HAW HAMBURG

Bachelor's Thesis

Thamo Weichler

Combining ATS and Non-ATS Traffic in Shared TSN Queues

Faculty of Engineering and Computer Science
Department of Computer Science

Fakultät Technik und Informatik
Department Informatik

Thamo Weichler

Combining ATS and Non-ATS Traffic in Shared TSN Queues

Bachelor's Thesis
Submitted to the Department of Computer Science
Faculty of Engineering and Computer Science
University of Applied Sciences Hamburg
In Partial Fulfilment of the Requirements for the Degree of
Bachelor of Science in *Informatik Technischer Systeme*

First Supervisor: Prof. Dr. Franz Korf
Second Supervisor: Prof. Dr. Tim Tiedemann

Hamburg, 16. May 2025

Thamo Weichler

Title

Combining ATS and Non-ATS Traffic in Shared TSN Queues

Index Terms
Time-Sensitive Networking, TSN, IEEE 802.1Q, IEEE 802.1Qcr, Asynchronous Traffic Shaping, ATS, Traffic Shaping, Scheduling, In-Vehicle Networks, Ethernet, LAN, VLAN

Abstract

Time-Sensitive Networking (TSN) enhances Ethernet-based networks with mechanisms to ensure deterministic communication, crucial for real-time applications. Among these mechanisms, Asynchronous Traffic Shaping (ATS), as defined in IEEE 802.1Qcr, provides bounded latency and aims to isolate traffic streams effectively. However, the standard does not explicitly address scenarios where unscheduled traffic shares egress queues with ATS-managed streams, leading to potentially undefined system behaviour.

This thesis investigates the challenges arising from the co-existence of ATS scheduled and unscheduled traffic within shared egress queues. Three strategies are proposed and evaluated through microbenchmarking to assign eligibility times to unscheduled frames, facilitating their integration into ATS queues without necessitating full ATS compliance for the entire traffic class.

The evaluation demonstrates that the presented strategies are capable of resolving the immediate conflict between scheduled and unscheduled traffic, often maintaining acceptable performance levels. Nonetheless, significant limitations are identified, including the absence of traffic rate enforcement for processed streams, lack of analytical latency bounds and the risk of priority inversion. These constraints render the strategies unsuitable for safety-critical applications. Despite these constraints, the strategies offer a flexible solution for non-critical applications, particularly when additional ATS schedulers are impractical. Their effectiveness, however, is highly sensitive to traffic patterns and network topology. To ensure system resilience, further research is recommended.

By providing a fundamental assessment of the identified problem, this thesis underscores the necessity for robust design practices.

Thamo Weichler

Thema der Arbeit

Kombination von ATS- und Nicht-ATS-Datenverkehr in gemeinsamen TSN Queues

Stichworte

Echtzeit Netzwerke, TSN, IEEE 802.1Q, IEEE 802.1Qcr, Asynchronous Traffic Shaping, ATS, Traffic Shaping, Scheduling, Fahrzeug Netzwerke, Ethernet, LAN, VLAN

Kurzzusammenfassung

Time-Sensitive Networking (TSN) erweitert Ethernet-basierte Netzwerke um Mechanismen zur Gewährleistung deterministischer Kommunikation, die für Echtzeitanwendungen unerlässlich ist. Ein wesentlicher Bestandteil ist das Asynchronous Traffic Shaping (ATS) gemäß IEEE 802.1Qcr, das eine rein lokale Planung und weitestgehende Separation von Datenströmen ermöglichen soll. Der Standard spezifiziert jedoch nicht explizit, wie mit nicht-geplantem Verkehr umzugehen ist, der sich Ausgangswarteschlangen mit ATS-gesteuerten Strömen teilt. Dies kann potenziell zu undefiniertem Systemverhalten führen.

Diese Arbeit untersucht drei Strategien, die darauf abzielen, nicht-geplanten Verkehr in ATS Ausgangswarteschlangen, sogenannten TSN Queues, zu integrieren, ohne eine vollständige ATS-Konformität für die gesamte Prioritätsklasse vorauszusetzen.

Die Evaluation zeigt, dass diese Strategien den unmittelbaren Konflikt zwischen geplantem und nicht-geplantem Verkehr lösen können und das häufig bei akzeptabler Leistung. Dennoch werden erhebliche Einschränkungen identifiziert, darunter das Fehlen einer Durchsatzbegrenzung für verarbeitete Ströme, das Fehlen analytischer Latenzobergrenzen und das Risiko einer Prioritätsumkehr. Diese Limitationen machen die Strategien für sicherheitskritische Anwendungen ungeeignet. Für nicht-kritische Anwendungsbereiche bieten sie jedoch eine flexible Lösung, insbesondere wenn die Nutzung zusätzlicher ATS-Scheduler nicht praktikabel ist. Ihre Wirksamkeit hängt jedoch stark von Mustern im Datenverkehr und von der Netzwerktopologie ab. Zur Gewährleistung der Systemstabilität wird weiterführende Forschung empfohlen.

Mit der grundlegenden Analyse des identifizierten Problems hebt die Arbeit die Notwendigkeit robuster Designpraktiken für echtzeitfähige Netzwerke hervor.

Contents

List of Figures

List of Tables

Abbreviations

ATS	Asynchronous Traffic Shaping
CAN	Controller Area Network
CBS	CommittedBurstSize
CBSa	Credit-Based Shaper algorithm
CIR	CommittedInformationRate
ECU	Electronic Control Unit
FIFO	First In - First Out
GETT	Group Eligibility Time Tagging
IFG	Interframe Gap
IPV	Internal Priority Value
LAN	Local Area Network
LIN	Local Interconnect Network

MAC	Media Access Control
MRT	MaximumResidenceTime
NETT	Non-ATS Eligibility Time Tagging
PCP	Priority Code Point
PSFP	Per Stream Filtering and Policing
QoS	Quality of Service
RTCS	Real-Time Computer System
RTS	Real-Time System
SETT	Super-Group Eligibility Time Tagging
SPQ	Strict Priority Queuing
TAS	Time-Aware Shaping
TETT	Tail-Element Eligibility Time Tagging
TSA	Transmission Selection Algorithm
TSN	Time-Sensitive Networking
VLAN	Virtual Local Area Network

1 Introduction

As embedded systems continue to evolve, the demand for predictable, low-latency communication across Ethernet networks has become critical in domains such as automotive, avionics, factory automation, and robotics. To meet these stringent real-time requirements, the IEEE 802.1Q Time-Sensitive Networking (TSN) standard enhances traditional Ethernet with a suite of mechanisms that ensure bounded delay and prioritisation of traffic classes. Among these mechanisms is Asynchronous Traffic Shaping (ATS), a traffic shaping method that schedules frame transmissions using credit counters to regulate the egress traffic rate.

Despite its benefits, the TSN standard does not define how to handle traffic that is not subject to ATS scheduling but shares the same egress queue as ATS-scheduled traffic. This oversight leads to ambiguity in system behaviour and creates scenarios where latency guarantees may be compromised, undermining the reliability TSN aims to provide.

This thesis addresses this gap by introducing and analysing three strategies for assigning inferred eligibility times to unscheduled frames. These strategies are designed to allow unscheduled traffic to co-exist within ATS queues. Through controlled experiments and systematic evaluation, the thesis investigates the feasibility, performance, and trade-offs of each approach, ultimately providing guidance for future TSN network design.

Chapter 2 provides the technical background, reviewing the TSN standard with particular focus on ATS and related concepts.

Building onto that, Chapter 3 defines the core problem termed the Non-ATS Conflict, illustrating why unscheduled frames in ATS queues create ambiguous behaviour and potential timing violations.

Chapter 4 outlines several conceptual approaches to resolving this conflict and introduces the three specific strategies selected for further exploration.

Chapter 5 details the evaluation methodology, including simulation setup, network configurations and the performance metrics used to assess each strategy.

Chapter 6 presents and analyses the simulation results across a range of conditions, comparing the behaviour and effectiveness of the proposed strategies against native ATS shaping.

Finally, Chapter 7 summarises the key findings and discusses directions for future work.

2 Technical Background

In this chapter establishes the foundational concepts necessary to understand the timing, queuing and scheduling mechanisms that underpin deterministic communication in modern Ethernet networks. The chapter starts with an overview of real-time systems, defining hard, firm and soft real-time requirements. Building on that, the Time-Sensitive Networking family of IEEE 802.1 standards is introduced, highlighting key-mechanisms for achieving bounded delay and isolation between traffic classes. This section will explain the forwarding process of frames by switches, with focus on Asynchronous Traffic Shaping. Together, these core topics provide the technical vocabulary and architectural context for the solution strategies introduced later in this thesis.

2.1 Real-Time Systems

Real-time systems (RTSs) operate under strict timing constraints. Each task in an RTS must complete within a specified time window, known as a deadline. These deadlines are typically determined by external, application-specific factors such as physical dynamics, user interaction requirements or safety margins. In RTSs, correctness of produced results is defined not only by the logical validity of them, but also by their timeliness. Therefore, the goal of an RTS is not to maximise speed, but to guarantee a response that is generated as fast as necessary, reliably and within strict temporal bounds. The need for predictability makes determinism a fundamental characteristic of Real-Time Systems [3, 4].

If a deadline is missed, the produced result may not be usable anymore, which might result in degraded performance, compromised functionality or complete system failure. Based on the severity of consequence, RTSs are classified into three categories [5]:

- **Soft RTSs:** Occasional deadline misses are tolerable, even though they degrade performance or user experience. A missed deadline does not invalidate the result. Examples include video streaming and online gaming [3].
- **Firm RTSs:** Late results are considered invalid and are typically discarded. While the system may experience a temporary loss in Quality of Service (QoS), its overall stability is not compromised. Typical use cases include automated trading where outdated data cannot be acted upon but the application continues running [5].
- **Hard RTSs:** Deadline misses are intolerable and late results are discarded. A missed deadline constitutes system failure. Live audio processing equipment operates under hard real-time constraints, as missed deadlines lead to perceptible audio artefacts [3].

Depending on the application domain, hard RTSs may also be referred to as safety-critical real-time systems. In areas such as healthcare, aerospace, automotive and industrial automation, missing a hard deadline may result in hazardous consequences, including risks to human life [3].

One form of RTSs are Real-Time Computer Systems (RTCSs). As the name suggests, these systems integrate hardware and software components that must operate under strict timing constraints. Importantly, the sub-systems do not function independently. Rather, they form a distributed RTCS in which the components must communicate in a deterministic and timely manner.

An example of a RTCS are autonomous vehicles, which are composed of multiple interconnected components such as sensors, Electronic Control Units (ECUs) and actuators. For example, when the obstacle detection sub-system identifies a hazard, it must inform the braking controller to initiate emergency deceleration. This process must respect deadlines that account for vehicle speed and stopping distance to ensure sufficient time to avoid collisions. Both the detection and braking sub-systems must respond within their respective timing bounds, but most importantly the communication network must also deliver messages within bounded latencies. Transmitting control signals too late may result in a failure to respond in time, rendering the system ineffective. Consequently, in-vehicle communication networks must meet real-time constraints.

2.2 Time-Sensitive Networking

Traditional in-vehicle networks predominantly utilise bus technologies such as Local Interconnect Network (LIN), Controller Area Network (CAN) and FlexRay, valued for their robustness and real-time capabilities. However, with the advent of advanced driver assistance systems, autonomous driving features and infotainment applications, the limitations of these networks in terms of transmission rates have become increasingly evident. To meet these complex requirements, automotive manufacturers have resorted to deployment of multiple dedicated in-vehicle networks, each tailored to a specific functional domain and level of traffic criticality, thereby isolating safety-critical traffic from less time-sensitive data. While it facilitates safety certification and reduces the throughput requirements of each individual network, this approach results in increased wiring complexity and higher development and production costs [6, 7].

A more scalable and cost-effective solution lies in the consolidation of domain-specific in-vehicle networks into a unified high-speed communication backbone, capable of supporting traffic of varying criticality levels. This approach requires a technology that provides both high throughput and deterministic real-time guarantees [6, 8].

Ethernet, a widely adopted standard in Local Area Networks (LANs), provides the necessary bandwidth and has broad industry support. However, standard Ethernet lacks inherent support for real-time communication. To address this, the IEEE 802.1 TSN Task Group has developed a suite of standards that extend the Ethernet Data Link layer to support deterministic transmission behaviour. These standards are collectively referred to as TSN, with the IEEE 802.1Q serving as the foundational standard. Throughout this paper, references of the IEEE 802.1Q standard specifically refer to revision IEEE 802.1Q-2022, which incorporates multiple amendments and sub-standards related to time-critical operations [9, 2].

