated security demands of next-generation deterministic networks, accommodating diverse domains, service providers, and stakeholder requirements; and 3) the adaptability and flexibility offered by P4 data plane programming.

Requirements from different application domains can be found in deliverable D1.1: DETERMINISTIC6G use cases and architecture principles [DET23-D1.1]. Concerning security, the requirements include:

- **Flexibility and adaptability**: Flexible security functions in order to adapt to the different requirements coming from each use case and proposing different levels of security to achieve an optimal balance between the criticality of the application and its performance.

  The confidentiality, integrity, and availability (CIA) needs to be assured but depends on the needs of each use case scenario.

  Security requirement can change depending on the state of the applications and environment (e.g., emergency situations, degraded operation mode).

  Dedicated security functions applied to safety in industrial processes (e.g., virtual emergency switches, timely availability and prioritisation of crucial information in devices).

- **Cyber-physical security**: Security functions that consider cyber-physical systems and cascading effects between different domains that could negatively impact the safety of assets and humans.

- **Security of more open systems**: Securing more open systems that can be accessed from the exterior (e.g., through web applications), involving multi-providers, use of private or public cloud services, involving non-managed devices and applications, integrating open-source software, and using wireless communications (e.g., IoT networks, 5G/6G networks).

  DoS and Slow DoS attacks need to be blocked before they overly impact the services.

  The use of wireless communications can be vulnerable to jamming attacks, fake devices and base stations.

- **Ensuring time synchronisation**: Securing time synchronisation where the level of tolerated deviations depends on the use case. In general, the synchronisation protocol exchanges do not need to be encrypted but the integrity and the authenticity of the packets needs to be assured.

- **Interoperable security**: Secure interoperation in multi-provider and multi-tenant environments. This includes the possibility of negotiating the security properties and levels (e.g., Security Service Level Agreements).

  Interoperability of security requirements in factories that involve different stakeholders and considering the impact of Cyber Physical Systems (CPS) on communication, computation, and security and vice versa.

- **Accountability**: RCA and forensics could be necessary in case of accidents.

- **Encryption**: Encryption of data and control plane exchanges is needed in most applications to prevent unauthorized access. Sensitive data needs to be identified, encrypted and stored securely to prevent unauthorized access and disclosure.

  In applications that depend on continued flow of data, such as vehicle positioning and obstacle detection, encryption is not always possible and can allow attackers to disrupt them by injecting/replaying packets and requests.

- **Data privacy**: Collection and transmission of sensitive data about the user, such as their biometric data, medical history, or personal information, must be encrypted and stored securely to prevent unauthorized access and disclosure.

- **Redundancy**: In case of a network failure or other disruptions, some use cases should have redundant control mechanisms in place to ensure that the application can still operate safely and that the service continuity is assured.

- **Security monitoring**: Depending on the use case, packet loss, jitter, synchronization accuracy, and communications may need to be monitored in a more precise or less precise way to detect any deviations from expected behaviour.

  Network, application and/or system behaviour analysis and anomaly detection need to consider the requirements of deterministic communications.

- **Automation of security management**: The complete automation of the security management may be necessary, but, in some cases, human interventions (i.e., security management with user-in-the-loop) might be necessary or preferred to identify the nature of a security breach and determine how to deal with it.
- **Mitigation, countermeasures and prevention**: Response to attacks should be timely to prevent critical loss of services. Strategies need to be adapted to the needs of the use cases and might require applying Moving Target Defence (MTD) techniques, logical or virtual separation of shared communication resources (e.g., network slicing), etc.

3.2.2 Security and vulnerabilities

Deterministic networks are designed to provide predictable and guaranteed data transmission characteristics, which are critical for applications requiring high reliability and low latency, such as autonomous vehicles, industrial automation, and remote surgery. These deterministic characteristics also introduce specific security vulnerabilities that include:

- Predictable traffic patterns that attackers can easily disrupt, using, e.g. DoS attacks;
- If centralized network control is necessary for managing timing, resources, and orchestration, they become single points of failure;
- Time synchronization (e.g., using IEEE 1588 PTP) can be disrupted also using DoS attacks;
- If resource reservation protocols are used, these can be disrupted by either blocking legitimate requests or making illegitimate ones;
- Other vulnerabilities that could impact determinism are, for instance: coordination across different layers can provoke cascading effects, limited redundancy to optimize efficiency can impact system resiliency, supply chain risks and insider threats can introduce unexpected vulnerabilities, IoT device vulnerabilities due to limited security controls, and interoperation with legacy systems can introduce security gaps and inconsistencies, and quantum computing could break traditional cryptographic protocols.

Addressing these vulnerabilities requires a multi-faceted approach that includes advanced encryption methods, anomaly detection systems, robust authentication mechanisms, and continual monitoring and assessment to adapt to new threats. There is also a significant push towards developing AI-driven security solutions that can predict and mitigate attacks in real-time, and post-quantum cryptography to safeguard against quantum threats.

Vulnerabilities on time synchronisation and distribution of timing information are described in more detail in Sections 4.4 and 5.4 of the deliverable D2.2: First report on time synchronization for E2E time awareness [DET23-D2.2]

3.2.3 Standards

The key standardization efforts related to deterministic and time-sensitive networking include:

- IEEE 802.1AE (MACsec) [IEEE-8021AE]: specifies the Media Access Control security (MACsec) protocol for Layer 2 data confidentiality, data integrity, and data origin authenticity. MACsec secures all traffic on an Ethernet link, including TSN streams to avoid tampering and spoofing of Ethernet frames;
- IEEE 802.1AR (Secure Device Identity) [IEEE-8021AR]: defines unique device identifiers (DevIDs) for Ethernet devices. It is used to authenticate a device and establish a secure communication channel;
- IEEE 802.1X [IEEE-8021X]: provides port-based Network Access Control (PNAC), an authentication mechanism to ensure that only authenticated devices can introduce streams or access specific network resources;
- IEEE 802.1Qci [IEEE-8021Qci]: is part of the IEEE 802.1Q standard and deals with Per Stream Filtering and Policing. It ensures that each stream is restricted to its reserved resources and doesn't exceed them. This can act as a security measure against misbehaving or malicious devices trying to disrupt the network;
- IEEE 802.1AS [IEEE-8021ASrev] (Timing and Synchronization for Time-Sensitive Applications): defines the time synchronization required in TSN. Ensuring the integrity and authenticity of time synchronization messages is a security requirement clearly addressed by the standard;
- IETF DetNet Security [IETF-RTC9055]: specifies the security considerations for deterministic networks. Various threats are considered, such as eavesdropping, denial of service, and spoofing, and potential mitigation techniques are proposed;
- 3GPP supports TSN Time Synchronization, treating the entire end-to-end 5G system as an IEEE 802.1AS "time-aware system". This ensures synchronization of various network elements like User Equipment (UE), Next Generation Node B (gNB), User Plane Function (UPF), and TSN Translators (TTs);
- 3GPP enhancements in Release-16/17 include the integration with IEEE TSN with support for uplink synchronization via 5GS, multiple working clock domains connected to the UE, and time synchronization of UEs with the TSN GM. It also aims to enhance support for deterministic applications beyond IEEE TSN, exposing network capabilities to support time-sensitive communication services and optimizations;
    - 3GPP TS 23.501, System Architecture for 5G System; Stage 2 (clauses 4.4.8, 5.27, 5.28, Annex H, Annex I on support for TSN and clauses 5.6.10.2, 5.7.6.3, 5.8.2.5.3 on Ethernet forwarding);
    - 3GPP TS 23.502, Procedures for 5G System; Stage 2 (Annex F on support for TSN);
    - 3GPP TS 23.503, Policy and Charging Control Framework for the 5G System; Stage 2 (clause 6.1.3.23 on support for TSN);
    - 3GPP WID: 830042 (Vertical_LAN) 5GS Enhanced support of Vertical and LAN Services;
    - 3GPP Liaison Statement (S2-2003508) to IEEE on specification maturity for IEEE TSN integration work in 3GPP Release-16, with further enhancements expected in Release-17.

Besides these standards focusing on deterministic networking, more general network security techniques and guidelines need to be considered to ensure mission-critical applications, such as industrial automation or automotive networks. Security-by-design and layered defence-in-depth approaches are needed.

In the work undertaken as part of the DETERMINISTIC6G project, evaluation of existing standards is important for shaping the definitive security architecture. The priority of the monitoring framework developed here is to establish a robust system that could serve as the cornerstone for cyberattack detection, mitigation, and prevention, specifically targeting threats that could undermine the predictable behaviour of deterministic networks. The approach taken does not encompass the actual implementation of security protocols such as for the encryption, integrity of communications, and authentication of devices. Instead, it focuses on the ability to recognize anomalies in the usage patterns of these protocols. This approach is chosen to complement the broader security measures, ensuring that any irregular activity, which could signal a potential security breach, is identified promptly and addressed effectively.

The framework thus places significant emphasis on the surveillance of protocol behaviour, particularly scrutinizing communication patterns for deviations from expected norms that could indicate attacks, enhancing the surveillance capabilities of the system. This allows the framework to detect when the established deterministic network protocols, that maintain stringent timing and delivery guarantees, are operating in such a way as to ensure the predictability and confidentiality of data transmissions, where even minimal disruptions can have critical repercussions.

3.3 Monitoring techniques

In this section we present different techniques that can be used for enhanced security-by-design. 5G/6G security monitoring and management rely on these innovative technologies to ensure real-time vigilance and dynamic response to threats. Here, we are using them in an innovative way to solve the challenges of security of deterministic networking. INT integrates monitoring within the data flow, minimizing overhead. P4 programmability allows network devices to be tailored for specific security functions, enhancing flexibility and response to threats. High-precision telemetry delivers detailed, timely data for monitoring, while security analysis processes data to identify and react to anomalies. AI/ML-based security analysis, including Change Point Detection (CPD) and Root Cause Analysis (RCA) will be addressed in deliverable D3.5: Multi-domain end-to-end schedules [DET25-D3.5]. Together, these techniques form an adaptive security infrastructure capable of protecting future 6G networks against sophisticated cyber threats.

3.3.1 High-precision network monitoring techniques

Network monitoring: Active vs Passive

Active network monitoring (depicted in the left side of Figure 1) deduces network performance by injecting, and then observing and analysing the operational status of the probe packets. This proactive approach can detect potential performance problems before they happen in the network. It has no privacy issues since the generated probe packets are artificial. However, these probe packets may increase network traffic, thus disturbing the normal traffic flow, and they may not be operated exactly

as the real traffic to be measured. Consequently, the measurement results do not always accurately reflect the actual network performance.