The following sections provide an overview of the core principles defined by the TSN standard. While these principles are broadly applicable to various types of network nodes, this discussion will primarily focus on switches and their behaviour in handling frames during the forwarding process.

It should be noted that aspects of the IEEE 802.1Q standard that fall outside the scope of this thesis will be simplified or omitted for clarity.

2.2.1 TSN Ethernet Frames and Traffic Prioritisation

Time-Sensitive Networking (TSN) extends Ethernet to enable differentiation between multiple classes of traffic, each with its own latency and bandwidth requirements. To distinguish between these classes, TSN leverages traffic prioritisation, enabling the network to handle each frame according to a specific QoS.

In TSN networks, every frame is tagged with an IEEE 802.1Q tag, as illustrated in Figure 2.1. This is the Virtual Local Area Network (VLAN) tag which is part of the Ethernet Media Access Control (MAC) header. The VLAN tag consists of several fields. Apart from identifying the type of the tag (TPID), the VLAN associated with the frame (VID) as well as the so-called drop eligible indicator (DEI), the most important field is the 3-bit Priority Code Point (PCP) [2].

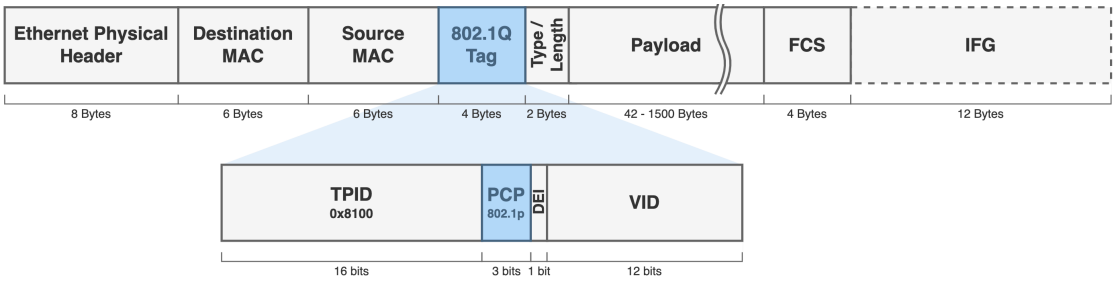


Figure 2.1: TSN Ethernet frame format highlighting the PCP [1, 2].

The PCP field encodes the priority of the frame, mapping it to one of eight priority levels, with 7 being the highest priority and 0 being the lowest. Each priority is associated with a distinct QoS level within the network. Generally, higher priority values correspond to more time-sensitive traffic, while the lowest priority level is treated as best-effort [2].

Although priority can technically be assigned on a per-frame basis, TSN organises frames into streams. A stream represents a continuous flow of related Ethernet traffic, most often tied to a specific application or function. Each stream is assigned a single priority level and all frames within a stream inherit its priority, so that all frames belonging to the same stream experience consistent handling across the network. Since priorities are assigned statically, it is necessary to have prior knowledge of all communication streams expected at runtime, including their bandwidth and latency requirements [2, 10].

Two general aspects of Ethernet communication are relevant to this research, despite them not being specific to TSN frames. Firstly, the total frame size varies depending on

the encapsulated data, as illustrated in Figure 2.1. Secondly, an Interframe Gap (IFG) is inserted between successive frame transmissions. This gap, typically 12 B in length, provides a short idle period that allows network nodes to process the received frame and prepare for the next transmission [1].

2.2.2 TSN Switches

A typical network switch features multiple physical bidirectional ports. Each of these can be logically separated into an ingress and an egress port. When a frame arrives at an ingress port, the switch processes and forwards it toward the next node, based on the frame's destination.

TSN switches are specialised devices that implement the IEEE 802.1Q standard. The forwarding process in a TSN switch is composed of multiple stages, each contributing to ensuring timely and prioritised transmission. These stages will be introduced in reverse order in the following sections, with each step being illustrated by an expanded version of the diagram shown in Figure 2.2.

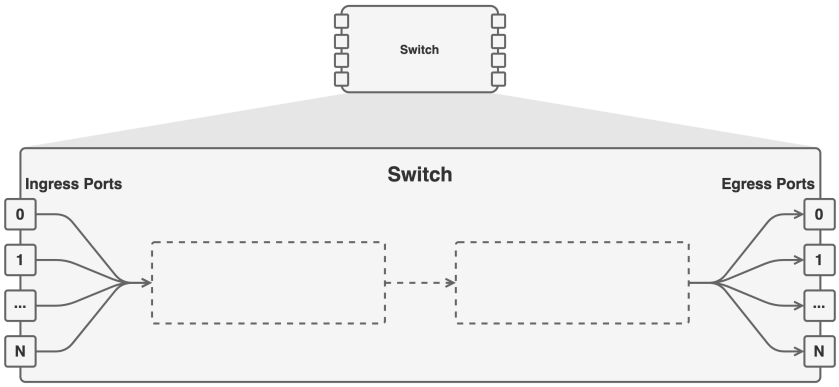


Figure 2.2: The forwarding process of a frame by a TSN switch consists of multiple steps.

2.2.3 Traffic Shaping and Transmission Selection

Transmission Selection: Multiple frames of varying priorities may be buffered simultaneously within a switch. If these frames are to be forwarded to the same node, they must be sequentialised and transmitted one after the other. The transmission order is determined by an algorithm called Strict Priority Queuing (SPQ), in a process known as transmission selection.

As illustrated in Figure 2.3, each egress port maintains a dedicated First In - First Out (FIFO) queue for each priority, resulting in eight so-called TSN queues [2]. Due to their FIFO nature, frames within each queue are transmitted in their order of arrival [2].

When multiple TSN queues contain buffered frames, SPQ selects the head-of-line frame of the highest-priority queue to be forwarded next. This prioritisation process is defined by Section 8.6.8 of IEEE 802.1Q [2].

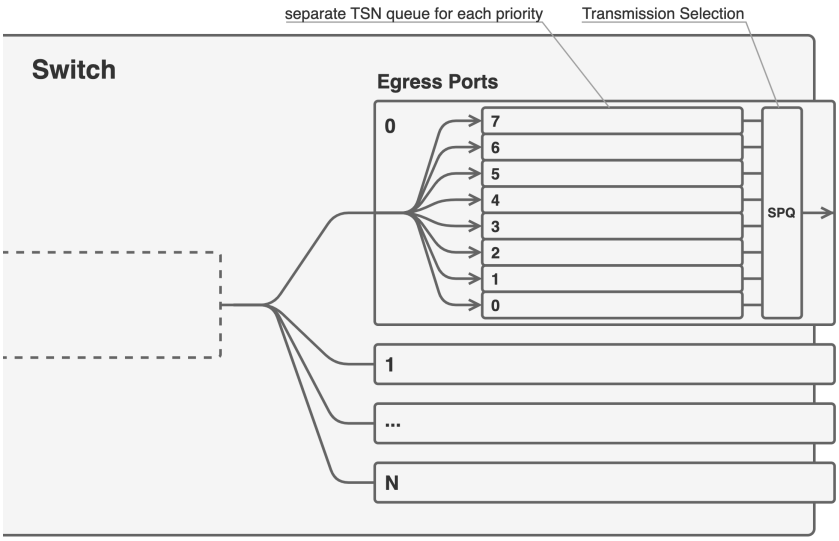


Figure 2.3: Each egress port comprises of one FIFO TSN queue per priority. The transmission selection SPQ selects the next frame to be transmitted from the highest-priority TSN queues.

Traffic Shaping & Transmission Selection Algorithms: When using SPQ, higher-priority traffic can monopolise the available link capacity, potentially preventing lower-priority frames from being transmitted, resulting in starvation. To prevent this, a maximum bandwidth allocation can be set for each priority.

As illustrated in Figure 2.4, each TSN queue can be statically configured with a specific traffic shaping algorithm called Transmission Selection Algorithms (TSAs) which enforces the pre-determined bandwidth constraint. TSAs limit the rate at which consecutive frames are transmitted by temporarily delaying frames that exceed the configured rate and releasing them at controlled intervals. This creates gaps between successive transmissions, smoothing out bursts and enabling lower-priority traffic to make use of the freed-up bandwidth [10].

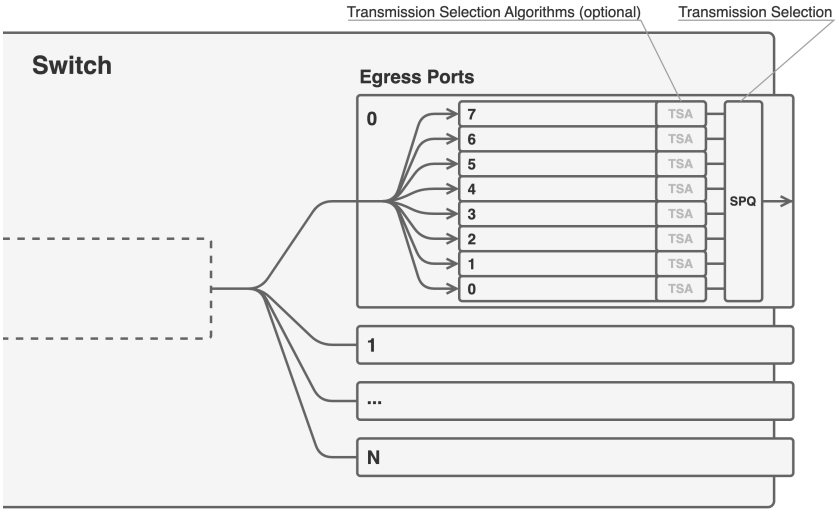


Figure 2.4: Each TSN queue can be configured with a TSA, that enforces a configured traffic rate and determines the queues eligibility for transmission.

To do so, the TSAs work in conjunction with the transmission selection mechanism SPQ described previously. Each TSA associated with a TSN queue signals SPQ whether its queue is eligible for transmission. SPQ then selects the highest-priority eligible queue to transmit the next frame.

By default, a TSN queue is considered eligible as long as it contains at least one frame. However, when a TSA enforces a shaping policy, eligibility becomes conditional. In such cases, the queue is only marked as eligible if transmitting its next frame would not exceed the configured traffic rate. Otherwise, the TSA marks the associated queue as not eligible, temporarily excluding it from transmission selection and delaying its contained frames [2].

Bursts and Source Shaping: As previously noted, traffic shaping not only enforces bandwidth limits but also smooths out sudden bursts of frames, which reduces instantaneous queue build-up. This results in more consistent latencies meaning lower jitter and therefore overall more predictable network behaviour [10].

Because TSN queues often buffer traffic from multiple concurrent streams, a burst from one stream can delay others. Shaping such bursts early benefits all affected streams, making shaping directly at the source generally advisable.

However, source shaping alone is often insufficient. In typical deployments, traffic from multiple sources may arrive concurrently at different ingress ports and converge on a shared egress port. This can create micro-bursts that emerge from uncoordinated arrivals. As a result, in-network reshaping is typically required [10].

2.2.3.1 Credit-Based Shaper Algorithm

The first actively shaping TSA, known as the Credit-Based Shaper algorithm (CBSa), was introduced by the IEEE 802.1Qav amendment from 2009 [11] and has been part of the IEEE 802.1Q standard since the 2018 revision [9].

CBSa is based on a token bucket algorithm and maintains a credit counter, typically expressed in bit. Initially, the credit counter is set to zero. When the associated TSN queue is not empty, the queue becomes eligible for transmission as long as the credit counter is non-negative [2, 12].

At the start of a transmission, the credit counter is decremented by the size of the frame. Since the credit counter initially is zero, this causes the counter to become negative [2].

As long as the counter is negative, or while the ATS queue is not empty, credits accumulate at a constant rate in bit/s. If the queue becomes eligible for transmission but is not selected for transmission by the SPQ, the credit counter continues to increment beyond zero [2].

This mechanism means that if the queue was eligible for transmission a long time before actually being selected for transmission due to higher-priority traffic being transmitted first, the counter may have accumulated enough credit to immediately transmit several frames back-to-back in a burst [2].

Annex L of the IEEE 802.1Q standard analyses the maximum burst size in relation to the characteristics of interfering traffic, demonstrating that it is bounded. However, CBSa does not feature a parameter that allows direct control over this maximum burst size [2].

2.2.3.2 Asynchronous Traffic Shaping

Asynchronous Traffic Shaping (ATS) was introduced to address several limitations of the CBSa. Developed initially by Specht and Samii as the Urgency-Based Scheduler [13], it was renamed to ATS and standardised in IEEE 802.1Qcr-2020 amendment [14, 9]. It is now formally specified in Section 8.6 of the IEEE 802.1Q standard [2].

Unlike the CBSa, which shapes traffic based on TSN queues and thereby entire priority-levels per egress port, ATS enables per-stream shaping. This allows for each stream to be treated differently based on its requirements and expected traffic profiles [2].

A fundamental distinction between CBSa and ATS lies in how they determine when a frame becomes eligible for transmission. In CBSa, eligibility of the entire TSN queue is determined by a non-negative credit counter. Whereas ATS calculates an explicit eligibility time for each individual frame. The eligibility time is a precise timestamp indicating when transmission of that particular frame may begin [2].

To enable this, ATS separates its shaping functionality into two core components [2]:

- **ATS schedulers** operate on a per-stream basis, computing the eligibility time for each incoming frame. The timestamp is stored in a node-local eligibilityTimeTag, which is stripped before the frame leaves the node.
- **ATS TSAs** are queue-level mechanisms that enforce these eligibility times. ATS TSAs ensure that the frames of their TSN queues are forwarded once the eligibility times have been reached. When a frame reaches its eligibility time, it is marked as eligible for SPQ.

As depicted in Figure 2.5, ATS schedulers are positioned before the egress ports and are decoupled from the TSN queues. Thus, multiple streams scheduled by separate ATS schedulers can share the same TSN queue [2].

TSN queues configured with ATS TSAs are called ATS queues. They are sorted based on the assigned eligibility times, so the head-of-queue frame is always the one becoming eligible next [2].

ATS schedulers utilise credit counters with credits measured in bit. In ATS, the constant rate at which credits accumulate is called CommittedInformationRate (CIR) in bit/s. Unlike it was the case with the CBSa, ATS allows the specification of a maximum limit

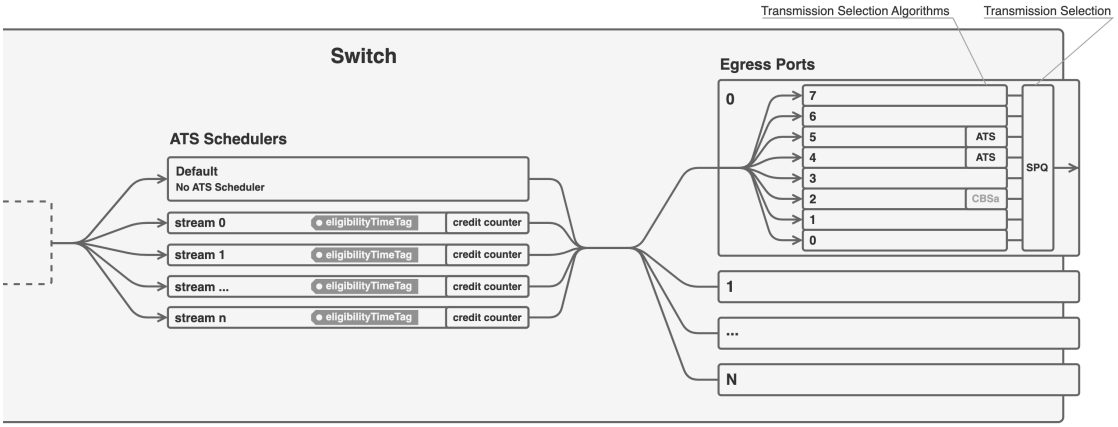


Figure 2.5: Schematic view of a TSN switch with multiple ATS schedulers. Two TSN queues feature the ATS TSA. Each TSN queue can feature a different TSA, as indicated by the CBSa configured for the TSN queue of priority 2.

for the credits. This limit is called CommittedBurstSize (CBS) and is also measured in bit. The credit counter is initialised at this positive limit [2].

Three rules of operation apply to ATS schedulers [2]:

- credits accumulate at the CIR up to the CBS
- a frame becomes eligible for transmission once the credit counter is at least equal to the length of the frame
- when a frame becomes eligible for transmission, the credit counter is decremented by the frame's length in bit

The impact of the CBS mechanism is multifaceted. Firstly, because the credit value is bounded and cannot exceed this defined maximum, it must be at least equal to the length of the largest frame subject to ATS scheduling, in order to ensure that the scheduler can eventually accumulate sufficient credits to transmit even the largest admissible frame.

Secondly, the CBS enables precise control over the maximum burst size, independent of cross-traffic. For streams with short bursts, a low CBS may introduce undue delay, as it forces the frames to be spaced further apart in time. Conversely, a higher limit permits short bursts of consecutive frames, improving responsiveness at the cost of potential short-term interference on other streams. Importantly, increasing the CBS does not violate the long-term bandwidth constraint defined by the CIR. Rather, it allows temporary bursts within a controlled envelope, followed by idle periods to maintain compliance.

Figure 2.6 shows an example of the ATS shaping process. The first frame that arrives becomes immediately eligible for transmission and is transmitted to the next node. The second frame arrives after the credits have replenished. The third frame is not delayed since there are still enough credits available for the transmission. A fourth frame from another stream, illustrated in grey, arrives and is transmitted. This causes the fifth frame to be delayed and selected for transmission a bit later than per its eligibility time.

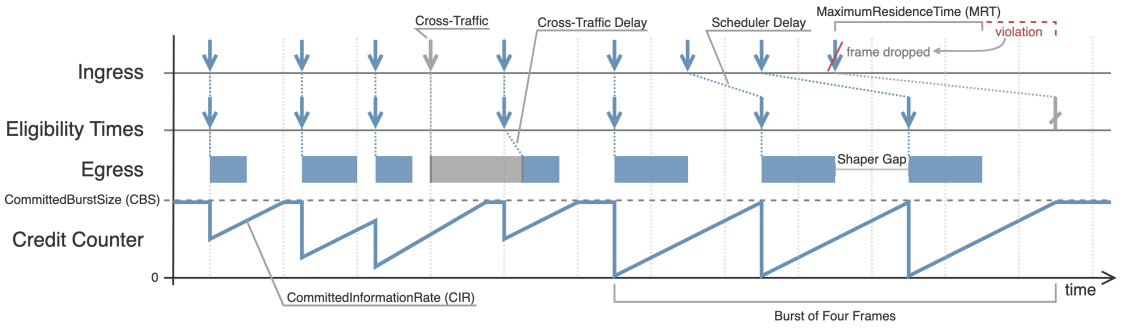


Figure 2.6: Behaviour of ATS with cross-traffic and an MRT violation.

The last four frames of Figure 2.6 are transmitted in quick succession, exceeding the enforced transmission rate over a prolonged period of time. Initially, the first frame of the four is forwarded without delay. The second is briefly delayed and the third faces an even longer delay. As the backlog grows, subsequent frames are assigned eligibility times further into the future.

To limit the delay introduced by the ATS scheduler, a third parameter called MaximumResidenceTime (MRT) can be used. On the contrary to what the name might suggest, MaximumResidenceTime (MRT) does not cap the total time a frame may reside in a node, but simply limits the maximum additional delay that ATS is allowed to introduce to a frame [2].

In the example, the calculated eligibility time for the last frame exceeds $T_0 + \text{MRT}$, where T_0 is the frame's arrival time. Therefore the frame is discarded and not queued.

This way, the MRT effectively prevents buffer overflows. Furthermore, as discussed in Section 2.1, in some types of real-time communication, frames that exceed their deadlines are no longer useful. By discarding such frames early, unnecessary forwarding is avoided and resources are preserved for traffic that can still meet its timing requirements.

ATS Scheduler Groups and Group Eligibility Times: For ATS schedulers, the IEEE 802.1Q standard defines a concept called ATS scheduler groups to facilitate efficient hardware-based sorting [2, 15]. Ideally, a sorting hardware component only has to compare the head elements of multiple FIFO queues. This is exactly what SPQ does with the individual TSN queues: While frames can overtake frames of other priorities by being selected first, they cannot overtake frames of the same priority, since they reside in the same FIFO queue.

Based on the ATS scheduling algorithm presented above, any frame could theoretically overtake any other frame, as long as they belong to different streams. As the number of streams is not limited, there would have to be an infinite number of FIFO queues for hardware-based sorting of this kind to be possible.

To address this, streams with the same ingress port and priority, or more specifically their ATS schedulers, are grouped into a single ATS scheduler group, as shown in Figure 2.7. Within each group, frame reordering is not allowed. This policy is enforced using a shared group variable called group eligibility time, which holds the latest assigned eligibility time by any ATS scheduler within the ATS scheduler group. For each frame, an ATS scheduler must assign the maximum of the calculated eligibility time and this shared value as the eligibility time to the frame, ensuring consistent ordering. After assignment of an eligibility time higher than the current group eligibility time, the ATS scheduler must update the shared variable accordingly [2].

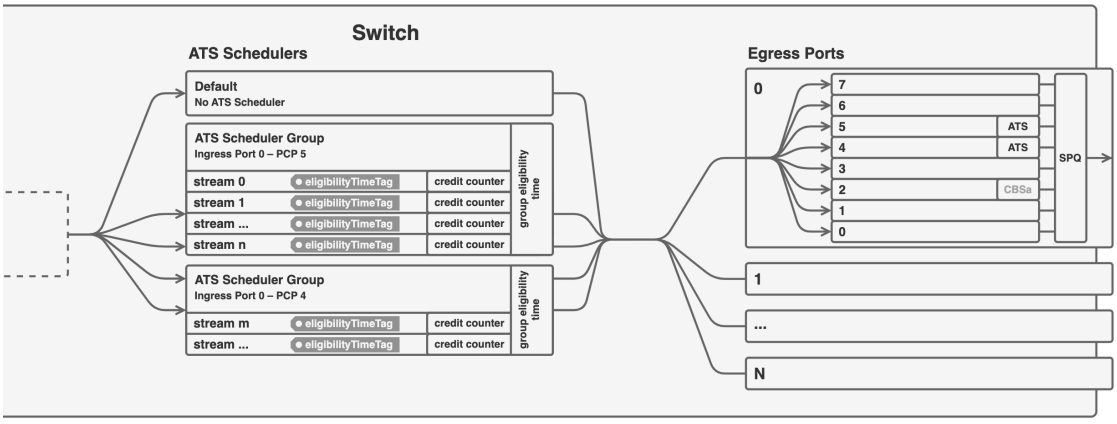


Figure 2.7: ATS scheduler groups consolidate all ATS schedulers assigned to streams that share a common ingress port and priority. The ATS schedulers are constrained by a shared group eligibility time.

Summary of ATS: To bring it all together, when a frame arrives at its ATS scheduler instance, the scheduler calculates the local system time when it will have accumulated the amount of credits needed to forward the frame. This calculation is based on the current number of credits held by the scheduler, its accumulation rate CIR and the frame's length.

The ATS scheduler then compares the calculated eligibility time against its group eligibility time, choosing the higher value as the actual eligibility time. If that value exceeds the MRT, the frame is discarded. If not, a corresponding eligibilityTimeTag is added to the frame and the group eligibility time is updated if necessary.

Later on when buffered in the TSN queue, the frame is blocked by the ATS TSA until its eligibility time is reached, at which point it becomes eligible for transmission. The frame may not be forwarded instantly upon becoming eligible but rather experience further delay due to cross-traffic.

2.2.4 Stream Identification

As established in the previous section, each stream may be handled differently within a TSN node. To determine the appropriate processing, each arriving frame undergoes stream identification, which aims to identify the stream to which it belongs [2]. This process is illustrated in Figure 2.8.

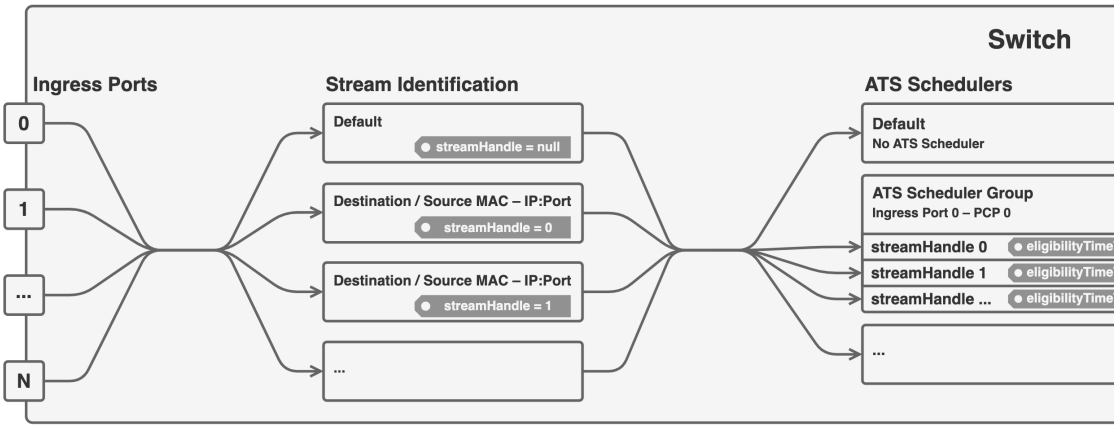


Figure 2.8: Stream identification maps frames to stream handles based on fields such as source, destination, MAC and IP addresses, and port numbers. If no rule matches, an empty stream handle is assigned. ATS schedulers are assigned to handle frames based on the stream handles.