Passive network monitoring (depicted in the right side of Figure 1) overcomes the proactive approach's limitation by analysing the real network packets via proxy reporting, from switches, or traffic mirroring, via packet capturing or sniffing. As it works on ground-truth [NNJ+20], it can provide information to detect any anomalies in the network at the working moment, including performance and security issues. However, this approach faces privacy issues when analysing production traffic. Another challenge is in the scalability of the approach when processing huge amounts of aggregated traffic. This challenge is usually mitigated by sampling, such as, update cache statistics for only 1 per 128 packets, or aggregating, i.e., cumulating several packets in 1 second, for example. This mitigation leads to loss of precision as it can miss micro information, such as, small flows, and glitches or irregularities of traffic which occur during a micro interval of time. Furthermore, large sampling intervals are not suitable for real-time reactions.

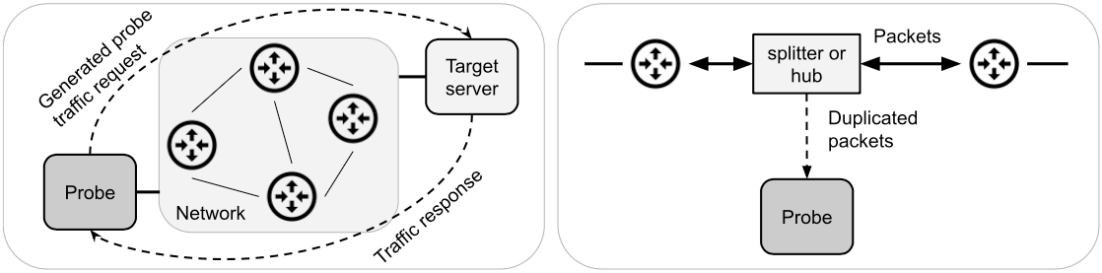

Figure 1. Active (left) vs. Passive (right) network measurements

High-precision monitoring requirements

A network monitoring framework is usually provided together with a reaction framework to get a complete closed-loop solution. The monitoring function needs to provide fine-grained (i.e., high precision per packet inspection) information at device level (e.g., queue length, queue delay, link utilisation), packet level (e.g., Internet Protocol, IP, and port source/destination, ...), and flow level (e.g., contributing flow whose packets occupy at least a small fraction of the queue). The information concerning the network must be precise enough so that users/controllers are able to react to ensure that the network natively supports critical services. It must also provide the full-coverage statistics to avoid missing microbursts that are caused by a lot of data bursts in a very short time, usually at millisecond level. Consequently, it also needs to provide high-resolution measurements at a smaller timescale to enable faster control loops for countermeasure.

The high-precision monitoring implemented in this architecture can operate in either an active or passive mode. This translates to in-band or out-of-band analysis of network packets, respectively. In the in-band, or active mode, any mitigations or countermeasures applied will have a direct influence on the network traffic, functioning akin to an Intrusion Prevention System (IPS). This approach allows for immediate response to threats. Conversely, in the out-of-band, or passive mode, the response mechanisms are more indirect, operating similarly to an Intrusion Detection System (IDS). Here, the system primarily generates alerts based on detected anomalies, which then require subsequent

management and intervention. The high precision monitoring certainly causes overhead in the networked system. The overhead must not be mitigated via sampling, aggregating, nor coarse-grained counter, but for example via on-demand monitoring. The on-demand monitoring is suitable for high scalability because it gives the ability of customisable statistics and reports. It measures only the required metrics and only when being requested.

P4 - A Programming Protocol-independent Packet Processors language

Programming Protocol-independent Packet Processors (P4) is an open-source, domain-specific programming language for network devices, specifying how networking devices (switches, routers, NICs, filters, etc.) process packets. It has recently emerged and is potentially becoming a disruptive instrument [FFA+19] enabling the programming and customizing of the data plane as well as the control plane of next-generation networks. Before P4, Software Defined Networking (SDN) API's like OpenFlow would only allow to program the control plane. P4 introduces the ability to program the network devices, thus, it significantly reduces the need to have dedicated hardware devices by introducing virtualized devices and APIs to process and control network traffic.

The architecture of a P4 switch is depicted in Figure 2. Here, an incoming packet is received and parsed. This initiates, as a preamble, an immediate match-action rule based on the data readily available in the packet's frame. Subsequently, a buffer is established, allowing for potential modifications to the frame as specified by the user. Following any adjustments, a postamble match-action rule is systematically applied to the frame. Once these steps have taken place, an output interface is assigned to the frame and an egress action of the frame is performed.

A match-action rule refers to the core principle of how P4 processes network packets and is composed of two parts:

- **Match**: This part of the rule involves checking the packet's header fields against a set of predefined criteria. If the packet's header contains particular bit patterns that match these criteria, the rule is considered a match. The matching criteria can be very simple, like checking for a specific IP address, or more complex, involving multiple fields and conditions;
- **Action**: If the packet matches the rule, an action is taken. Actions in P4 can range from modifying the packet's headers, copying the packet, dropping the packet, forwarding it to a particular port, or any other defined operations that can be applied to the packet within the switch.

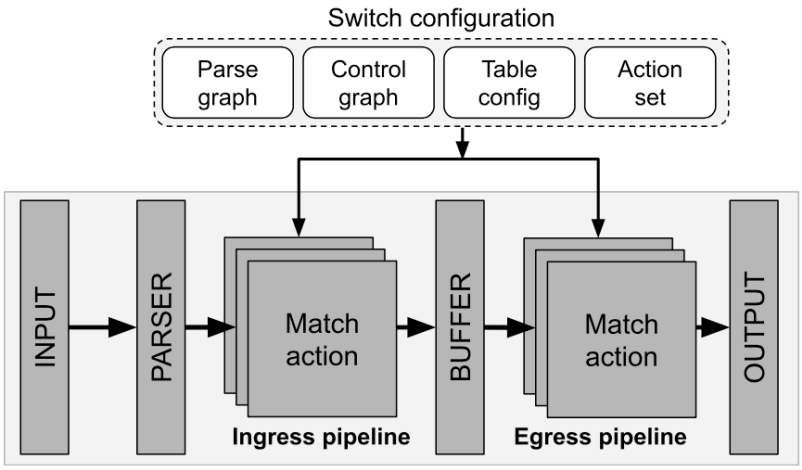

Figure 2 – P4 pipelines

The simplest way to model each block of this pipeline, particularly the match-action blocks, is by a state diagram. For example, if we need to ensure that the "EtherType" of the frame corresponds to the VLAN TAG (0x8100) then the following state diagram is placed in the first match-action block of the pipeline as in Figure 3. Similarly, since TSN traffic is at Open Systems Interconnection (OSI) layer 2, we only need to parse the Ethernet header and then check whether the Virtual Local Area Network (VLAN) tag is present. If it is not the case, then we can drop the frame.

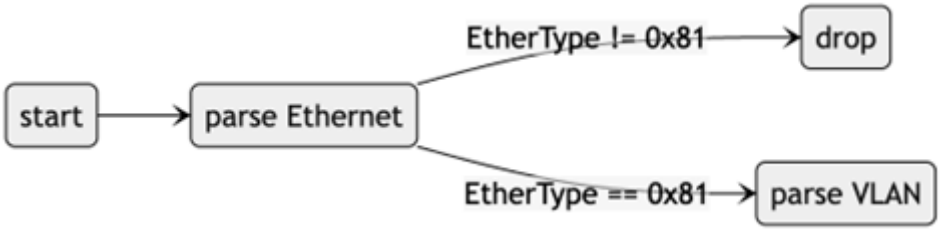

Figure 3 – State machine to parse VLAN packets

In-band Network Telemetry (INT)

Network telemetry has emerged as a mainstream technical term [LWZ+21] to refer to an automated process for remotely collecting, processing and reporting network state information by the data plane, without requiring intervention or work by the control plane. INT combines data packet forwarding with network measurement. It uses packets in the data plane to carry telemetry instructions and collected metric results. Currently, the INT research is led by two working groups, the Internet Engineering Task Force (IETF) IP Performance Measurement (IPPM) and the P4 Application (p4.org). IETF conducts research and standardization on the architecture and protocol of INT. It promotes in-situ Operation Administration and Maintenance (OAM) which complements current out-of-band OAM mechanisms, based on ICMP or other types of probe packet, to compose the set of tools used for OAM. The p4.org focuses on the implementation of INT using the programable data-plane and proposes basic implementation ideas using the P4 language to leverage network softwarization techniques.

P4.org defines the INT data plane specification, including the INT system, the telemetry metadata, INT encapsulation and implementation examples. An INT system consists of programmable switches, an INT collector that receives and extracts the INT data generated by the switches, and eventually a network controller that configures the switches.

An INT-capable switch plays one of three roles: source, transit or sink. The source node decides which packets carry the data, then embeds a telemetry instruction bit map into the packets to indicate the network information to be measured. While matching and forwarding a packet, the transit node interprets the bit map to attach required information, for example the switch id, hop latency, etc. The sink node reports all collected telemetry data to an INT collector and removes the telemetry data from the packet. A node will not attach its information into a packet if the resulting packet size is greater than its Maximum Transmission Unit (MTU).

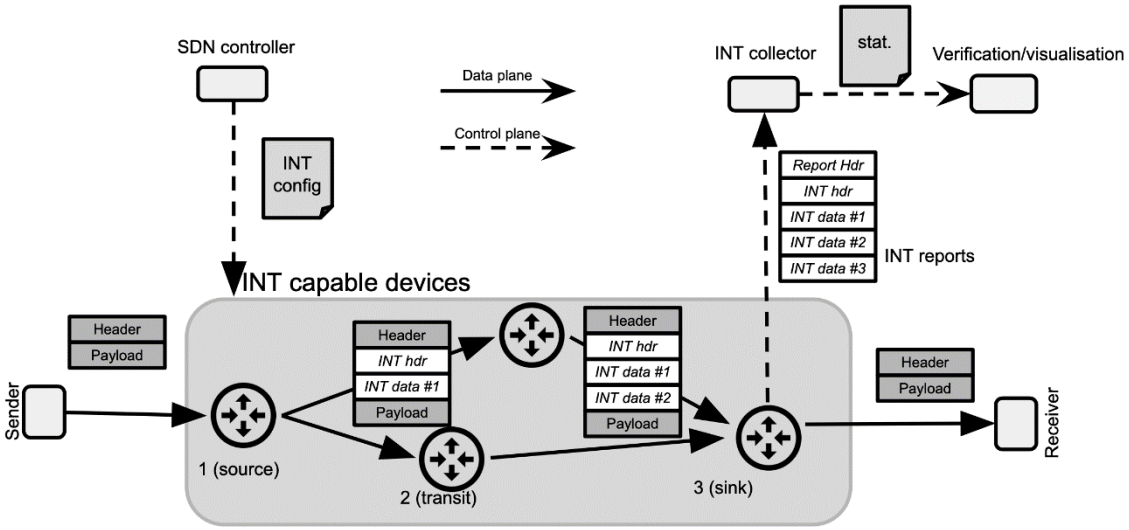

Figure 4. Example of an INT system in INT-MD mode

The P4 Working Group defines three INT modes of operation. In the INT-XD (eXport Data) mode, an INT-capable device plays all three roles, i.e., it selects INT packets, collects and then sends telemetry data to the collector. In the INT-MX (eMbed instructXions) and INT-MD (eMbed Data, shown in Figure 4) modes, distinct devices assume the roles of source and sink. Conversely, additional devices function as transit nodes. While all devices undertake the transit role to gather data, there is a key distinction: in INT-MX mode, the source and sink devices directly export the aggregated information to the collector. In contrast, in INT-MD mode, transit devices embed the collected data into packets, which are then relayed to the subsequent node in the network path.