The IEEE 802.1Q standard specifies which fields may be used for stream identification, including the frame's source and destination MAC and IP addresses, along with transport-layer port numbers. These fields are matched against predefined rules to assign a corresponding stream handle. If no rule applies, an empty handle is assigned. Importantly, the PCP field is not permitted for stream identification [2].

Processing decisions are based on the stream handle. For ATS schedulers in particular, the combination of stream handle and priority is used for the mapping of frames to schedulers [2].

This reliance on preconfigured mapping rules and scheduler assignments underscores the need for all processed streams to be statically defined, as previously stated in Section 2.2.1.

2.2.5 Forwarding Process Example

Now that all fundamental concepts have been introduced, the complete frame forwarding process can be illustrated by an example. The configuration of the switch used for this example is illustrated in Figure 2.9. The switch is configured to identify four distinct streams from four different sources, tagging each with a unique stream handle. Of these, the streams associated with stream handles 2 and 3 both are priority 4 streams and are assigned an ATS scheduler, while the remaining two are not. Specifically, the stream with stream handle 0 corresponds to unshaped best-effort traffic at priority 0 and the stream with stream handle 1 is priority 3 traffic and therefore shaped by CBSa in this configuration.

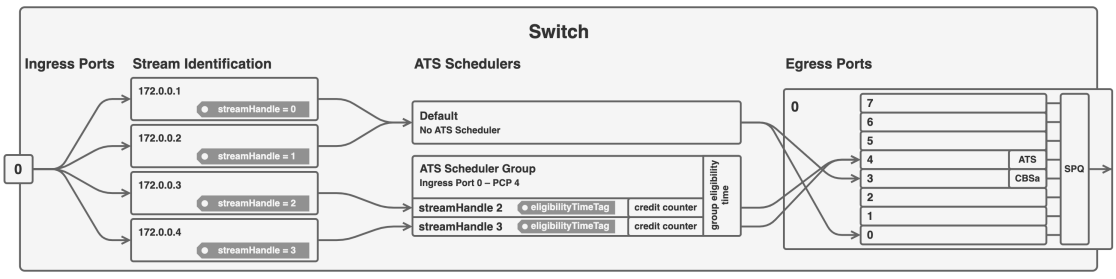


Figure 2.9: Schematic view of a switch configured to identify four streams, two of which are priority 4 and are subject to ATS shaping, one is priority 3, which is CBSa shaped and a last one is unshaped in priority 0.

Five frames arrive at the switch with the associated stream handles being 3, 3, 4, 0, 1.

The first arriving frame (Frame 0) is tagged with stream handle 3 and passed to its corresponding ATS scheduler. As the first frame of its stream, it is marked immediately eligible and assigned an eligibilityTimeTag equal to the current system time. The scheduler updates the shared group eligibility time, and the frame is enqueued in the ATS queue for priority 4. Being eligible, it is selected by SPQ for transmission.

While Frame 0 is transmitting, two more frames arrive. Frame 1 also belongs to stream 3 and is handled by the same ATS scheduler. However, due to the recent transmission of frame 0, the scheduler's credit counter has not yet replenished. Therefore, Frame 1 is temporarily ineligible. Frame 2, from a different stream with stream handle 4, is assigned to the other ATS scheduler that is within the same ATS scheduler group. It is delayed as well despite being the first frame of that stream, due to the shared group eligibility time, becoming eligible after Frame 1 to maintain order. Both frames are queued in the same TSN queue.

After Frame 0 completes transmission, neither of the ATS-shaped frames is yet eligible.

Meanwhile, Frame 3, a best-effort stream handle 0 frame arrives. As no ATS scheduler is configured for it, it is enqueued directly into the TSN queue for priority 0 without an eligibilityTimeTag. Since no other frame is eligible, it is selected for transmission.

During its transmission, Frame 4 arrives from the stream with stream handle 1. It is queued under priority 3 and therefore subject to CBSa shaping. Since this is the first frame of its stream, the TSN queue becomes immediately eligible, but must wait for the completion of Frame 3.

Once the best-effort transmission ends, and Frames 1 and 2 become eligible, they are transmitted next in order, followed by the lower-priority Frame 4, assuming no further arrivals.

3 Problem Statement & Literature Review

To reiterate on the concepts introduced in the previous chapter, any two frames of different streams will be enqueued in the same TSN queue if they share the same priority and contend for the same egress port of a TSN node. The latter occurs when both are transmitted over the same physical link, in other words, when they share the same next hop.

Figure 3.1 illustrates a simple TSN network with two streams of priority 4, each emitted by a different source and routed through an intermediate switch to a common sink. Consequently, all corresponding frames are enqueued in the same TSN queue, which is governed by an ATS TSA, as depicted.

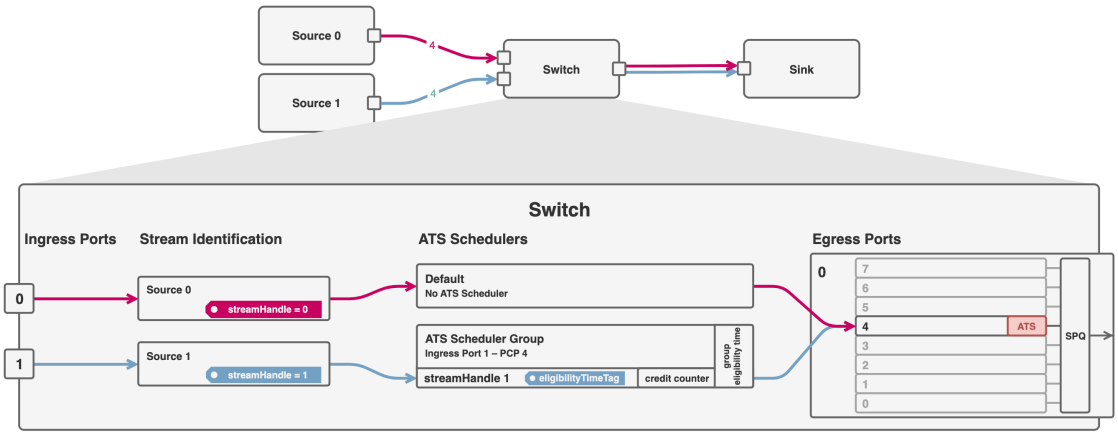


Figure 3.1: The upper part of this figure shows a simple network topology with three TSN nodes. Two streams originate from separate sources and are forwarded to a shared sink via an intermediate switch. The lower part illustrates the logical internal configuration of the TSN switch. The blue stream frames from source 1 are scheduled by an ATS scheduler, while the pink stream frames from source 0 are not. Since both streams share the same priority, this configuration results in a Non-ATS Conflict.

As explained in Chapter 2, an ATS queue requires all frames to carry an eligibility-TimeTag, assigned by an ATS scheduler. These tags are used to sort the queue and determine when a frame becomes eligible for transmission. Accordingly, all streams feeding into an ATS queue must be ATS scheduled.

However, as shown in Figure 3.1, the pink stream emitted by source 0 bypasses the ATS scheduler and is routed along the default path. Its frames lack an eligibilityTimeTag and are therefore incompatible with the ATS queue into which they are enqueued. As a result, they cannot be properly sorted, nor can their eligibility be assessed by the ATS TSA. This situation is referred to as a Non-ATS Conflict.

This issue could be avoided if such configurations were deemed invalid by the IEEE 802.1Q standard. However, the standard neither forbids nor safeguards against them [2]. This leads to ambiguity and may result in inconsistent or unpredictable system behaviour which might constitute in system failure. Given the critical nature of many TSN use cases, such uncertainty is likely intolerable, as previously discussed in Chapter 2.

In general terms, a Non-ATS Conflict occurs whenever unscheduled frames are directed to an ATS queue. Currently, this poses a significant, yet implicit, design constraint in TSN networks: Streams sharing a priority level and egress port must either all be ATS scheduled or none of them can be.

This thesis aims to develop and evaluate strategies that address this constraint and combine ATS and Non-ATS traffic within shared TSN queues. Through performance evaluation under controlled conditions, the goal is to provide insights into the behaviour, limitations, and practical viability of these approaches. As the reliable deployment of TSN-based communication in safety-critical applications hinges on strict latency guarantees, this work also investigates whether such guarantees can be upheld or are compromised under the proposed strategies.

As of May 2025, no prior study has examined this specific topic. Although works such as [16] derive latency bounds for networks employing multiple traffic shaping algorithms, none address the definitional gap in IEEE 802.1Q. This thesis introduces foundational concepts to close that gap and to support future research on handling unscheduled traffic contending for ATS queues.

4 Concepts

This chapter extends the established TSN terminology and introduces a range of strategies aimed at mitigating the Non-ATS Conflict identified in Chapter 3. Each strategy is briefly outlined to convey its fundamental behavior and design rationale. Out of the nine proposed strategies, three are selected for in-depth evaluation. Before introducing the strategies themselves, the concept of Non-ATS is clarified.

4.1 Non-ATS

In this thesis, a stream is referred to as a Non-ATS stream if it does not pass through an ATS scheduler and contends for an ATS queue that is used by at least one ATS-shaped stream.

The second condition is important. If the Non-ATS stream were the sole contender for the ATS queue, the Non-ATS Conflict could be trivially resolved by removing the ATS TSA from the queue without consequence.

By this definition, a Non-ATS stream can only exist if there is at least one other ATS-shaped stream in the network with the same priority, sharing at least one common hop. Consequently, in standard TSN deployments, the presence of Non-ATS stream inevitably leads to a Non-ATS Conflict.

4.2 Non-ATS Handling Strategies

As illustrated in Figure 4.1, the Non-ATS Conflict can be addressed in one of three general approaches:

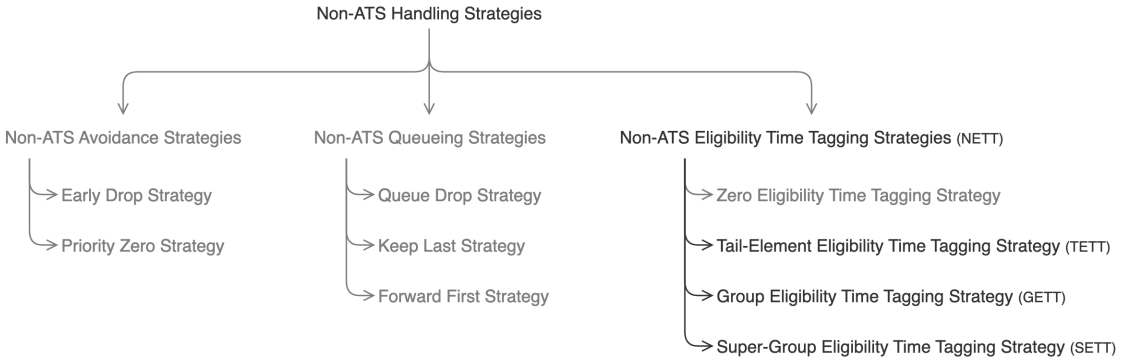


Figure 4.1: Categorised strategies for dealing with Non-ATS traffic in TSN nodes.

1. **Non-ATS Avoidance strategies** aim to prevent the conflict entirely by ensuring that unscheduled frames are never enqueued in ATS queues.
2. **Non-ATS Queuing strategies** propose enhancements to the ATS queues themselves, allowing them to process both ATS scheduled and Non-ATS frames simultaneously.
3. **Non-ATS Eligibility Time Tagging strategies** focus on transforming incoming Non-ATS frames so that they are compatible with the existing, unmodified ATS queues. This involves assigning synthetic or inferred eligibilityTimeTags to otherwise unscheduled frames.

All strategies inherently involve a critical trade-off between two competing objectives: Minimising the impact on the delay of concurrent streams and integrating Non-ATS traffic. In other words, a strategy may either prioritise protecting other traffic from interference caused by Non-ATSs traffic, or focus on ensuring that Non-ATS frames can still be processed in a controlled and predictable manner, without violating latency guarantees or introducing erratic behaviour. This trade-off is not binary. It exists on a continuum, allowing for a wide range of design variations depending on the specific performance goals and system constraints.

4.2.1 Non-ATS Avoidance Strategies

Non-ATS Avoidance strategies are generally the simplest to implement, as they leverage existing mechanisms defined by IEEE 802.1Q to prevent Non-ATS frames from reaching their intended ATS queue. Since these strategies do not aim to combine ATS and Non-ATS traffic in TSN queues, they will be presented briefly but not analysed any further in this thesis.