INT is originally applied for network performance measurement to provide real-time network status, such as packet delay, packet loss, available bandwidth, QoS measurement, etc. It has been extended to fault location or RCA for network troubleshooting because it can provide answers for the questions: "how did this packet get here?" via the sequence of network devices a packet visited along its path, "why is this packet here?" via the set of rules of packet matched upon at every network device along the way, "how long is this packet delayed?" via the series of time the packet spent buffered in every switch, "why was the packet delayed?" via the flows the packet shared each queue with. It can usefully detect microburst occurring at millisecond level, which is difficult for SNMP (Simple Network Management Protocol) or NetFlow-based tools due to their technical limitation of second-by-second traffic monitoring cycles.

However, INT has some disadvantages. It reduces the payload ratio of the packets due to the encapsulation of telemetry instruction and metadata. The processing of INT such as, encapsulation, filling and extraction of telemetry instruction and metadata, introduces overhead on switches which may lead to drop packets.

3.3.2 Signature-based network security analysis techniques

Signature-based network security analysis techniques try to detect security related issues on a monitored network by observing network packet events and identifying patterns that match the signatures of known threats. These signatures are often created based on characteristics of known attacks, such as network traffic patterns, specific protocol attributes, or sequences of packets.

Signature-based detection is efficient at identifying known threats and is fast and accurate when dealing with recognized attack patterns. It has been a fundamental component of security systems for many years. However, it also has limitations. It cannot detect zero-day attacks or previously unknown threats. The signature databases need frequent updates to stay relevant and effective. New threats emerge continuously, so the database must be kept up to date to detect the latest malware and attack techniques. Sophisticated attackers can also evade signature-based detection by modifying or obfuscating their attacks. To address these limitations, a combination of security techniques, including signature-based detection and machine learning, is commonly used to detect a broader range of threats, including those that don't have known signatures.

3.4 Security architecture and enablers

A security-by-design 6G architecture should integrate comprehensive E2E analysis and RCA to proactively identify and rectify vulnerabilities and threats across the network. At its core, it leverages AI/ML-based analysis for robust detection and diagnosis of security incidents, streamlining the response, which includes reaction, mitigation, and prevention strategies. The architecture is equipped with a specialized process pipeline that is engineered to collect and generate network traffic data. This data is vital for both the training and the validation phases of artificial intelligence and machine learning models. It utilizes sophisticated data extraction methods that feed selected and pre-processed features into these models, enabling them to accurately discern security and performance anomalies, especially those affecting applications that depend on low-latency or deterministic networking.

Crucial to 6G deterministic operation is INT, which facilitates the gathering of network information without imposing additional traffic, and a P4-based programmable data plane that allows for bespoke and adaptable traffic handling and security policy enforcement. The system should employ high-precision telemetry techniques, which ensure that network performance data are captured with the utmost accuracy, essential for the low-latency and deterministic demands of future 6G applications. This synergy of advanced network programmability, real-time telemetry, and AI-driven analytics constitutes a resilient and adaptive architecture geared for deterministic 6G applications.

3.4.1 High-level architecture

In this section, we will illustrate an example of a security and enforcement service deployment architecture that encompasses High-Level Architecture (HLA) functional blocks and services. This

architecture is based on the initial architecture [JRJ+20] [Inspire-5Gplus whitepaper] developed in the INSPIRE-5Gplus project [Inspire-5Gplus]. The architecture not only enhances security but also instils trustworthiness and accountability in the management of 5G/6G network infrastructures spanning multiple domains. In DETRMINISTIC6G, the architecture is currently undergoing an extension to address the needs of deterministic networking more comprehensively. This enhancement involves the integration of high-precision monitoring methods along with the incorporation of both TSN and DetNet domains. This integration aims to provide a robust and efficient framework, ensuring that the network meets the stringent demands of precision, reliability, and consistency essential for deterministic networking environments. It will be described in detail in the deliverable D1.4: Final report on DETERMINISTIC6G architecture [DET25-D1.4]. In the following we provide a first description.

The assurance framework can be divided into several Security Management Domains (SMDs) to ensure resilience and to segregate security management concerns, such as those for the Radio Access Network (RAN), Edge, and Core Network.

Each SMD is tasked with intelligent (i.e., AI-based) security automation within its designated scope and encompasses a suite of functional modules, including a Security Data Collector, Security Analytics Engine, Decision Engine, Security Orchestration, and others. These modules provide a variety of security management services, accessible both within their respective domains and across domains via an integration fabric.

A specialized SMD, referred to as the E2E SMD, is specifically designed to oversee the security of E2E services, such as E2E network slices that extend across multiple domains. By decoupling E2E security management from other domains, this approach avoids monolithic systems, reduces the overall system's complexity, and facilitates the independent evolution of security management at both domain-specific and cross-domain levels.

These functional modules operate intelligently in a closed-loop manner (i.e., collection-detection-reaction as depicted in the steps 1 to 5 of Figure 5), enabling AI-driven software-defined security (SD-SEC) orchestration and management in accordance with expected Security Service Level Agreements (SSLA) and regulatory requirements. By embracing service-based and SD-SEC models, this framework establishes adaptable security measures capable of responding to dynamic changes in the threat landscape and evolving security requirements in next-generation mobile networks.

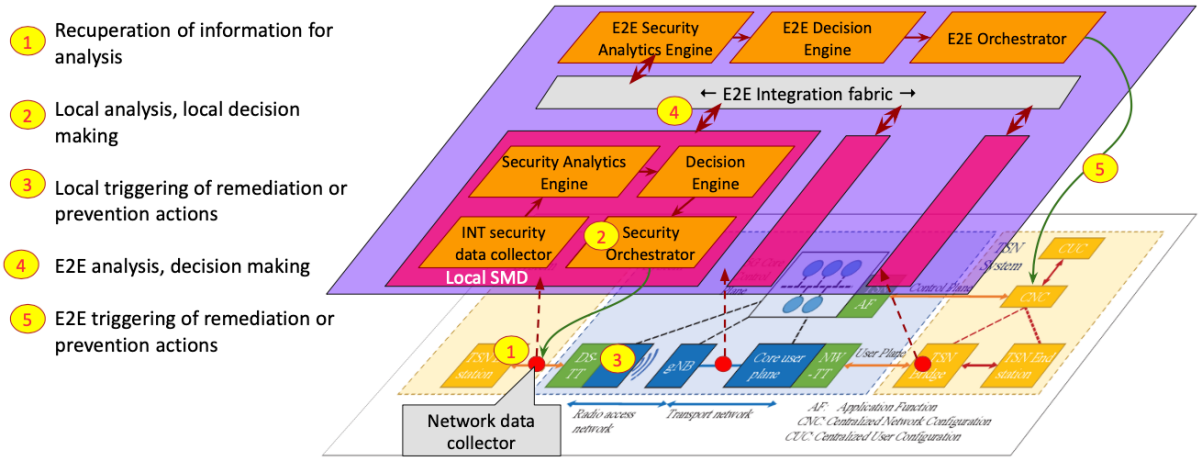

Figure 5: High level architecture of E2E security monitoring & management framework

3.4.2 Security Data Collectors

The primary role of the Security Data Collector (SDC) (step 1 of Figure 5) is to aggregate all data originating from security enablers at the domain level, which is essential for the functioning of security management processes, such as the Security Analytics Engine. The SDC relies on network, system, and application probes that capture and pre-process event data. The types of data that the SDC gathers encompass:

- Performance monitoring data (e.g., counters and statics data);
- Security monitoring data (e.g., traffic meta-data, packet capture, session data);
- Event/alarm data (e.g., system logs, application traces, system traces);
- Machine learning reference datasets for learning and prediction phases;
- External data (e.g., Cyber Threat Intelligence, external data sets).

We will present in, the following, detail our implementation of two types of high-precision security data collectors that are suitable for time sensitive networks.

3.4.3 Security Analytics Engine

The main function of the Security Analytics Engine (SAE) is to derive insights and predictions on a domain's security conditions based on data collected in that specific domain or even from other domains. In the context of the DETERMINISTIC6G project, the SAE can provide Anomaly Detection and RCA services. The Anomaly Detection service has the capabilities of detecting and/or predicting anomalous behaviour due to malicious or accidental actions by identifying patterns in data or behaviour that do not conform to the expected normal behaviour. It leverages data aggregated by the SDC from the managed entities of the domain, including performance and security monitoring data, events and alarms, generated by system logs and packet traces. The RCA service identifies the cause of the observed security incidents by analysing and correlating data from other services (e.g., Anomaly Detection service). The RCA determines the origin of the anomaly and the location in the network where a corrective action should be applied to prevent the problem from occurring. For this, the probes that capture the data used by the RCA need to be deployed in different observation points strategically defined to facilitate the localisation of the anomalies. As a result, the RCA service may provide the information needed by other security functions to determine the actions that should be triggered to correct or prevent the security incidents in the 5G/6G networking environment.

The techniques for the detection of anomalies and root causes include ML/AI, feature extraction, Complex Event Processing, Deep Packet Inspection, Change Point Analysis and more. These are, for instance, the detection of Distributed DoS (DDoS), the analysis of encrypted network traffic, the detection of misbehaviour, anti-GPS spoofing, assessment of encrypted channel protection, security SSLA assessment, and RCA in Industrial Campuses. The SAE is an essential component providing the information needed by the Decision Engine, Orchestrator, Moving Target Defence, etc. for the prevention, mitigation, and reaction to cyber-attacks.

3.4.4 Decision Engine

The Decision Engine (DE) functional block assumes the role of overseeing the diverse actions generated by security assets and the SAE. Its primary function is to determine the most suitable decisions for securing a targeted service in operation. Serving as a central component, the DE functions as an arbitrator, mediating between the security assets and the broader platform responsible for domain management.