4.2.1.1 Early Drop Strategy

The Early Drop strategy is among the simplest approaches to preventing the Non-ATS Conflicts, with minimal impact on concurrent streams. Its conceptual idea is to discard Non-ATS frames before they are enqueued in the ATS queue.

Several standardised mechanisms can serve as stream filters fit for this purpose, most notably Per Stream Filtering and Policing (PSFP) as defined in Section 8.6.5.2 of IEEE 802.1Q, introduced in 2017 [17]. In its essence, PSFP enables filtering based on stream handles and priorities, allowing frames identified as part of a Non-ATS stream to be dropped systematically [2].

However, since this strategy unconditionally prevents all Non-ATS frames from reaching their intended destinations, it is unsuitable in scenarios where such traffic is expected to be delivered. Instead, its primary utility lies in improving network resilience by mitigating the impact of misconfigurations that would otherwise result in undefined system behaviour. In such contexts, a robust logging mechanism is strongly recommended, as silently discarding frames is considered poor practice from both an operational and diagnostic standpoint. Misconfigurations and resilience are beyond the scope of this thesis.

It is also worth noting that PSFP is an optional feature of the standard and may not be supported by all network hardware. In such cases, the Early Drop strategy may not be technically feasible [2].

4.2.1.2 Priority Zero Strategy

The Priority Zero strategy temporarily alters the priority of Non-ATS frames: During the forwarding process, these frames are assigned priority zero. Their original priority is re-

stored before transmission to the next node. This approach leverages the optional concept of Internal Priority Values (IPVs), as defined in Section 8.6.5.4 of IEEE 802.1Q [17].

According to the standard, priority zero is intended for unshaped, best-effort traffic [2]. If this convention is respected within a given network configuration, assigning priority zero prevents Non-ATS frames from being enqueued in ATS queues. However, this effectively downgrades the QoS guarantees of the affected Non-ATS stream, as its original priority is not maintained along the entire path.

Although the strategy could theoretically be adapted to assign a different, non-zero priority, legitimate use cases might be limited. In scenarios where temporary downgrading is acceptable, it is generally more advisable to permanently reclassify the Non-ATS stream at the source, simplifying the configuration which makes the system easier to maintain.

4.2.2 Non-ATS Queuing Strategies

The second category includes Non-ATS Queuing strategies. These strategies modify the ATS TSA or other components of the TSN queue to accommodate frames that lack an eligibilityTimeTag. The resulting TSN queue is capable of processing both ATS and Non-ATS frames. Therefore, it can be referred to as a shared TSN queues.

As discussed earlier in Chapter 2, queue sorting is carried out by hardware for efficiency reasons. Consequently, altering queuing behaviour will most likely require a substantial hardware change.

Just like the previous category, the Non-ATS Queuing strategies will be presented briefly but will not be further analysed in the proceedings of this thesis.

4.2.2.1 Queue Drop Strategy

The Queue Drop strategy is similar to the Early Drop strategy, but Non-ATS frames are dropped later in the forwarding process by the shared TSN queue itself. In comparison to the Early Drop strategy, this avoids the dependency on the optional PSFP feature. However, it does require the use of a non-standard shared TSN queue, which may not be feasible either.

4.2.2.2 Keep Last Strategy

The Keep Last strategy prioritises ATS frames, keeping Non-ATS frames behind them in the shared TSN queue at all times. This ensures that Non-ATS frames are only eligible for transmission when the shared TSN queue is free of ATS frames.

While this approach helps to preserve low delay requirements of ATS frames of the same priority, it may significantly impact lower-priority traffic that shares the same egress port. This happens because Non-ATS frames accumulate at the back of the shared TSN queue as long as any ATS frame is buffered in it. Once the last ATS frame is transmitted, the accumulated Non-ATS traffic can momentarily consume the available bandwidth, since it is not rate limited by a scheduler. While frames of the same or higher priority than the Non-ATS stream can overtake Non-ATS frames, lower-priority traffic may experience significant delays or even starvation.

4.2.2.3 Forward First Strategy

The Forward First strategy inverts the behaviour of the Keep Last strategy by prioritising Non-ATS frames over ATS frames. In this approach, Non-ATS frames are added to the front of the shared TSN queue and transmitted ahead of any ATS traffic. This ensures that Non-ATS frames are forwarded with minimal delay, avoiding any build-up behind queued ATS frames.

However, this prioritisation entails certain trade-offs. Depending on the characteristics of the Non-ATS traffic, it may introduce significant additional delay for frames of equal or lower priority that traverse the same egress port. Such delays have the potential to compromise the deterministic behaviour expected of shaped traffic. Nonetheless, this approach may remain acceptable for Non-ATS streams with low traffic volume.

4.2.3 Non-ATS Eligibility Time Tagging Strategies

The final category of Non-ATS Handling strategies involves Non-ATS Eligibility Time Tagging (NETT) strategies. These tag Non-ATS frames with eligibilityTimeTags to allow them to work with standard ATS queues. Despite not being modified, the involved ATS queues are capable of handling Non-ATS traffic and can therefore also be referred to as shared TSN queue.

The key difference between employing these strategies and standard ATS shaping is the absence of an ATS scheduler for the Non-ATS streams. Instead of using a statically configured ATS scheduling algorithm to calculate eligibility times, the eligibility time of Non-ATS frames is determined dynamically by concurrent ATS traffic in the TSN node.

However, some NETT strategies leverage algorithms that are closely related to ATS scheduling algorithms, allowing standard ATS schedulers to be adapted for their implementation. The complexity of incorporating these strategies varies depending on the nature of the ATS schedulers. For software-based ATS schedulers, the necessary modifications are generally straight forward. In contrast, for hardware-based implementations, the extent of required changes depends on the specific NETT strategy and may range from minor adjustments to more substantial modifications. Detailed discussions of these considerations will be provided in the subsequent sections.

The approach of assigning eligibilityTimeTags to Non-ATS frames provides finer control over their eligibility times compared to other Non-ATS Handling strategies. Since the eligibility times are inferred based on concurrent ATS traffic and active schedulers, multiple Non-ATS frames can be spaced out in time, depending on various factors discussed later on. This temporal distribution helps alleviate the problem of frame accumulation at the tail of shared TSN queues, that was identified for the Keep Last strategy.

4.2.3.1 Zero Eligibility Time Tagging

The Zero Eligibility Time Tagging strategy effectively replicates the behaviour of the Forward First Non-ATS Queuing strategy discussed in Section 4.2.2.3. By assigning an eligibilityTimeTag of zero to all Non-ATS frames, these frames are always sorted to the front of the shared TSN queue and become eligible for transmission immediately upon arrival. Adapting both software- and hardware-based ATS schedulers to implement this tagging behaviour is trivial.

At first glance, this strategy might seem equivalent to configuring an ATS scheduler with an effectively infinite CIR and CBS. However, this is not the case. An ATS scheduler would assign the current system time rather than zero and would still be constrained by the group eligibility time. For the scheduler to behave as intended, it would need to bypass this constraint entirely.

This presents a further implementation challenge: Ignoring the group eligibility time effectively introduces a new ATS scheduler group capable of overtaking all others. Supporting such a group would likely require hardware modifications, as discussed in Section 2.2.3.2.

Furthermore, due to its equivalence to the Forward First strategy, this approach shares the same key limitation: it lacks any form of rate limiting for Non-ATS traffic. As a result, a misbehaving Non-ATS stream can impose unbounded interference on concurrent streams.

Given its functional similarity to a Non-ATS Queuing strategy, implementational complexity and relatively simple behaviour, the Zero Eligibility Time Tagging strategy will not be considered further in this thesis.

4.2.3.2 Tail Element Eligibility Time Tagging

The TETT strategy assigns each Non-ATS frame the same eligibilityTimeTag as the current tail-element of the shared TSN queue into which it will be enqueued. If the shared TSN queue is empty, the tag is set to the arrival time of the frame. Formally, TETT is defined as:

$$\text{eligibilityTime}_{\text{TETT}}(a_f, q_f) = \begin{cases} \text{eligibilityTime}_{\text{tail}}(q_f, a_f), & \text{if } q_f \neq \emptyset \\ a_f, & \text{otherwise} \end{cases} \tag{4.1}$$

with

- f: the frame being tagged
- a_f: the local system time of arrival of f
- q_f: the TSN queue that f will be enqueued into
- $\text{eligibilityTime}_{\text{tail}}(q_f, a_f)$: eligibility time of the tail frame in q_f at time a_f

Unlike Zero Eligibility Time Tagging, TETT passively limits the transmission rate of Non-ATS frames based on the presence and timing of ATS frames in the same shared TSN queue. As Non-ATS frames are distributed behind multiple ATS frames in the queue, the likelihood of excessive build-up behind a single ATS frame is reduced, especially under conditions of steady ATS traffic.

From an implementation standpoint, however, TETT presents significant challenges. Current ATS scheduler implementations, whether hardware- or software-based, are not inherently designed to interface directly with TSN queues. Enabling such access constitutes a pre-requisite for implementing TETT, thereby complicating the adaptation of existing ATS scheduler architectures.

4.2.3.3 Group Eligibility Time Tagging

In the GETT strategy, each Non-ATS frame is tagged with the current group eligibility time of the ATS scheduler group corresponding to its ingress port and priority. If the frame's arrival time is later than this group eligibility time, the arrival time is used instead. Formally:

$$\text{eligibilityTime}_{\text{GETT}}(a_f, g_f) = \max(\text{eligibilityTime}_{\text{group}}(g_f, a_f), a_f) \tag{4.2}$$

with

- f: the frame being tagged
- a_f: the local system time of arrival of f
- g_f: the ATS scheduler group associated with the ingress port and priority of f
- $\text{eligibilityTime}_{\text{group}}(g_f, a_f)$: the group eligibility time of g_f at time a_f

Similar to TETT, the GETT method enables indirect shaping of Non-ATS frames through concurrent ATS traffic. This time, the influence is limited to all ATS schedulers sharing the same ATS scheduler group.

An advantage of GETT over TETT lies in its ease of implementation: GETT can be emulated by appropriately configuring a standard ATS scheduler, unlike it was the case with Zero Eligibility Time Tagging strategy. To emulate GETT behaviour, the ATS scheduler must be configured with effectively infinite CIR and CBS values. If such parameterisation is not possible, the scheduler can be simplified by removing the credit-based calculation logic, retaining only the comparison of a frame's arrival time against the group eligibility time. As a result, both software- and hardware-based schedulers require no or only minimal modifications to support GETT.

The difference between GETT and TETT is that with GETT, a Non-ATS frames might be inserted before a currently queued ATS frame of the same priority, as long as they are part of different ATS scheduler groups. With TETT this is not possible, as the Non-ATS frames are always enqueued last.

4.2.3.4 Super-Group Eligibility Time Tagging

The SETT strategy extends the concept of GETT by using the maximum group eligibility time among all ATS scheduler groups within a TSN node, referred to as the super-group eligibility time. Formally:

$$\text{eligibilityTime}_{\text{SETT}}(a_f) = \max\left(\bigcup_{g \in G} \text{eligibilityTime}_{\text{group}}(g, a_f) \cup \{a_f\}\right) \tag{4.3}$$

with

- f: the frame being tagged
- a_f: the local system time of arrival of f
- G: the set of all ATS scheduler groups in the TSN node
- $\text{eligibilityTime}_{\text{group}}(g, a_f)$: the group eligibility time of g at time a_f

From an implementation perspective, SETT requires the scheduler to retrieve the maximum group eligibility time (super-group eligibility time) from the ATS scheduler group instance table, that stores all group eligibility times as defined in Section 8.6.5.6 of IEEE 802.1Q [2]. Unlike TETT, this modification is relatively modest and should not fundamentally alter the scheduler's architecture.

Like the previous two strategies, SETT enables indirect shaping of Non-ATS frames, but now influenced by all ATS schedulers on the node. It merges the characteristics of GETT and TETT in that it ensures each Non-ATS frame is enqueued at the very back of the TSN queue, just like TETT does, while offering similar simplicity in terms of implementation as GETT.

Unlike the Keep Last strategy, SETT allows the Non-ATS frames to become eligible for transmission over time instead of keeping them stuck behind any queued ATS frame.

4.3 Selected Strategies for Evaluation

In conclusion, among the nine methods introduced in the preceding chapter, the TETT, GETT and SETT strategies have been selected for detailed evaluation in the subsequent chapters. These strategies were chosen based on their potential to mitigate the Non-ATS Conflict. For clarity, the selected NETT strategies are as follows:

- **TETT:** Aligns Non-ATS frames' eligibility times with the tail of the respective target TSN queue.
- **GETT:** Uses the current group eligibility time of the associated ATS scheduler group.
- **SETT:** Assignes the maximum group eligibility time across all ATS scheduler groups in the TSN node.

5 Methodology

This chapter presents an overview of the OMNeT++ simulation framework employed for modelling network scenarios and data collection [18]. It details the implementation of the NETT strategies within OMNeT++, along with the quality assurance measures applied to ensure the reliability and validity of the simulation results. The chapter then outlines the benchmarking methodology and discusses the design rationale behind the selection of specific network topologies used for evaluation. Key performance metrics, derived from the problem statement, are introduced and their significance to the research objectives is explained. Additionally, the configuration parameters of the various scenarios are examined in terms of their potential influence on network performance. Finally, a custom-developed software tool for automated data collection and analysis is presented.

5.1 OMNeT++ Simulation Framework

Simulations provide a cost-effective and flexible alternative to physical testbeds, allowing for extensive testing under controlled conditions while avoiding the complexities and expenses associated with hardware-based evaluations. They are particularly useful for conducting parameter studies and isolating the effects of specific variables, making them the most suitable evaluation strategy for this thesis.

OMNeT++ is an open-source discrete event simulation framework, widely used for network modelling and analysis [18]. Its modular architecture and extensive parameterised component library enables the creation of network configurations that closely resemble real-world scenarios, but also supports the creation of non-standardised models implemented for research, making it an ideal choice for evaluating new strategies.

The INET framework extends OMNeT++ by implementing a comprehensive set of communication standards including wired and wireless Data Link layer protocols and more [19].

CoRE4INET further enhances the capabilities of INET by implementing the IEEE 802.1Q standard [20]. As of INET version 4.4, many CoRE4INET features have been integrated, providing substantial support for TSN modelling and rendering CoRE4INET largely obsolete for this thesis. However, as of the latest version (v4.5.4), INET does not yet support ATS scheduler groups. The required components were implemented by T. Lübeck from the CoRE research group [21] in May 2024 [22] and are expected to be integrated into future INET releases. These components are used in the experiments conducted in this thesis.

OMNeT++ provides repeatable deterministic simulations with picosecond resolution. Events and metrics generated during simulation runs are recorded in result files, which, for example, may contain all changes in a specific queue's length of a network node or the exact latency experienced by any frame that was sent. All events and calculated metrics are captured using signals, which are implemented within OMNeT++ modules. By incorporating custom signals into existing or new modules, user-specific metrics can be recorded, enabling detailed analysis of particular mechanisms [23].

5.2 Implementation of the NETT Strategies in OMNeT++

The implementation of the NETT strategies requires modifications to be done to several existing OMNeT++/INET modules, as well as the development of new components. The necessary changes and additions are outlined in the following paragraphs.

As explained in the previous chapter, GETT and SETT strategies can be realised by modifying the existing ATS scheduler module, specifically the `inet.protocolelement.shaper.GroupEligibilityTimeMeter`, as they are very similar from an implementational stand point [22].

Implementing the GETT strategy is relatively simple. The calculation of the eligibility time has to be changed to match Equation 4.2, as discussed in Section 4.2.3.3.

The modifications needed for realising SETT are similar but a bit more involved. The module `inet.protocolelement.shaper.GroupEligibilityTimeTable` [22], responsible for managing group eligibility times across all ATS schedulers, was extended to support access to the super-group eligibility time. The former ATS scheduler module was then configured to use this enhanced table and attach an eligibilityTimeTag to each processed frame, in accordance with Equation 4.3.

Implementing the TETT mechanism requires a different approach. Rather than using a scheduler-like module, which would pose challenges within OMNeT++'s architecture, a more practical solution is introducing a filter module placed before the shared TSN queue. This filter checks each incoming frame for the presence of an eligibilityTimeTag. If the tag is missing and the shared TSN queue is not empty, the filter retrieves and copies the eligibilityTimeTag of the tail frame in the shared TSN queue to tag the Non-ATS frame with it, in accordance with Equation 4.1.

Additionally, the `inet.queueing.queue.PacketQueue` module needs to be extended to emit two custom signals required for delay analysis. The metrics recorded from these signals are discussed in Section 5.7.5.

Further modifications are necessary to integrate the changes into the existing module structure. These technicalities will not be discussed further.

5.3 Quality Assurance

To ensure the validity of the results generated by the simulations, it is essential to verify that the custom modules operate as intended.

5.3.1 Verification and Validation

In the context of testing, two types of assessment are commonly considered: verification and validation. Verification is the process of ensuring that an implementation adheres to its specification. It is primarily a concern of software engineering and can be supported by automated testing tools for software quality assurance. Validation, by contrast, takes a broader perspective, asking whether the correct system has been built, that is, whether the system's behaviour aligns with its intended purpose. This process requires domain knowledge and critical thinking and is therefore a largely manual process [23].

That said, any requirements derived from the problem domain should be tested automatically to verify their implementation. Since the requirements and modifications introduced in this thesis are relatively straight forward, successful verification also yields a strong confidence in the system's validity.

5.3.2 Unit Tests

Most testing procedures rely on unit tests, which are undoubtedly effective for verifying individual software components. However, as noted in the OMNeT++ manual, applying unit tests in the context of OMNeT++ modules is challenging due to their tight integration into the simulation framework. Individual modules typically cannot be instantiated or operated in isolation, as they respond to a variety of events and maintain complex internal states [23].

Moreover, unit tests are particularly useful for detecting implementation errors early in order to be able to trace bugs to their source easily. In this project, however, only minor modifications were made to an otherwise well-established simulation system, as covered in Section 5.2. As a result, any deviation from expected behaviour can be directly attributed to the introduced changes without further debugging effort. For these reasons, unit testing was deemed unnecessary, and comprehensive system tests were considered sufficient.

5.3.3 System Tests

System tests treat the complete simulation process as a black box. Typically, they operate by supplying known inputs and verifying that the resulting outputs conform to expected behaviour.

For this thesis, a dedicated system test suite was implemented in python [24]. Since the suite was created early in the development phase, it was essential to design it in a way that would not need to be altered, despite significant changes to the simulation code. This was achieved by structuring the tests around the result files produced by OMNeT++ simulations, rather than relying on internal implementation details.

The system test suite automatically executes any simulation specified in a given OMNeT++ network configuration file. After completion, the generated results are imported automatically so that they can be used to verify the correctness of the assigned eligibility times for each Non-ATS frame.

For every enqueued frame, the results contain the information of the associated stream, arrival time, current TSN queue length and the assigned eligibility time. The system test

logic reflects the expected behaviour of the different NETT strategies described in Chapter 4: When using TETT, a Non-ATS frame enqueued into a non-empty shared TSN queue should inherit the eligibility time of the current tail element of that shared TSN queue. Similarly, for simulations using the GETT and SETT strategies, the assigned eligibility times must match the corresponding group eligibility time or super-group eligibility time at the moment of enqueuing. These information can simply be inferred from earlier entries in the result file. The tests offer decisive verification of the system's functional behaviour and confirm the correct implementation of the NETT strategies, without depending on internal implementation specifics.

Importantly, there is no need to explicitly control the input during these tests. The result files contain metadata about the simulation setup, including the specific NETT strategy used. This allows the system test suite to infer the expected behaviour automatically, enabling it to validate outputs without requiring manual tracking or verification of input configurations.

5.3.4 Regression Tests

Another testing methodology used throughout the development of the simulations is regression testing. In contrast to unit or system tests, regression tests are not designed to verify the correctness of simulation results or system behaviour. Instead, their purpose is to detect unintended changes in behaviour introduced during development, so-called regressions. These can occur as a result of performance optimisations, code refactoring, or other modifications that unintentionally affect previously stable parts of the system.

In the context of OMNeT++, regression tests are particularly important as every small change can affect scheduling behaviour. Without regression tests, these differences would likely go unnoticed.

Besides checking for regressions, they serve another purpose: Regression tests provided a quick-check functionality during bug fixing. Since changing system behaviour is often the goal of bug fixes, regression tests can determine whether a specific change actually impacted the simulation results, without needing to wait for full system tests to complete.

One common approach to implementing regression tests is to log the simulation output and compare logs across multiple runs. However, this method is sensitive to irrelevant

changes such as modified debugging outputs. To address this, a more robust approach is employed by leveraging OMNeT++'s built-in fingerprint testing.

Fingerprint tests in OMNeT++ work by hashing all simulation events relevant to the simulation's result. Each unique execution path produces a distinct fingerprint hash. Even subtle changes in event order such as a different ordering of two frames will lead to a different hash and thus a failed test. Since log outputs are not included in the hash computation, non-functional changes are automatically ignored [23].

It should be noted, however, that fingerprint tests can produce inconsistent results across platforms due to minor differences in system architecture, mostly regarding floating-point arithmetic. This can lead to differing fingerprints even when running the same simulation on different processor or operating system [23].

5.4 Benchmarking

In the context of this study, benchmarking refers to the systematic evaluation of the selected NETT strategies to assess their performance, impact on affected streams and suitability for real-time communication requirements in TSN networks. Before introducing new mechanisms, it is crucial to determine their compliance with the core objectives of TSN: deterministic behaviour, bounded delay, bounded queue lengths, minimal frame loss and low jitter. This process involves designing specific scenarios and selecting appropriate metrics to evaluate how well the mechanisms perform under various loads and network configurations.

Macro- vs. Microbenchmarking: Benchmarking approaches can generally be divided into two categories: macrobenchmarking and microbenchmarking.

Macrobenchmarking refers to performance evaluation at a system-wide level, typically using aggregated metrics collected over extended periods or across the entire network. This approach is useful for obtaining a high-level understanding of a mechanism under realistic, often variable, workloads. However, while being informative, macrobenchmarking tends to mask localised effects, making it difficult to isolate the impact of specific mechanisms or configurations. Additionally, since macrobenchmarking is not as controlled regarding the exact order of frame transmissions, it is less effective at testing worst-case

edge cases, which may be overshadowed by the volume of collected data or might not be tested at all.

In contrast, microbenchmarking focuses on detailed, targeted evaluations of specific components or scenarios within the system. It involves controlled experiments that isolate particular system aspects, such as queuing behaviour at individual nodes or delay variations under certain traffic patterns. Microbenchmarking allows for a more in-depth understanding of causality and system dynamics, explaining why certain behaviours occur, rather than simply observing their occurrence. Additionally, it allows for the intentional construction of edge cases to assess their potential impact and failure modes for risk analysis.

However, microbenchmarking does have its disadvantages. Manually constructing specific scenarios can be time-consuming compared to running simulations based on pre-recorded real-world network data. Moreover, since microbenchmarking focuses on isolated micro-level effects, deriving insights about larger-scale performance might be challenging or even impossible. That said, as long as the data gathered through microbenchmarking provides definitive answers to the research questions, these drawbacks are generally acceptable, depending on the context and the gained information.

In the context of this research, microbenchmarking was identified as the more suitable approach. By focusing on how particular NETT strategies influence shared TSN queue lengths at individual TSN nodes or how they alter timing behaviour of specific streams, microbenchmarking provides a clearer picture of the strategies' effectiveness and behaviours. This enables more precise interpretations of the results and aids in drawing sound conclusions about their applicability and the risks for real-world scenarios.

5.5 Scenario Topology Design

When constructing benchmarking scenarios for evaluating the performance of the NETT strategies, there are several important parameters that must be specified. This thesis differentiates between topology and stream configuration parameters. The network topologies used for testing are called scenarios. Each scenario can then be used for testing under different traffic conditions, defined by specific configurations. The configurations will be discussed later.

This section will focus on the parameters of scenarios, which are:

- the types and number of streams in the network,
- the number of TSN nodes in the network,
- the number of egress ports and
- the number of ingress ports.

The following sections will briefly discuss each of the parameters listed above, providing an explanation as to why specific combinations were selected for the scenarios and how they contribute to a comprehensive evaluation of the solution strategies.

5.5.1 Traffic Types and Number of Streams

The scenarios in this research are designed around three distinct streams. Two of them arise directly from the nature of the Non-ATS Conflict. Specifically, a TSN network must feature both ATS and Non-ATS traffic sharing the same TSN queue for the conflict to emerge, as discussed in Section 4.1. As such, every scenario includes these two essential traffic types:

- The **dependent stream** — an ATS stream that shares its queue with Non-ATS traffic, making it directly susceptible to the Non-ATS Conflict.
- The **evaluation stream** — a stream carrying Non-ATS traffic that causes the Non-ATS Conflict.

Additionally, an optional third type of traffic can be included:

- The **independent stream** — another ATS-scheduled stream, which may or may not be queued in the same TSN queue. It is used to examine the indirect impact of the NETT strategies on unrelated traffic, or vice versa. It can also serve as a reference point for cross-verification of the behaviours observed in the dependent stream.

To improve visual clarity in illustrations, each stream is assigned a specific colour. Figure 5.1 provides an overview of these visual identifiers.