The DE entrusts the actual creation of mitigation actions to two distinct categories of assets: Cognitive Long-Term and Reactive Short-Term assets. These assets are equipped with algorithms for devising a cohesive mitigation strategy in response to identified threats:

- The Cognitive Long-Term assets will be based on advanced AI techniques and may use historical data from several sources to internally deduce correlations, potential forecasts and propose elaborated mitigation plans to the DE;
- The Reactive Short-Term assets will rely on simple rules to provide quick and mundane reactions to specific events. These rules will be akin to what a human operator would do in the given situation. Due to their simple and streamlined structure, the mitigations resulting from these assets can be rapidly computed and enacted.

The DE may depend on multiple "third-party" assets operating concurrently, awaiting their respective mitigation decisions. These decisions may rely on the DE without adhering to any prescribed sequence and may even conflict with one another. For example, a Reactive Short-Term asset might deem a device as legitimate, authorizing its traffic. Subsequently, a Cognitive Long-Term asset may identify the same device as a potential source of DDoS attacks. In such instances, the DE must mediate the conflicting responses, possibly by considering a confidence level and/or referring to a priority list established through statistical analysis. Furthermore, as the implementation of mitigation measures may take time through the underlying Security Orchestrator (SO), the DE must keep track of selected responses and may disregard newly received mitigation decisions to stabilize the protected system.

Specifically, the DE includes the necessary parsers to parse the alerts and identify the impacted service(s). Subsequently, it generates a new security policy using the Management of Security Policies Language (MSPL), tailored to the type of alert raised. This MSPL is adapted to interface with the underlying SO API. Finally, the DE transmits the newly generated policy to the Security Management Domain's (SMD) security orchestrator. While this initial reactive loop operates within an SMD, the local DE also escalates the alerts to the E2E level for additional reactions.

3.4.5 Security Orchestrator

The SO assumes the role of supervising various security enablers to enforce the security requirements outlined in the adopted security policies. The SO actively guides security management by engaging, via the integration fabric, with different elements such as SDN controllers, NFV MANO, and security management services.

The SO takes a proactive or reactive approach to implement security policies by allocating, chaining, and configuring virtual network security functions (VSF). These VSFs include components like virtual Intrusion Detection Systems (vIDS), virtual Firewalls, and virtual Authentication, Authorization, and Accounting (vAAA).

The SO receives input from an evolving system model, derived from structural information provided by network administrators, insights from monitoring systems that observe network deployments for changes, trust and reputation data from the Trust Management (TM) component, and intelligence and plans generated by the DE. This cognitive behaviour equips the SO with self-healing and self-protection capabilities for the entire managed system. It enables the orchestrator to automatically respond to the current context and promptly trigger appropriate countermeasures to mitigate ongoing attacks or proactively address anticipated threats.

Potential responses may include applying security policies to manage network traffic (e.g., dropping or redirecting it) through an SDN controller, as well as deploying, decommissioning, reconfiguring, or migrating the VSFs as needed.

3.4.6 Security scenarios

The following subsection present two scenarios concerning the disruption of time synchronization and latency.

3.4.6.1 Scenario 1: DoS Attack on PTP in Time-Sensitive Networking

Attack Generation: A manufacturing plant has recently upgraded its network to incorporate TSN to ensure precise timing and synchronization for its automation systems. One morning, the network experiences an unexpected disruption. An attacker has initiated a DoS attack by flooding the network with a surge of Precision Time Protocol (PTP) requests. These requests are designed to overwhelm the grandmaster clock, which is critical in maintaining time synchronization across all devices in the network.

Attack Detection: The network's security system, equipped with an AI/ML-based monitoring solution, quickly detects an anomaly in the traffic patterns. The system notices an abnormal variation in PTP message intervals, which is unusual for the typical network operation. By leveraging high-precision telemetry data, the system identifies that the timing accuracy across the network devices is deviating beyond acceptable thresholds – a clear indicator of an ongoing attack on the network's time synchronization mechanism.

Initial Mitigation Efforts: Upon detection, an automated script is triggered to temporarily rate-limit the incoming PTP traffic. This script is a measure to prevent the grandmaster clock from being completely overwhelmed. It allows legitimate PTP requests from known devices but limits others, (i.e., known devices that behave incorrectly or unknown devices) based on predefined thresholds of normal operation.

Further Analysis and Mitigation: Meanwhile, the AI/ML system begins a more in-depth analysis to distinguish between legitimate traffic and malicious packets. Using machine learning models trained

on normal PTP request patterns, the system starts to filter out the attack traffic by recognizing the signatures (i.e., unusual traffic patterns) that differ from the standard behaviour.

The PTP configuration is also adjusted to implement security measures like source address validation and message type limits. These configurations ensure that only trusted devices can send PTP messages and restricts the type and rate of PTP messages accepted. If PTP messages can be authenticated, then a way to detect the attack would be to identify invalid messages with wrong signatures.

Network Reconfiguration and Long-term Mitigation: To bolster defences against future attacks, the network administrators deploy P4 programmable switches with customized data planes. These switches are programmed to identify and block malicious PTP packets in-band, preventing them from traversing the network. This not only reduces the load on the GM clock but also allows for real-time mitigation of similar attacks. The overloading of the Grand Master (GM), time Receiver (tR) and/or the time Transmitter (tT) depends on how the time synchronisation protocol works, e.g., requiring acknowledgements or using UDP broadcasts).

Further analysis of the event can lead to an update in the ML models, and security protocols are revised to improve the detection capabilities of the monitoring against potential DoS scenarios to better predict and prevent such occurrences.

3.4.6.2 Scenario 2: Latency Disruption Attack in Deterministic Networks

Attack Generation: The network, that is used for controlling robots, experiences intermittent disruptions. An attacker has launched an attack specifically designed to disrupt the deterministic latency guarantees. This is done by injecting high-priority traffic that pre-empts the normal scheduling of packets, leading to jitter and delays in the critical control data for the robotic arms.

Attack Detection: The deterministic network's monitoring system, which constantly checks the conformance to the specified latency bounds using, e.g., INT, quickly flags that the latency metrics are not within the expected deterministic thresholds. The system uses advanced statistical models to detect the irregular traffic patterns, considering the historical latency performance data.

Initial Mitigation Efforts: Upon detection, the network's automated response system instantaneously initiates an alternative network slice. It involves allocating resources to be able to reroute critical packets through a less congested path and temporarily suspending (e.g., dropping or delaying packets depending on the user requirements) certain non-critical network operations to stabilize the latency. It is also possible to send suspicious packets to a dedicated slice for further analysis.

Further Analysis and Mitigation: With the initial mitigation in place, the system employs AI/ML algorithms on each packet or session to perform RCA that analyses the traffic and detects the source of the high-priority disruptive packets. It applies ML to determine the probable cause and location of the attack.

Once the source of the malicious traffic is identified, the deterministic network's control plane dynamically updates the scheduling and prioritization policies to de-prioritize the packets (e.g., using Qci) identified as part of the attack. The security team reviews the incident to determine how to prevent these types of attack in the future, e.g., deploying a firewall, improving the ML models, that will make the system’s deterministic properties more robust.

4 Software Solution for Monitoring of Latency and Performance

In this section we detail our proposition of two high-precision monitoring techniques. They are suitable for deterministic networking systems because they introduce very low overhead on packet latency. Consequently, they do not disturb the deterministic behaviour of the system.

The first technique is non-disruptive monitoring, which is a passive monitoring solution. It does not introduce additional latency into network traffic. It is suitable for local monitoring, i.e., it can be easily deployed for capturing network packets at a given observation point and provide information on the current network status. The second technique is based on INT. This technique introduces some additional latency on the packets as we will see in the following section. However, it is a helpful tool for identifying security issues as it can provide historical information of a packet via the INT metadata added to the packet by switches/routers the packet passes through. We follow the guideline proposed by the P4 working group and use the P4 language to implement INT. The advantages of the P4 language include that it is open-source, and the implantation is independent of the network equipment vendor. We will present more details about this technique as it is proposed to be used and suitable for the DETERMINISTIC6G system.

4.1 Non-disruptive monitoring

MMT-Probe (Montimage Monitoring Tool) is a passive monitoring framework that combines data capture, filtering and storage, events extraction and statistics collection, traffic analysis and reporting, network, application, flow and user level visibility. Through its real-time and historical views, MMT facilitates network performance monitoring and operation troubleshooting. With its advanced rules engine, MMT can correlate network, system, and application events to detect performance, operational, and security incidents. An easy-to-use customizable graphical user interface makes MMT suitable for different user needs.

The framework is fully implemented by Montimage. It is released as an open-source solution under the Apache 2.0 license. It includes the following features:

- Real-time detection and prevention based on behaviour analysis of network, application, and system traces;
- Protocol analysis and Deep Packet and Flow Inspection at all OSI levels (2-7);
- Easy to define Security Service Level Agreements;
- Scalability and adaptability in different environments and for business activity monitoring;
- IoT wireless communication analysis;
- Highly configurable probes for lightweight and heavyweight data capturing and processing.

MMT-Probe is a passive monitoring solution as it works on a copy of network traffic. It captures network packets (e.g., using a network tap, SPAN mirroring port, Ethernet hub, or a wireless sniffer) to get a copy of them then analyses the packets through their copies. The original packets are not modified and continue their route to their destination. Consequently, MMT-Probe does not introduce any additional latency on the original packets, nor does it modify the packets.

MMT-Probe uses the deep packet inspection (DPI) technique to identify network protocol and application-based events by analysing protocols' field values, network, and application Quality of Service (QoS) parameters, and network key performance indicators (e.g., throughput, latency, jitter, time-to-respond). It provides statistics about the current network status to other modules for further analysis to determine whether the network respects the given SLA or not.

The solution can be deployed on standard (i.e., off-the-shelf servers) or dedicated (i.e., with custom hardware) computing systems. For instance, on a Raspberry Pi for low bandwidth traffic or multicore servers (bandwidths of 40 Gbps or more) using its parallelisation Data Plane Development Kit (DPDK) technique.

The implementation of MMT-Probe is freely available at https://github.com/DETERMINISTIC6G/deterministic6g_security-solutions/tree/main/mmt-probe

4.2 In-Band network monitoring and P4 programming

The P4 Working Group specifies eight sets of metrics that are used for monitoring general networking devices. We introduce additional metrics to monitor the performance of devices: the numbers of dropped packets and marked packets (i.e., setting the ECN packet field on). It is worth noting that we intend to keep the set of INT metrics that represent the internal information of a device as small as possible to reduce the length of the packets, hence reducing the bandwidth consumed by INT to transfer the metric values. We do not include the metrics that can be deduced outside the device by analysing the packets before or after it, e.g., bandwidth, throughput, or packet inter-arrival time.