Figure 5.1: Throughout this thesis, three colours will be used to highlight the three different streams. Blue for the dependent stream, pink for the evaluation stream and mint for the independent stream.

5.5.2 Number of TSN Nodes

The minimal network configuration considered in this thesis consists of three TSN nodes: A source, an intermediate TSN switch and a sink. While it is technically possible to define topologies with fewer nodes, such configurations are excluded from consideration. No node is configured to act as both source and sink simultaneously and all links are treated as if they were unidirectional. Given the assumption that modern Ethernet-based networks are full-duplex by default, this simplification remains realistic while facilitating scenario design and result interpretation.

One way to vary the number of TSN nodes is by distributing the streams across multiple sources. When multiple streams originate from the same source, their traffic is inherently serialised in the source's egress queue, as only one frame can be transmitted to the switch at any given moment. This serialisation prevents simultaneous frame arrival at the switch, which limits the observability of certain behaviours of the NETT strategies. To address this, scenarios include between one and three sources, allowing simultaneous traffic injections of up to all three streams.

In scenarios with only a single source, the independent stream is omitted. For the independent stream to yield additional insight, it must be at least partially isolated from either of the other streams. Scenarios where all traffic originates from the same source were therefore deemed to be not beneficial enough and were excluded in favour of more informative configurations.

Intuitively, the number of sinks can be adjusted as well. In a two-hop topology, this directly affects how many egress ports are used on the intermediate switch. These implications are discussed in more detail in the next section.

Lastly, it is possible to increase the number of nodes without altering the basic topology by adding additional switches between the source and sink. This linear increase in the

number of hops can reveal behavioural trends in NETT strategies after multiple stages of forwarding. However, the effects of additional hops can be approximated by considering scenarios in which traffic has already been sequentialised at the source rather than by an upstream switch. Since sequentialisation behaviour, not the origin of it, is the dominant factor for the strategies under investigation, we can assume comparable results whether sequentialisation occurs at the source or an upstream switch.

The major advantage of relying on this approximation is simplification: by limiting the complexity of the network, microbenchmarking becomes more practical. It becomes easier to craft specific frame arrival orders at the switch, which is crucial for exploring edge cases.

5.5.3 Number of Egress Ports

Constructing topologies in which multiple egress ports at the intermediate switch are utilised requires the use of multiple outgoing links from the switch to downstream nodes. These links could technically connect to the same downstream node, resulting in a parallel-link configuration. However, doing so provides no tangible benefit in the context of this research. Assigning each link to a distinct downstream node results in a more transparent configuration that is easier to observe and analyse. Therefore, no two nodes are connected via multiple links, as shown in Figure 5.2.

Introducing multiple sinks allows for the streams to be distributed across different egress ports. Due to the nature of the Non-ATS Conflict, the dependent stream and evaluation stream must share the same egress port, as this shared queue is central to the conflict. This is illustrated in Figure 5.2. However, as can be seen in the same picture, the independent stream can be configured to use a separate egress port, rendering it isolated from the other two streams. This decoupling allows for the analysis of influence of Non-ATS Eligibility Time Tagging strategies across streams of multiple different egress ports.

5.5.4 Number of Ingress Ports

Since each source connects to the switch through a dedicated link, distributing the streams across multiple sources utilisees multiple ingress ports at the switch.

Varying the number of utilised ingress ports has multiple effects. As discussed earlier, when multiple streams originate from the same source, their traffic is sequentialised at the source. By separating the sources, each separated stream can transmit in parallel allowing for configurations where multiple frames arrive at the switch at the same time.

Secondly, due to the different ingress ports, the streams are part of different ATS scheduler groups, making such topologies valuable for conducting tests targeting effects associated with them.

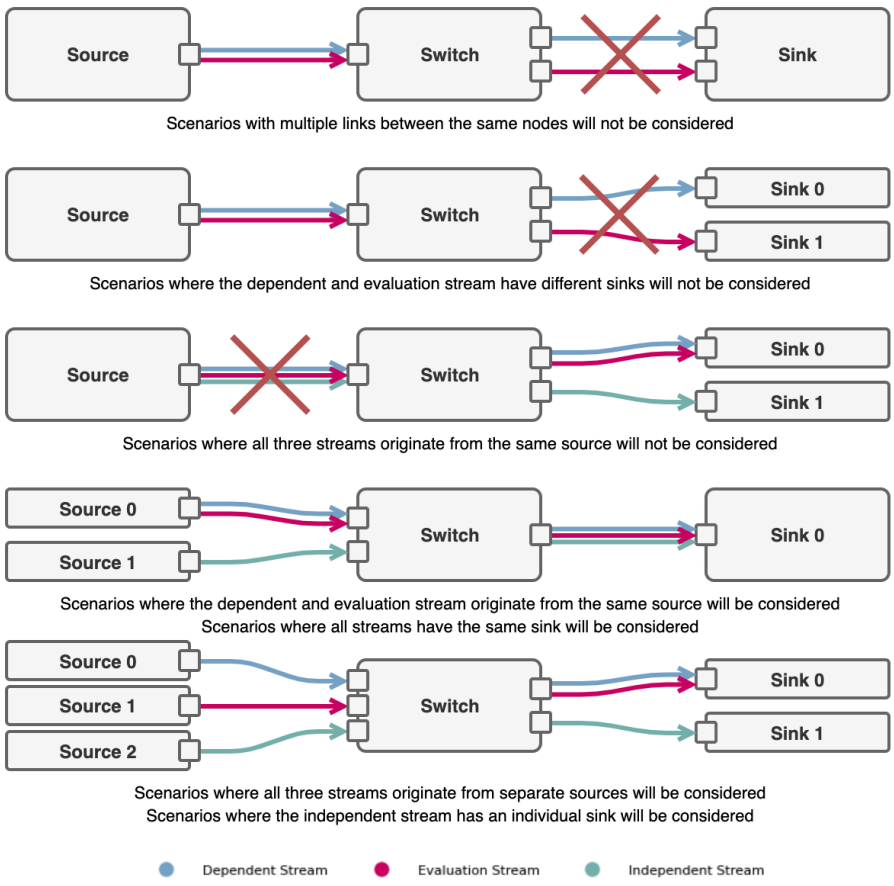


Figure 5.2: Configurations that illustrate what kinds of scenarios will be considered in this thesis. Note that the arrows show logical streams and not physical links. Physical links are not explicitly shown as they are discernible by the ports of the nodes.

5.5.5 Selected Scenarios

Based on the parameters discussed above, a total of eight scenarios were constructed. Each scenario represents a distinct condition under which the Non-ATS Conflict can occur. All selected scenarios are presented in Figure 5.3.

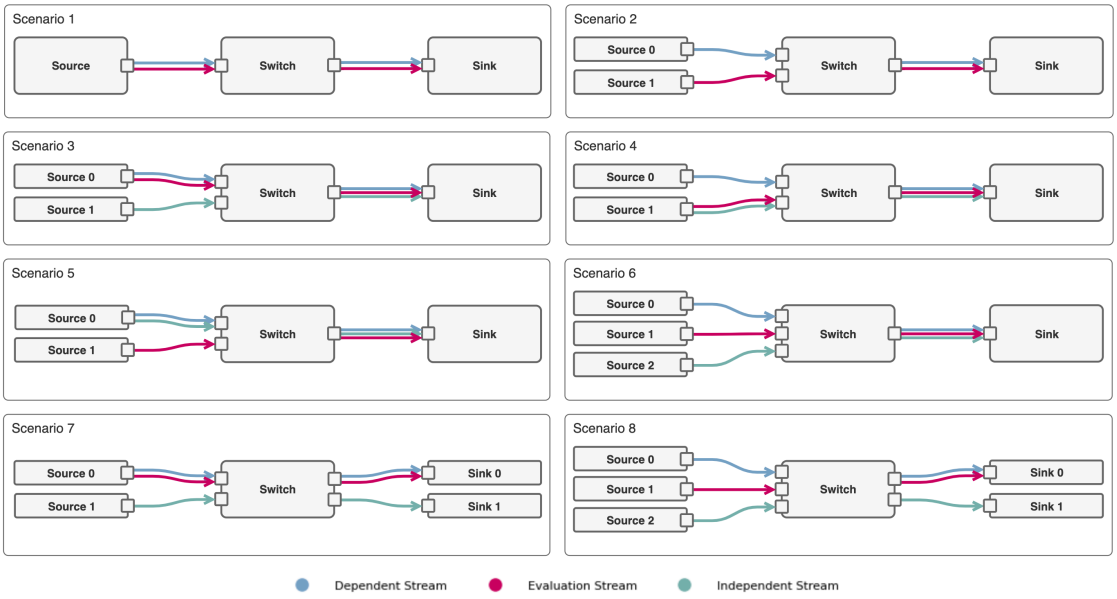


Figure 5.3: The eight scenarios selected for evaluation of the NETT strategies.

Scenarios 1 and 2 do not feature an independent stream. These scenarios are the only possible ones without it, given the applied limitations detailed in the previous sections.

The remaining six scenarios do incorporate the independent stream with varying numbers of sources and sinks.

The evaluation chapter will revisit specific scenarios in the context of the presented results.

5.6 Reference and NETT Setups

To evaluate the effectiveness of the examined NETT strategies, each of the eight selected scenarios is tested under two distinct setup types: The reference setup and the NETT setups. While the underlying scenario remains constant across all setups, the treatment of the evaluation stream differs.

The reference setup serves as a baseline against which all other results are compared. In this setup, the evaluation stream is shaped using a dedicated ATS scheduler, making it a fully IEEE 802.1Q compliant ATS stream.

In contrast, NETT setups leave the evaluation frames unscheduled. In these setups, the evaluation stream is handled using one of the three selected NETT strategies instead.

By ensuring that the reference setups are as similar to the NETT setups as possible, any observed differences in performance can be directly attributed to the presence or absence of a NETT strategy or ATS scheduling mechanism.

5.7 Performance Metrics

Before introducing the configuration parameters and selected values for testing, it is beneficial to first define the performance metrics used for evaluation. Selecting appropriate metrics at the outset allows the analysis to focus on the most relevant aspects of system behaviour. Crucially, different metrics are influenced by different subsets of configuration parameters. By choosing the metrics first, parameters that do not meaningfully affect them can be excluded from consideration, thereby reducing experimental complexity and preventing redundant or uninformative configurations from being tested.

In the context of network performance evaluation, several key metrics are commonly considered, including but not limited to:

- bandwidth utilisation and throughput
- queue lengths
- frame drop rates
- lost frames
- latency and delay
- jitter

Given that the primary objective of ATS is to ensure bounded latency, minimal jitter and bounded link utilisation, these metrics appear to be well-suited for the focus of this thesis. However, upon closer examination, especially the measurement of latency may not be as conclusive as initially expected. The following sections will go over each metric and discuss their relevance for the thesis.

5.7.1 Bandwidth Utilisation and Throughput

In networks with unknown or dynamic parameters, measuring bandwidth utilisation and throughput is essential to assess overall efficiency and detect potential bottlenecks. However, in the context of microbenchmarking, these metrics might become obsolete. The environment is fully controlled, with all traffic rates and link capacities predefined. As such, bandwidth usage and throughput can be inferred from the configuration and the focus shifts to more relevant metrics that provide insights into the local effects and timing behaviour of NETT strategies under test.

5.7.2 Queue Lengths

Queue lengths are important for understanding the internal behaviour of a network. They provide insights into congestion levels and the interactions between different priorities.

In the context of this thesis, TSN queue lengths are particularly valuable because they reveal how the selected NETT strategies influence resource contention within the queues. For example, if a NETT strategy consistently causes a build-up in the shared TSN queue, this may indicate ineffective regulation. Conversely, stable or low shared TSN queue lengths suggest that a strategy is successfully maintaining scheduling discipline and minimising conflict between streams.

Importantly, TSN queue lengths can also act as indicators for whether configurations behave as intended. If a configuration is designed to underutilise the available bandwidth, but the measurements show consistently high queue lengths, this discrepancy may point to a flaw in the configuration.

5.7.3 Frame Drop Rates

In the context of TSN there are two kinds of frame drops that can commonly occur. Firstly, frames can be dropped by an ATS scheduler due to MRT exceedance, as explained in Section 2.2.3.2. Since most of the configurations in this thesis are explicitly designed to not exceed the MRT, any occurrence of MRT-related drops is a clear indicator that the implementation is not behaving as expected. In this sense, MRT drop rates are used primarily as a validation mechanism.

Secondly, frames can be dropped due to buffer overflows. These kinds of drops will not be considered in the evaluation. This is because all TSN queues are modelled with infinite buffer capacity, thereby eliminating queue-based frame loss entirely. Any insight that would have been gained from observing buffer-induced drops can instead be inferred from monitoring queue lengths.