Specifically, our framework monitors the following metrics:

- **Device ID** is the unique identification of the device;
- **Level 1 Ingress and Egress port IDs** are the IDs of the ports on which the packet was received and sent;
- **Hop latency** is the time, in microsecond, that it takes for the packet to be processed within the device;
- **Queue ID and queue occupancy** is the identification of a queue and the build-up of traffic in the queue, expressed as the number of packets when the packet was sent out;
- **Ingress timestamp** is the device's local time, in nanosecond, when the packet was received;
- **Egress timestamp** is the device's local time, in nanosecond, when the packet was sent out;
- **Level 2 ingress and egress port IDs** are the IDs of the logical ports, applicable for layer 3 switched virtual interface, on which the packet was received or sent;

- **Egress port TX link utilisation** is the current usage of the egress port the packet was sent through;
- **Numbers of ECN (Explicit Congestion Notification) marked packets and dropped packets** are the number of packets that have been marked and dropped by the switch, respectively.

As such, the above metrics can be used to identify, for example, the dominant contributing IP sender whose packets occupy most of a queue during a given interval. Indeed, by using the ingress and egress timestamps, one can get the set of packets that were present in the queue during the given interval. Furthermore, since the INT collector also can extract the IP source field of those packets, one can then easily identify their dominant IP source(s).

4.2.1 P4-based Monolithic Application

The P4 working group proposes different encapsulation of INT at different OSI layers as shown in Figure 6. We are implementing the encapsulation over TCP and UDP layers (the first two variants in the Figure) as they are suitable for the TCP/IP network that is the most popular network protocol stack.

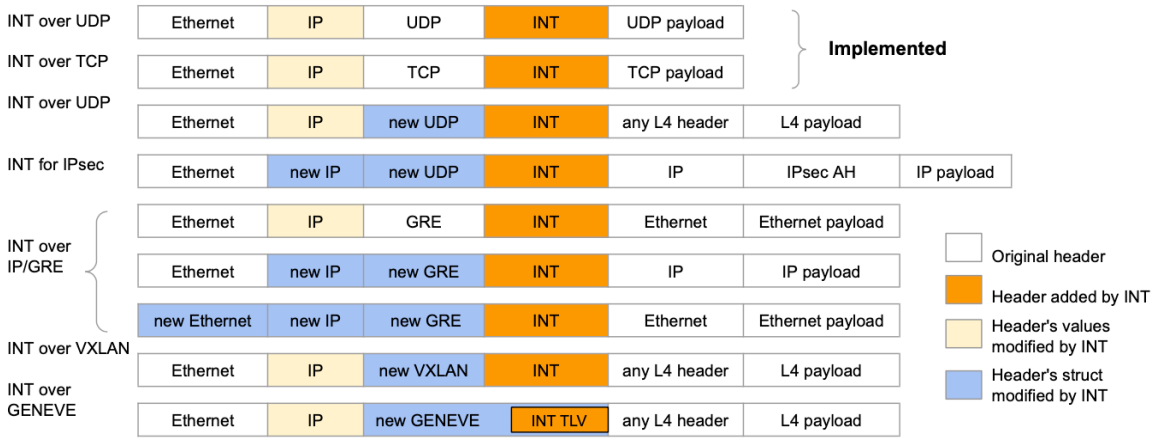

Figure 6: Implementation of INT over TCP/UDP

In Figure 7 we show the details of an INT packet header encapsulated over TCP as proposed by the P4 working group. “Shim” is the INT header which contains the general information such as the total length of the INT metadata that is appended to the packet, while the Differentiated Services Code Point (DSCP) is a copy of the original DSCP in the IP header. An INT packet is identified by a specific and unique value of DSCP in the IP header. When a packet arrives at an INT capable network node, e.g., at the INT source node, the DSCP in the IP header is assigned this specific and unique value which is chosen by the network administrator. This value must be unique to distinguish INT packets from non-INT packets. When the Shim header is removed at the INT sink, the original DSCP value is restored from the Shim header to the IP header.

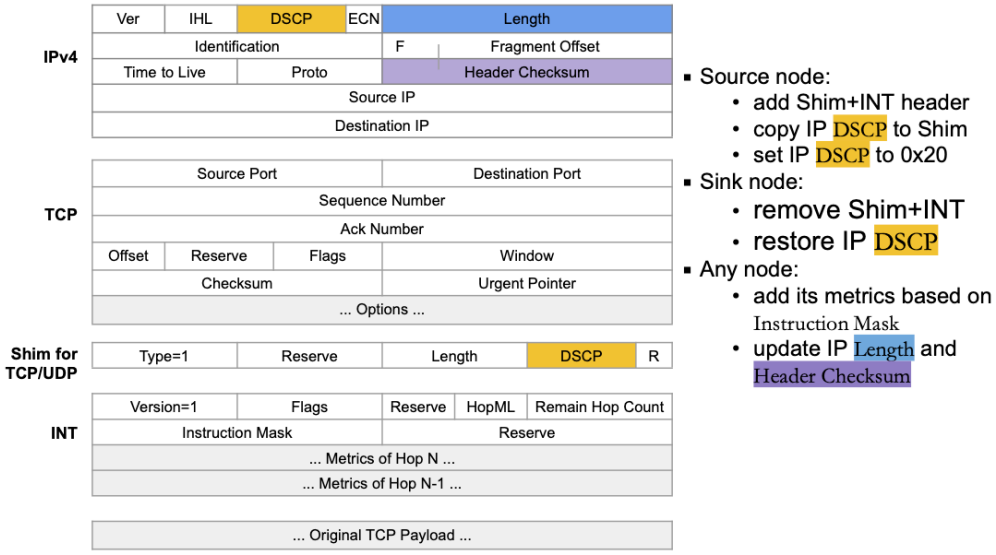

Figure 7: INT packet structure over TCP

A P4 program is inherently monolithic in nature, presenting challenges in crafting programs that are both reusable and modular. Additionally, current P4-based INT frameworks are predominantly configured as main programs dedicated to network monitoring functions. However, there is potential for the primary P4 program to execute different network device logics. For instance, it could function as a Low-Latency Low-Loss Scalable throughput (L4S) switch [HBM+23] or a TSN-enabled switch [NJM+23], demonstrating the versatility of P4 programming beyond its conventional use.

Our INT-based monitoring framework is improved over the existing P4-based INT [Hyu+19] to conform to INT version 2.0 by supporting the INT-MX mode, which does not exist in version 1.0 of the P4 Working Group specification. Although our framework does not support the type-length-value (TLV) data type as P4 does not natively support multiple fields of different lengths.

Our solution is implemented as a library, P4-INT, which is called by the main program. Our P4-INT code consists of three main building blocks, so-called “controls” in P4 language, to realize the three roles of an INT node. This separation allows to easily upgrade each component separately. These blocks are called when the packet is at the P4 parsing stage to perform one of the following actions depending on the role of the current node: (i) embed the required metric values into the packet if the node acts as an INT source or transit node, or (ii) send telemetry data to the INT collector, then remove any telemetry data and reset the packet header to its initial state if the packet is processed by an INT sink node.

In addition to the P4 Working Group metrics, which are supported by the devices, we implement counters to compute the statistics of the numbers of dropped or marked packets. These counters are not packet specific since they show the accumulated total numbers of dropped or marked packets of the device. Thus, we use a P4 “register” to store a counter. A register element is reset to zero to avoid repeatedly reporting once its value has been embedded inside an INT packet.

The framework needs to capture the network device states at the packet level which stands for the atomic event-level in the network. This would heavily consume device resources if all metrics of all

packets were to be analysed, especially with high throughput traffic. To mitigate this high resource consumption, we design P4-INT to perform conditional measurements: on a device, it only measures the metrics requested via the INT instruction bit map; the measurement of given metrics and the complete INT-capabilities of the device can be activated or deactivated at runtime by the device control plane. The overhead in device resource consumption can also be reduced by triggering measurements only for flows matching filters configured by the control plane.

The implementation of INT in P4 language is freely available at https://github.com/DETERMINISTIC6G/deterministic6g_security-solutions/tree/main/int-p4

4.2.2 INT collector

Fast Reaction in a Short Timescale

A monitoring framework is often deployed together with a detection and reaction framework. It should provide precise networking information enabling the latter to react in an appropriate way to ensure the network actually sustains critical services. Our framework provides high precision (i.e., packet by packet) network state information by relying on a push mechanism that uses network sockets, Kafka or Redis message buses, rather than a temporal database [Hyu+19], to deliver the collected metrics values in near real-time.

We extended our existing network traffic analyser, MMT, so that it can act as an INT collector by implementing two new plugins in its deep packet inspection library to parse the two protocols, INT metadata and INT report, which are defined by the P4 Working Group. Originally, MMT is a software solution with a plugin architecture, to passively analyse network packets. By using the new plugin to decode the INT metadata protocol, MMT can eventually be deployed on-the-fly behind an INT-capable device to capture and extract INT metadata directly from the egress packets in the data plane.

Huge Reports generated by Collectors

Whereas MMT can analyse network traffic with very high throughput, it may saturate a connected third-party application with a huge number of reports. For instance, a collector, without any further processing, will generate one report per INT packet. We overcome this bottleneck by implementing two new filters in MMT. MMT reports only the metric values according to the condition predefined by the users via these filters. As such, the framework gives users fine-grained control on the whole monitoring chain, from collecting metrics to forwarding reports.

The event-based filter allows MMT to generate a report only when some metrics' values change. It reduces unnecessary reports while preserving the fine-grained information. Figure 8 shows an example of an event-based filter to tell MMT to send only the reports, named “vary-latency”, when matching the two following conditions:

- the “Hop latency” metric, designated in MMT by the term “int.hop_latencies”, is being collected;
- latency values of each queue change with respect to the last report.

If these conditions are satisfied, then the collector will send a report containing the values of the prescribed metrics in the attributes expression to a Redis message bus that has been configured beforehand.

```
event-report vary-latency {
  event = "int.hop_latencies"
  delta-cond = {"int.hop_latencies","int.hop_queue_ids"}
  attributes = {"ip.src","int.hop_switch_ids","int.
   hop_ingress_times","int.hop_egress_times"}
  output-channel = {redis}}
```

Figure 8 - An event-based report to retrieve only the change of queue latency

The query-based filter allows MMT to periodically generate statistics by performing some query operations on a window of INT metadata. The current supported operations are listed below:

- “sum”: that returns the sum of values;
- “count”: that returns the number of values;
- “avg”: that returns the average of values;
- “var”: that returns the variance of values in the group;
- “diff”: that returns the difference between two consecutive values;
- “last”, “first”: that returns the last or first value in the group respectively.

The implementation of INT collector is freely available at https://github.com/DETERMINISTIC6G/deterministic6g_security-solutions/tree/main/int-collector .

4.2.3 Overhead and packet latency

The INT computation of the device may introduce an extra latency when parsing INT instructions and embedding required metrics into INT packets. Indeed, if all the metrics are collected, the device at source node will add to an INT packet 48 bytes to carry the collected values and 12 bytes of the INT protocol header. Consequently, each packet at the egress port of the INT device contains 60 additional bytes.

To evaluate the latency overhead, we carried out initial experiments of INT on the communication over a simulated 5G testbed. In a next step, we will perform these same experiments in a 5G network with a 5G air interface, and a TSN/DetNet network. The 5G testbed is a simulated environment implemented by using two opensource solutions: UERANSIM (https://github.com/aligungr/UERANSIM) for simulating gNodeB and UEs, and open5Gs (https://open5gs.org) for the core network. We implemented simple client and server programs to actively measure the E2E packet latency. The client resides inside a UE and the server beyond the core. We put virtual switches BMv2 (https://github.com/p4lang/behavioral-model) in front of the client and server to perform INT as in Figure 9. The BMv2 switches run our INT implementation in P4 language. The INT function can be enabled or disabled on these switches. When enabling INT, an INT layer will

be introduced into every packet exchanged between these switches. This layer is between the TCP or UDP packet header and the payload. The INT reports are sent to the INT collector, but they are in this case ignored as we are focusing on the latency overhead caused by the processing of INT inside the 2 BMv2 switches when the packets pass through these switches.

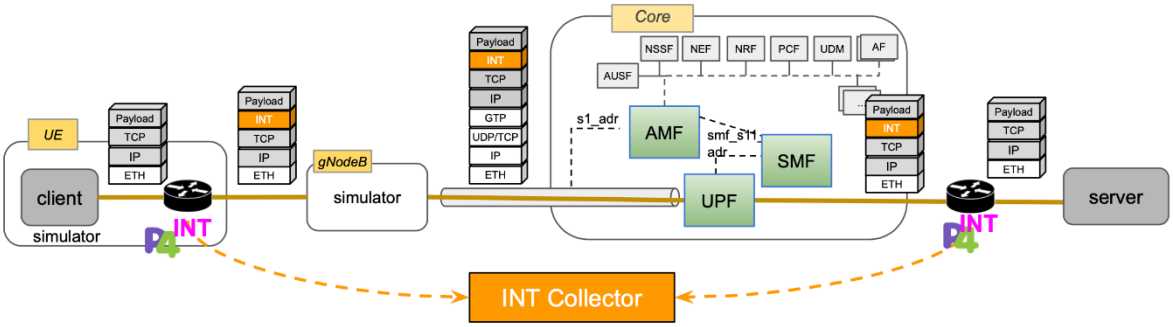

Figure 9: Evaluation of INT overhead on packet latency over a 5G network simulation

The Round-Trip Time (RTT) for the communication between the client and the server is measured actively via the packets generated by the client: it sends the current time to the server. The latter simply sends the received content back. The client then compares the current time and the one received from the server to get the RTT. We want to note that our objective in this test is not to measure RTT but the overhead on the RTT caused by the INT processing.

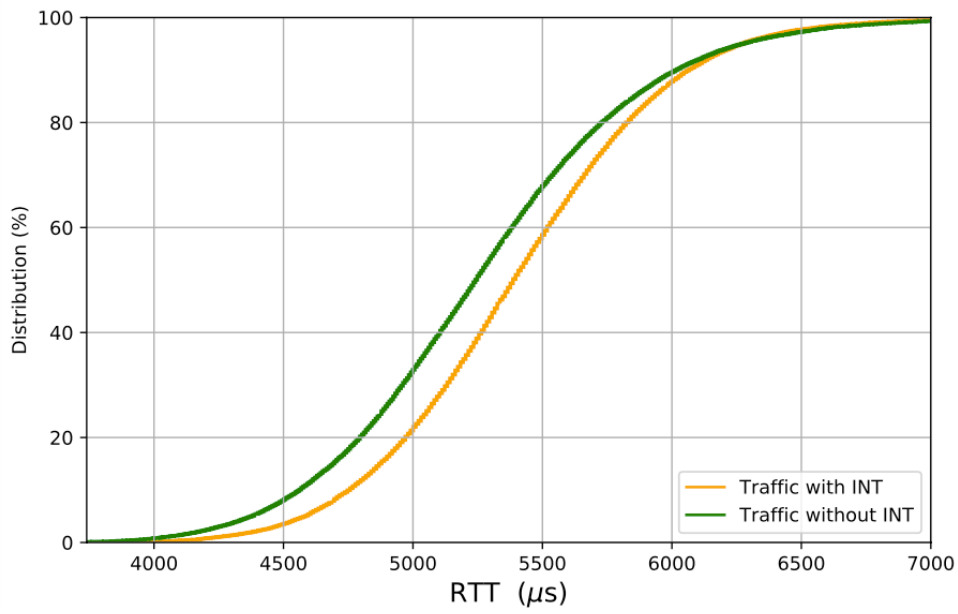

Figure 10 – Overhead of INT processing in packet latency

We conducted several measurements with different TCP packet sizes, 100, 200, 500 and 1000 bytes. For each measurement we sent 10000 packets. We tried to avoid queuing delay or congestion that

could disturb our measurements as a side effect by introducing a 100 ms delay between two consecutive packets in the client program.

We present the results in the Cumulative Distribution Function (CDF) diagram in Figure 10. The horizontal axis stands for the measured RTT values and the vertical axis for their distribution. We can see that more than 95% of the RTT values lie between 4000 to 6500 µs. The average latency without and with INT are 5281 µs and 5415 µs respectively. Therefore, the additional mean latency increases by 2.54% when enabling INT. This low overhead is a promising result for a monitoring system. The overhead would be reduced when INT is processed by physical switches since they use dedicated hardware acceleration techniques. We recognize that this is not a definitive analysis, and we will need to conduct the experiments over a 5G air interface and TSN/DetNet network as a next step in the evaluation phase of the DETERMINISTIC6G project. Nonetheless, we feel this is an encouraging result and the additional latency introduced by INT should not alter the bounded latency SLA requirement of the DETERMINISTIC6G system.

5 Software solution for Attack Generation & Detection

In this section we present two open-source solutions for attack traffic generation and detection. The detection is currently done by recognising the attack signature. We will explore AI/ML attack detection techniques in the next phases of the DETERMINISTIC6G project.

5.1 Network traffic attack generation

5Greplay [ZHW+21] is an open-source network traffic fuzzer dedicated for 5G protocols. It is open source and developed by Montimage. It is released as open source under Apache 2.0 license. 5Greplay can be used to replay or modify traffic towards 5G components in real-time. It can be useful for identifying issues or vulnerabilities in 5G networks and for testing different scenarios. It includes the following features:

- Online and offline processing of network packets/sessions;
- Systematically manipulating and injecting malformed packets that are accepted by the 5G network target components to evaluate their robustness;
- Simulate DoS attacks by amplifying network traffic;
- Performing 5G security test cases;
- Stress and fuzz testing of a 5G component.

The main workflow of 5Greplay is depicted in Figure 11. The input of 5Greplay is network traffic, which can be from a network interface controller (NIC) or pre-captured and saved in the form of a PCAP file, a set of mutation rules, and a configuration file. Once a packet is processed by the tool, the context written by the user in the mutation rules will determine if the packet will be mutated or not. If the

packet is to be mutated, the specified action embedded in the mutation rules will determine what type of mutation of the packet must be done. The mutated packet is then forwarded to the output NIC, together with the non-mutated packets, depending on the default action contained in the configuration file. The output packets can also be saved as a PCAP file, which can be used for further investigations and repeating the tests.

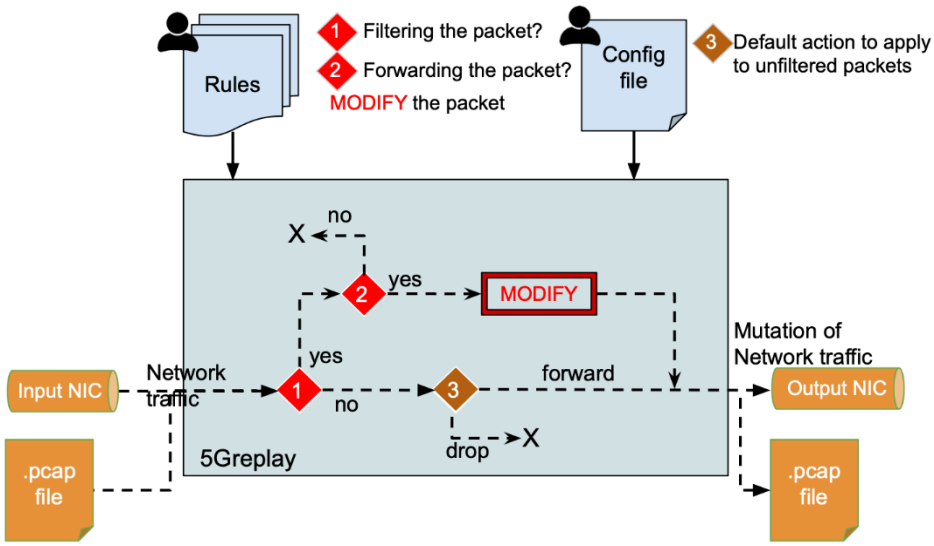

Figure 11 – 5Greplay main architecture

5Greplay relies on an opensource C library, MMT-DPI, which is also developed by Montimage. This library implements Deep Packet Inspection (DPI) to analyse network packets to classify the network protocol being used or the application a packet belongs to, and extracts network and application-based events, such as protocol attribute values, and network and application QoS parameters. DPI is an advanced technique that classifies packets by examining the content of packets passing through a given checkpoint and uses signatures in the packets' headers and payloads to identify their type when not possible to do otherwise, e.g., when the packets are encrypted. MMT-DPI has a plugin architecture for easily adding new protocols and analysis techniques (including machine learning techniques). It provides a set of public APIs for integration with third party probes. MMT-DPI supports more than 800 network protocols and applications, especially 5G network protocols, such as, GTP, GTPv2, NGAP, NAS, NAS-AP, HTTP2. During the DETERMINISTIC6G project, we will extend the tool to support dedicated protocols concerning time sensitive network (e.g., PTP, gPTP) and other relevant protocols (e.g., MPLS, SNMP, PCEP, GMPLS).

After adding the appropriate parser for a protocol to MMT-DPI, 5Greplay can use it for generating attack traffic using the protocol. For DoS attacks, 5Greplay modifies each sequence number in the PTP header and payload with increasing numbers, then multiplies this packet to amplify the power of the attack. In fuzzing attacks, instead, 5Greplay can generate random PTP packet payloads and inject them into the network to test the robustness and resilience of the DETERMINISTIC6G platform.

The tool is freely available at https://github.com/DETERMINISTIC6G/deterministic6g_security-solutions/tree/main/5greplay .

5.2 Signature-based attack detection

MMT-Security is a signature-based monitoring solution that allows analysing network traffic according to a set of properties. These properties contain signatures that formally specify security goals, or malicious behaviours related to the monitored system. The MMT-Security property model is inspired by Linear Temporal Logic (LTL) and has the following two types of properties;

- Properties that describe the normal, legitimate behaviour of the application or protocol under analysis. Consequently, the non-respect of the property indicates a potential violation of a safety or security requirement, e.g., all the ports in a computer must be closed unless they are being used by an authorised application;
- Attacks that describe malicious behaviour corresponding to an attack model, a vulnerability or misbehaviour. In this case, the identification of the property indicates the detection of a potential incident, e.g., a big number of requests in a short period of time could be a DoS attack.

MMT-Security properties are expressed in XML format, due to its simplicity and straightforward structure verification. A property is structured as a binary ordered tree as shown in Figure 12. A property itself is composed of two main components: a "*context*" on the left branch and "*trigger*" on the right branch. This separation enhances the clarity and effectiveness of security verification processes. The *context* defines the conditions that need to be met for the property to even be considered, and it acts as a prerequisite for evaluating the *trigger*. The *trigger* contains the specific conditions or events that must be satisfied for the property to be considered valid. The left nodes are "*atomic events*" which are essentially the basic, indivisible actions or conditions that the system is expected to meet. These atomic events are derived from traces or logs of the system's operation.

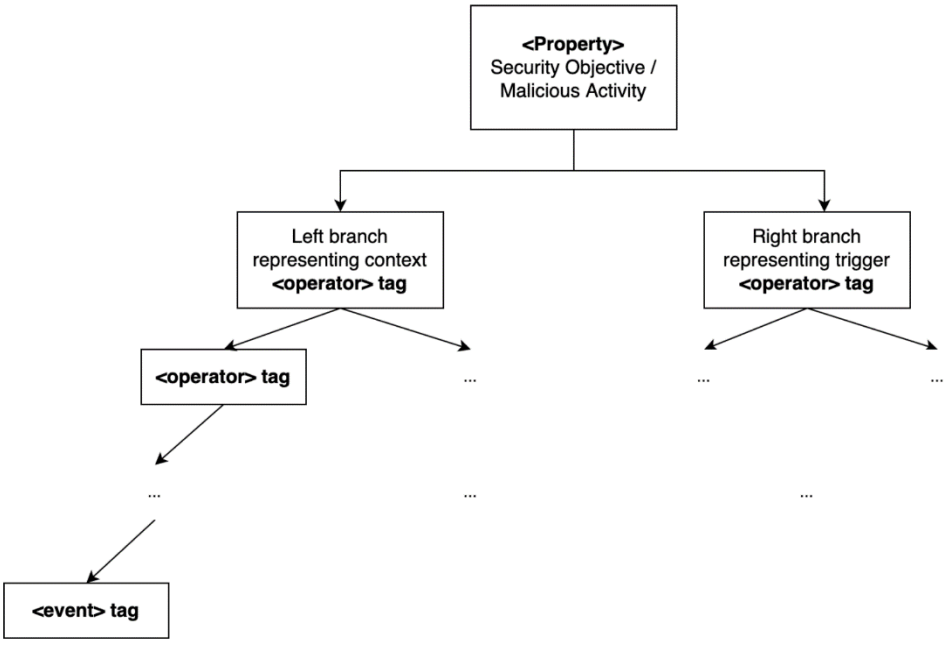

Figure 12: Property structure in MMT-Security

The Appendix provides two examples of rules: 1) A simple rule that involves detecting a DoS caused by a compromised network element that issues more than one control plane packet within a millisecond; and 2) A more complex rule that detects a variation of packet latency greater than a given threshold.

The tool is freely available on the project's github (see Table 1) and from https://github.com/DETERMINISTIC6G/deterministic6g_security-solutions/tree/main/mmt-security .

6 Conclusions

This deliverable provided a description of the security monitoring framework developed under the DETERMINISTIC6G project. It is a document designed to augment the accompanying software release with a contextual analysis of deterministic networks and high-value emerging techniques for security monitoring. It provided an overview of the key challenges, required standards and security-by-design principles that underpin the protection of deterministic networks. Based on the requirements and architecture developed as part of the project, the framework is now in its first release, ready to enhance the resilience of deterministic E2E communication scenarios. The flexibility of this framework is illustrated by its alignment with the 3GPP ZSM model, which is ready to address the dynamic security needs of current and future networks.

In the ever-complex landscape of wireless, virtualized, and edge-driven networks, the monitoring framework is an essential enabler for the envisioned E2E and multi-domain security architecture. It can harness state-of-the-art AI/ML algorithms for proactive threat intelligence, alongside innovative in-band networking and P4 programmable data planes for real-time policy enactment. The system is designed for dynamic adaptation, capable of swiftly responding to new threats with real-time updates to its defence strategies. Through the integration of high-precision telemetry and programmable data plane, the architecture achieves a precise and adaptable security solution, addressing the challenges that exist for security management in low-latency and deterministic network applications.

Future work will concern further developing the security-by-design architecture, the specification of the scenarios, the description of the ML algorithms, and the implementation of the E2E multi-domain security concepts. The complete final security-by-design architecture will be presented in deliverable D1.4: Final report on DETERMINISTIC6G architecture [DET25-D1.4]. In deliverable D3.5: Multi-domain end-to-end schedules [DET25-D3.5], the topics that will be further developed are:

- The AI/ML-based analysis for the detection, the identification of the root causes, and the response;
- The process pipeline for network traffic collection and generation for learning and test phases;
- The features extracted for feeding the ML models;
- The dynamic and automated security management based on closed-loop security;
- The security of E2E scheduling and multi-domains.

Appendix: Example of rules

1. A rule for detecting a DoS caused by a compromised network element

This rule employs temporal logic and consists of two events linked to consecutive packets from the same session. It enables detection if these packets fall within a specific time interval, defined as [delay_min, delay_max], where the delays are measured in microseconds.

```
<beginning>

<!--This rule detects whether there exists one flow having two
consecutive packets within 1 millisecond in the control plane traffic-
->

<property value="THEN" delay_units="ms" delay_min="0" delay_max="1"
property_id="1" type_property="ATTACK" description="DoS attack in the
control plane: two consecutive TCP packets from a single flow within
1 millisecond">

          <event value="COMPUTE" event_id="1" description="First
          packet" boolean_expression="( ip.src != ip.dst )"/>

          <event value="COMPUTE" event_id="2" description="Second
          packet" boolean_expression="(((ip.src == ip.src.1)
          &amp;&amp; (ip.dst == ip.dst.1)) &amp;&amp; (tcp.dest_port
          == tcp.dest_port.1))"/>

</property>

</beginning>
```

2. A rule that detects a variation of packet latency greater than a given threshold

This rule is designed to ascertain if the variation in packet latency between two observation points exceeds a predetermined threshold. It uses INT to embed the timestamp in the packets going through the first observation point so that they can be used in the second observation point to calculate the variations in the latency of each packet. In this way, one can detect abnormal variations in the latency and it doesn't matter if the clocks of the two observation points are synchronised or not. It uses two embedded functions (i.e., functions written in the C language that are used to verify the rule during the execution of the analysis). One function is for determining if the packets observed are in the same session and the other to calculate the latency variation.

```
<beginning>

<!--

This rule detects whether the variation of packet latency between 2
observation points is greater than a given threshold.

        - at each observation point, we put an INT node which is a P4
          switch.
```

- when a packet passing a P4 switch, a local timestamp T of the switch is embedded inside the packet w.r.t. via Inband Network Telemetry protocol.

- as we have 2 P4 switches, at INT-collector we obtain T1 and T2 for each packet. The latency of the packet between these 2 switches is L = T1-T2 (Note: L is not an absolute time value as T1 and T2 are measured by 2 different local clocks)
- for all packets in a time window of a TCP session, if the difference between their L values is greater than 100 microsecond, then an alert will be raised.
    + this verification is applied only for packets at a given time window
    + time window is 1 millisecond
    + as network state might be different at different time window, then packet latency can be also involved

Note: the threshold of 100 microsecond and the time window of 1 millisecond are just given as example. They are configurable by modifying them in the XML rule.

Assumption: the frequency of clocks at the two P4 switches is constant. This assumption ensures that the latency values L1 and L2 of two packets are comparable.

Explanation of XML rule:

- event 1: first packet
    + (ip.src != ip.dst): the packet is not sent to its source
    + (int.num_hop == 2): the packet passed through 2 INT switches
- event 2: second packet
    + #is_same_session(ip.src.1, ip.dst.1, tcp_src_port.1, tcp.dest_port.1, ip.src, ip.dst, tcp_src_port, tcp.dest_port): return true if the packets are in the same TCP session
    + (int.num_hop == 2): the packet passed through 2 INT switches
    + #latency_variation(int.hop_ingress_times.1, int.hop_ingress_times) : return the variation of latencies of two packets

```
Note:

            + #is_same_session is implemented by a C macro

            + #latency_variation is implemented by a C function

            + ip.src.1 is used in event 2 to refer to "ip.src" of
              the event 1

-->


<property value="THEN" delay_units="ms" delay_min="0" delay_max="1"
property_id="2" type_property="ATTACK" description="Latency variation
of packets of the same TCP session must less than 100ns if they are
in a time window of 1ms">

       <event   value="COMPUTE"   event_id="1"   description="First
         packet"     boolean_expression="((ip.src     !=     ip.dst)
         &amp;&amp;(int.num_hop == 2))"/>

       <event   value="COMPUTE"   event_id="2"   description="Second
         packet"    boolean_expression="((#is_same_session(ip.src.1,
         ip.dst.1, tcp.src_port.1, tcp.dest_port.1, ip.src, ip.dst,
         tcp.src_port, tcp.dest_port) &amp;&amp; (int.num_hop == 2))
         &amp;&amp;     (#latency_variation(int.hop_ingress_times.1,
         int.hop_ingress_times) > 100)) "/>

</property>


<embedded_functions><![CDATA[

#define is_same_session(s1_ip, d1_ip, s1_port, d1_port, s2_ip, d2_ip,
       s2_port, d2_port) (s1_ip == s2_ip && d1_ip == d2_ip && s1_port
       == s2_port && d1_port == d2_port)


// an array of timestamp to represent "hop_ingress_times" attribute
of the INT protocol

typedef struct mmt_u64_array_struct{

         uint32_t len;

         uint64_t data[64];

} mmt_u64_array_t;


static inline int64_t get_latency(const void *a){

         mmt_u64_array_t *hop_ingress_times = (mmt_u64_array_t *) a;
```

```
            // not enough 2 INT nodes
            if( hop_ingress_times->len < 2 ) return 0;
            //get the difference of timestamps between 2 hops
            // need to divide to 1000 to get microsecond from nanosecond
            return (hop_ingress_times->data[0] - hop_ingress_times ->
            data[1]) / 1000;
}

static inline uint64_t latency_variation(const void *a, const void
*b){
            int64_t l1, l2;
            l1 = get_latency( a );
            l2 = get_latency( b );

//return the absolute value of difference between the 2 latencies
            if( l1 > l2 )
                        return (l1 - l2);
            else
                        return (l2 - l1);
}
]]></embedded_functions>

</beginning>
```

References

[BL21]	Boyang Zhou, Liang Cheng: Mitigation of Scheduling Violations in Time-Sensitive Networking using Deep Deterministic Policy Gradient. FlexNets@SIGCOMM 2021: 32-37
[BL21]	Boyang Zhou, Liang Cheng: Mitigation of Scheduling Violations in Time-Sensitive Networking using Deep Deterministic Policy Gradient. FlexNets@SIGCOMM 2021: 32-37
[CB+22]	Chuwen Zhang, Boyang Zhou *et al*. TSN-Peeper: an Efficient Traffic Monitor in Time-Sensitive Networking, *2022 IEEE 30th International Conference on Network Protocols (ICNP)*, Lexington, KY, USA, 2022, pp. 1-11, doi: 10.1109/ICNP55882.2022.9940335.
[DET23-D1.1]	DETERMINISTIC6G deliverable D1.1: DETERMINISTIC6G use cases and architecture principles; online 6/2023: https://deterministic6g.eu/images/deliverables/DETERMINISTIC6G-D1.1-v1.0.pdf
[DET23- D2.2]	DETERMINISTIC6G deliverable D2.2: First report on time synchronization for E2E time awareness; online planned 12/2023: https://deterministic6g.eu/index.php/library-m/deliverables
[DET25- D1.4]	DETERMINISTIC6G deliverable D1.4: Final report on DETERMINISTIC6G architecture; online planned 6/2025: https://deterministic6g.eu/index.php/library-m/deliverables
[DET25- D3.5]	DETERMINISTIC6G deliverable D3.5: Multi-domain end-to-end schedules; online planned 4/2025: https://deterministic6g.eu/index.php/library-m/deliverables
[ESF23]	Ergenç, D., Schenderlein, R. and Fischer, M., 2023. TSNZeek: An Open-source Intrusion Detection System for IEEE 802.1 Time-sensitive Networking. arXiv preprint arXiv:2303.11492.
[FFA+19]	F. Paolucci, F. Civerchia, A. Sgambelluri, A. Giorgetti, F. Cugini, and P. Castoldi, “P4 Edge Node Enabling Stateful Traffic Engineering And Cyber Security,” Journal of Optical Communications and Networking, vol. 11, no. 1, pp. A84–A95, 2019
[GDJ+23]	Gourav Prateek Sharma, Dhruvin Patel, Joachim Sachs, Marilet De Andrade, János Farkas, János Harmatos, Balázs Varga, Hans-Peter Bernhard, Raheeb Muzaffar, Mahin Ahmed, Frank Dürr, Dietmar Bruckner, Edgardo Montes de Oca, Drissa Houatra, Hongwei Zhang, James Gross: Toward Deterministic Communications in 6G Networks: State of the Art, Open Challenges and the Way Forward. IEEE Access 11: 106898-106923 (2023)
[Hexa-X]	https://hexa-x.eu
[Hyu+19]	Hyun et al. “Real-time and fine-grained network monitoring using in-band network telemetry,” International Journal of Network Management, vol. 29, no. 6, 2019
[IEEE-1588PTP]	https://standards.ieee.org/ieee/1588/6825/
[IEEE-8021AE]	https://1.ieee802.org/security/802-1ae/
[IEEE-8021ASrev]	https://1.ieee802.org/tsn/802-1as-rev/
[IEEE-8021AR]	https://1.ieee802.org/security/802-1ar/
[IEEE-8021Qci]	https://1.ieee802.org/tsn/802-1qci/

[IEEE-8021Qav]	https://www.ieee802.org/1/pages/802.1av.html
[IEEE-8021X]	https://1.ieee802.org/security/802-1x/
[IETF-RTC9055]	https://datatracker.ietf.org/doc/rfc9055/
[INT+21]	The P4.org Working Group, “In-band Network Telemetry (INT) Data-plane Specification V2.1,” Tech. Rep., 2021.
[Inspire-5Gplus]	https://www.inspire-5gplus.eu/
[Inspire-5Gplus whitepaper]	https://doi.org/10.5281/zenodo.4288658
[JBG+19]	J. Farkas, B. Varga, G. Miklós , J. Sachs, 5G-TSN integration meets networking requirements, Ericsson Technology Review, August 2019.
[JRJ+20]	Jordi Ortiz Murillo, Ramon Sanchez-Iborra, Jorge Bernal Bernabé, Antonio F. Skarmeta, Chafika Benzaid, Tarik Taleb, Pol Alemany, Raul Muñoz, Ricard Vilalta, Chrystel Gaber, Jean-Philippe Wary, Dhouha Ayed, Pascal Bisson, Maria Christopoulou, George Xilouris, Edgardo Montes de Oca, Gürkan Gür, Gianni Santinelli, Vincent Lefebvre, Antonio Pastor, Diego R. López: INSPIRE-5Gplus: intelligent security and pervasive trust for 5G and beyond networks. ARES 2020: 105:1-105:10
[LWZ+21]	L. Tan, W. Su, W. Zhang, J. Lv, Z. Zhang, J. Miao, X. Liu, and N. Li, “In-band Network Telemetry: A Survey,” Computer Networks, vol. 186, no. December 2021
[MCP+23]	Mitev, Miroslav, Arsenia Chorti, H. Vincent Poor, and Gerhard Fettweis. "What physical layer security can do for 6g security." IEEE Open Journal of Vehicular Technology (2023).
[MHK+19]	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’. arXiv, Oct. 21, 2019. doi: 10.48550/arXiv.1908.09646.
[NJM+23]	N. S. Bulbul, J. J. Kruger, and M. Fischer, “TSN Gatekeeper: Enforcing stream reservations via P4-based in-network filtering,” in IFIP Networking 2023, 2023.
[NNJ+20]	N. Foster, N. Mckeown, J. Rexford, G. Parulkar, L. Peterson, and O. Sunay. Using deep programmability to put network owners in control. Computer Communication Review, 50(4):82–88, 2020. vol. 11, no. 1, pp. A84–A95, 2019
[SPS23]	Sharma, G.P. Patel, D., Sachs, J. et al., 2023. Towards Deterministic Communications in 6G Networks: State of the Art, Open Challenges and the Way Forward. arXiv preprint arXiv:2304.01299.
[SPS+23]	G. P. Sharma, D. Patel, J. Sachs, M. De Andrade, J. Farkas, J. Harmatos, B. Varga, H. -P., Bernhard, R. Muzaffar, M. Ahmed, F. Duerr, D. Bruckner, E.Montes de Oca, D. Houatra, H. Zhang and J. Gross, "Toward Deterministic Communications in 6G Networks: State of the Art, Open Challenges and the Way Forward," in IEEE Access, vol. 11, pp. 106898-106923, 2023, doi: 10.1109/ACCESS.2023.3316605: 10.23919/AEITAUTOMOTIVE58986.2023.10217195.
[ZC23]	Zhou, Boyang & Cheng, Liang. (2023). TSN-VM: A Real-time and Distributed Algorithm for Scheduling-Violation Mitigation in Time-Sensitive Networking. 10.36227/techrxiv.24588741.v1.

[ZHW+21]	Z. Salazar, H. N. Nguyen, W. Mallouli, A. R. Cavalli, and E. M. Montes De Oca, “5Greplay: A 5G Network Traffic Fuzzer - Application to Attack Injection,” in Proc. of ARES, 2021, vol. 1, no. 1, pp. 1–12.Alghamd and M. Schukat, ‘A Detection Model Against Precision Time Protocol Attacks’, in 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), Mar. 2020, pp. 1–3. doi: 10.1109/ICCAIS48893.2020.9096742.
[3GPP16-22261]	3GPP TS 22.261, “Service requirements for the 5G system,” v19.4.0

List of abbreviations

AI/ML	Artificial Intelligence / Machine Learning
AR	Augmented Reality
CNC	Centralized Network Controller
CPD	Change Point Detection
CUC	Centralized User Configuration
DET	DETERMINISTIC6G
DetNet	Deterministic Networking
DDoS	Distributed Denial of Service
DoS	Denial of Service
DPI	Deep Packet Inspection
DSCP	Differentiated Services Code Point
E2E	End-to-End
ECN	Explicit Congestion Notification
GCL	Gate Control List
GM	Grand Master
GMPLS	Generalized Multi-Protocol Label Switching
GPRS	General Packet Radio Service
gPTP	generic Precision Time Protocol
GTP	GPRS Tunnelling Protocol
HLA	High-Level Architecture
HTTP2	Hypertext Transfer Protocol version 2
ICT	Information and Communication Technology
IETF	Internet Engineering Task Force
IoT	Internet of Things
INT	In-band Network Telemetry
INT-XD INT	INT eXport Data mode,
INT-MX	INT eMbed instructXions mode
INT-MD	INT eMbed Data mode

IP	Internet Protocol
IPPM	IP Performance Measurement
L4S	Low-Latency Low-Loss Scalable throughput
MACsec	Media Access Control security
mMTC	massive Machine-Type Communications
ML	Machine Learning
MMT	Montimage Monitoring Tool
MPLS	Multiprotocol Label Switching
MTD	Moving Target Defence
MTU	Maximum Transmission Unit
NAS	Non-Access-Stratum protocol
NAS-AP	Non-Access-Stratum Access Point
NETCONF	Network Configuration Protocol
NGAP	Next Generation Application Protocol
OAM	Operation Administration and Maintenance
OSI	Open Systems Interconnection
P4	Programming Protocol-independent Packet Processors
PCEP	Path Computation Element Communication Protocol
QoS	Quality of Service
RAN	Radio Access Network
RCA	Root Cause Analysis
RESTCONF	Representational State Transfer Configuration Protocol
SAE	Security Analytics Engine
SDN	Software Defined Networking
SMD	Security Management Domain
SNMP	Simple Network Management Protocol
tR	time Receiver
TSN	Time-Sensitive Networking
tT	time Transmitter
URLLC	Ultra Reliable Low-Latency Communications
VLAN	Virtual Local Area Network
YANG	Yet Another Next Generation
ZSM	Zero-touch network and Service Management
PTP	Precision Time Protocol