5.7.4 Lost Frames

Frames that are not delivered to their destination by the end of a simulation may indicate starvation in the TSN queues, especially if certain streams are persistently delayed or blocked by higher-priority traffic. In the context of this study, undelivered frames are used for validation to check if they are within expected ranges and whether a configuration caused backlog of frames in queues.

As discussed later in Section 5.8.1, simulation runs are deliberately stopped after a predefined time. Therefore, some frames may still be in transit at the end of the simulation, meaning that unless the configuration was designed to guarantee full delivery within the simulation window, the presence of undelivered frames does not necessarily indicate misbehaviour.

5.7.5 Latency and Delay

Latency or end-to-end delay measures the total time a frame takes to travel from source to destination. While commonly used in network evaluations, it is not suitable for analysing mechanisms with highly localised effects, as in this study. This is because latency includes:

- the time spent in an egress queue at the source, which is unrelated to the tested NETT strategies, as well as
- transmission times, which vary with frame size and therefore introduce delay variations which are irrelevant for queuing analysis and make gathering insightful data on delay impacts of employed NETT strategies harder.

To address this, a more targeted metric called queuing delay is used. It includes only the time a frame spends in the TSN queue of the intermediate switch. The queuing delay can be broken down into two components:

- **Scheduler delay**: Time until the frame becomes eligible for transmission
- **Cross-traffic delay**: Time between eligibility and actual transmission, due to other queued frames being transmitted first

These delays were illustrated previously in Figure 2.6. Note that if a frame is delayed due to the transmission of another frame from the same stream, this time is still included in the cross-traffic delay. In other words, streams can impose cross-traffic delay on themselves.

Queuing delay is the primary metric used throughout this research. It directly reflects the behaviour of the tested NETT strategies within the switch, offering precise insight into its impact on delay of the streams. Rather than focusing on absolute queuing delay values, the relative changes observed when comparing NETT setups against reference setup results are of greater significance, as they reveal the strategies' influence on timing behaviour.

5.7.6 Jitter

Jitter is measured based on queuing delay, as the same arguments apply as for latencies: If frames are not sent at precise intervals, measuring total jitter would obscure the impact of the variance induced by the strategy under test.

Again, the focus is on relative changes in queuing jitter between different setups to evaluate the changes introduced by the employed NETT strategy.

5.8 Configuration Parameters

With the scenarios and evaluation metrics now established, the last step is to define explicit configuration parameters. This includes assigning concrete values and behaviours to each stream for the following variables:

- frame size
- bandwidth
- relative timing
- burst characteristics
- priority

Understanding the impact of the parameters and how they are related is crucial to developing relevant benchmarking configurations. Each parameter will be discussed in the following sections.

5.8.1 Frame Sizes and Bandwidth Utilisation

The bandwidth usage of each stream is determined by the size of the emitted frames and the frequency at which these frames are sent. For the purposes of this discussion, it is assumed that each stream sends frames at regular intervals, referred to as send intervals, beginning from a specified start time.

Frame Sizes: As discussed in Section 2.2.1, the payload size of TSN frames can vary. Furthermore, links of various capacities can be used, altering the physical limitations of the network. In theory, any combination of these parameters could be of analytical interest. However, exhaustively testing all possibilities is neither feasible nor necessary.

As outlined in Section 5.7.5, the chosen metric of queuing delay is designed to yield comparable results regardless of the exact frame sizes and link capacities used. Since both link capacities and frame sizes influence the transmission times of frames, setting one of them to a fixed value simplifies the analysis. Therefore, all link capacities are set to a constant 100 Mbit/s in all tested configurations. Furthermore, physical signal propagation delays are assumed to be zero, since they do not affect queuing delays either.

To facilitate analysis, frame sizes are selected so that their transmission times translate to round figures. This simplifies inferring how many frames contributed to a measured cross-traffic delay and aids in timing calculations for configurations involving specific frame arrival orders at the switch.

For these reasons, only three different frame sizes are used in this thesis: 1250 B, 625 B and 125 B. In a network with a bandwidth of 100 Mbit/s, these correspond to transmission times of 100 ms, 50 ms and 10 ms respectively. Each of these frame sizes *includes* the IFG, so the actual frame sizes are 12 B lower and the actual transmission times are 0.96 ms less than stated above. Treating the IFG as part of the frames simplifies further calculations such as the bandwidth usage of a stream. After all, IFGs constitute another form of overhead associated with each frame. Accordingly, the CIR parameters of the ATS schedulers are configured to account for them as well.

Bandwidth Utilisation: With the frame sizes and link capacities established, the target bandwidth usage for each stream can be chosen, which in turn dictates the send interval. Most configurations use multiples of 25 Mbit/s per stream, again simplifying timing calculations. If a source sends frames back-to-back, its bandwidth utilisation is 100 % of the link's capacity. As the gap between successive frames increases, the utilisation decreases. The relationship between the send interval and the target bandwidth is given by:

$$T_{\text{send}} = \frac{t_{\text{tx}} \cdot R_{\text{link}}}{R_{\text{target}}} \tag{5.1}$$

Where:

- T_{send} is the send interval, i.e., the time between the start of two successive frame transmissions,
- t_{tx} is the transmission time of a single frame, including IFG,
- R_{link} is the link capacity,
- R_{target} is the desired bandwidth utilisation of the stream.

Sim-Time Limit: In OMNeT++, simulations can be configured to terminate after a predefined duration. This parameter is called sim-time limit [23]. For the simulations conducted in this thesis, this limit was set to 1 s.

The minimum number of frames transmitted within 1 s for any given stream can be derived from the configuration featuring the largest frame size of 1250 B and the smallest bandwidth limit of 25 Mbit/s. Under these conditions, using Equation 5.1, the total number of transmitted frames is calculated as follows:

$$t_{\text{tx}} = \frac{1250\,\text{B}}{100\,\text{Mbit/s}} = 100\,\mu\text{s}$$

$$T_{\text{send}} = \frac{100\,\mu\text{s} \cdot 100\,\text{Mbit/s}}{25\,\text{Mbit/s}} = 400\,\mu\text{s}$$

$$\frac{1\,\text{s}}{400\,\mu\text{s}} = \underline{\underline{2500\ \text{frames}}}$$

A minimum of 2500 frames was considered sufficient to obtain statistically meaningful results for each stream.

5.8.2 Relative Timing between Streams

If multiple streams are configured with the same or harmonically related send intervals, their relative timing repeats periodically throughout the simulation run. This leads to a deterministic ordering in which the same stream is always processed first, followed by the others in a fixed sequence. Such behaviour does not reflect real-world conditions where clock jitters and drifts, OS scheduling and other dynamics introduce timing variations. Deterministic alignment can bias the simulation results, as certain streams may consistently experience more favourable queuing conditions than others.

To reduce such effects, a minimal amount of send interval jitter can be introduced to each stream if needed. Specifically, a uniformly distributed jitter of ± 1 ps is applied to the send interval of each stream. This variation itself is negligible, as it is more than five orders of magnitude below any duration of interest for the gathered results. By applying such minimal jitter, the frames from different streams are interleaved in various sequences over time, allowing the simulation to capture a more statistically representative behaviour of

the scheduling and queuing mechanisms, without favouring any specific stream due to persistent alignment.

However, care must be taken when introducing jitter. Even small amounts of jitter can accumulate over time, causing a phenomenon called drift. Drift refers to a gradual deviation of a stream's nominal schedule defined by its start time and send interval. If the jitter is biased over a period of time, the deviations add up, which can reintroduce systematic bias in the results. An example of drift caused by lagging jitter is shown in Figure 5.4. In the graph, the send interval of the UdpBasicBurst experiences lagging jitter three times in a row. Since each send interval is calculated starting from the actual send time, the jitters accumulate.

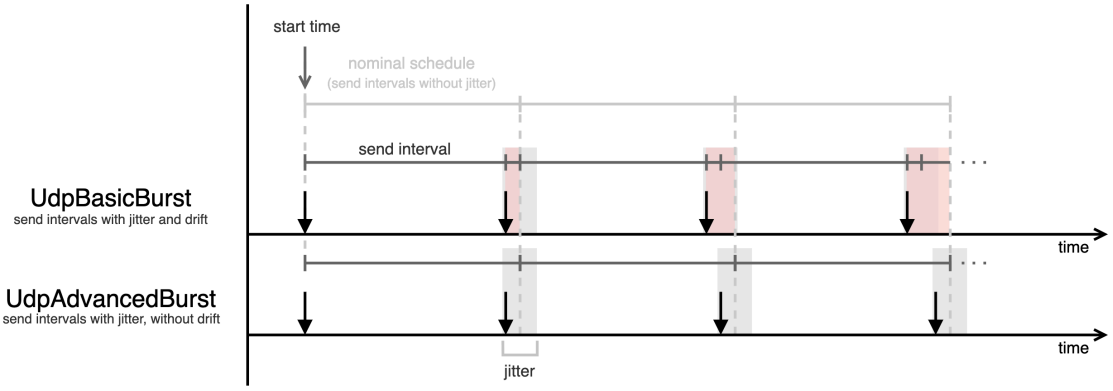


Figure 5.4: While UdpBasicBurst introduces drift when jitter is applied to the send interval, UdpAdvancedBurst maintains a stable jitter around the nominal schedule without causing drift.

To avoid such problems for the sake of controllable scenarios, a custom module named `inet.applications.udpapp.UdpAdvancedBurst` was implemented, based on OMNeT++'s existing `inet.applications.udpapp.UdpBasicBurst`. This new module supports a parametric separation between jitter and drift. When a jitter value is configured, `UdpAdvancedBurst` ensures that each actual send time remains centred around the intended nominal schedule without accumulating deviation. This behaviour is depicted in Figure 5.4. For instance, with a send interval of $T_{\text{send}} = 400\,\mu\text{s}$ and jitter of $\pm\,1\,\text{ps}$, the frame is sent at precisely $n \cdot 400\,\mu\text{s} \pm 1\,\mu\text{s}$ for integer $n > 0$. This preserves alignment to the timing structure while still achieving randomness in arrival ordering.

Nevertheless, it remains possible to configure drift for `UdpAdvancedBurst`.

5.8.3 Burst Characteristics of the Streams

In the previous sections, a fixed send interval was assumed. However, bursting streams are of greater practical interest, as they are a primary motivation for using traffic shaping mechanisms. Bursts place momentary stress on the system, making them valuable for testing scheduler behaviour under load.

The `UdpBasicBurst` module introduced in the previous section includes several parameters to control burst behaviour, as illustrated in Figure 5.5. Starting at the configured *start time*, traffic generation alternates between two phases: A burst interval of length *burst duration* during which frames are sent at a fixed *send intervals* and a sleep interval of length *sleep duration* during which no frames are sent. Jitter is applied to all send events, including the first frame of each burst, with the exception of the very first frame in the simulation.

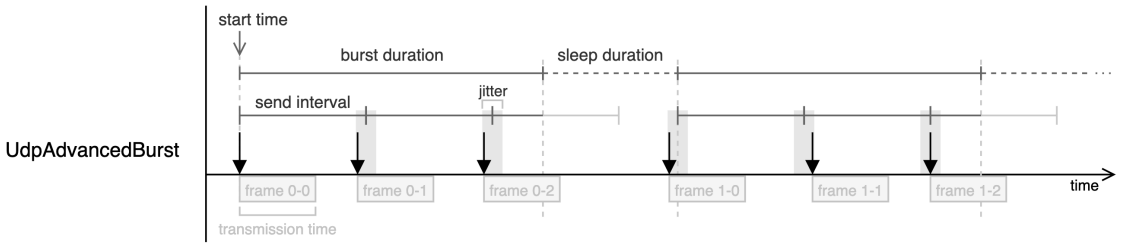


Figure 5.5: The UdpAdvancedBurst module has four main parameters. During the burst duration, frames are emitted each send interval. A non-drifting jitter can be applied to the send interval. Each burst duration is followed by a sleep duration. Drift can be applied using another parameter not illustrated here.

As shown in Figure 5.6, the number of frames sent per burst can vary due to jitter. Specifically, if the jitter applied to the last expected send time pushes it beyond the burst interval and into the sleep phase, the corresponding frame may be omitted in some bursts but not others. To prevent this inconsistency while preserving calculation simplicity, the UdpAdvancedBurst module introduces an optional maxPacketsPerBurst parameter. This limits the number of frames transmitted per burst to a fixed maximum, ensuring consistent burst sizes and making configurations easier to reason about. This parameter is optional, but improves usability in some cases.

When tuning the burstiness of ATS streams, the CBS parameter of the ATS scheduler can be adjusted to accommodate closely spaced frames, allowing them to pass through the scheduler without incurring excessive delay. Additionally, the MRT must be set

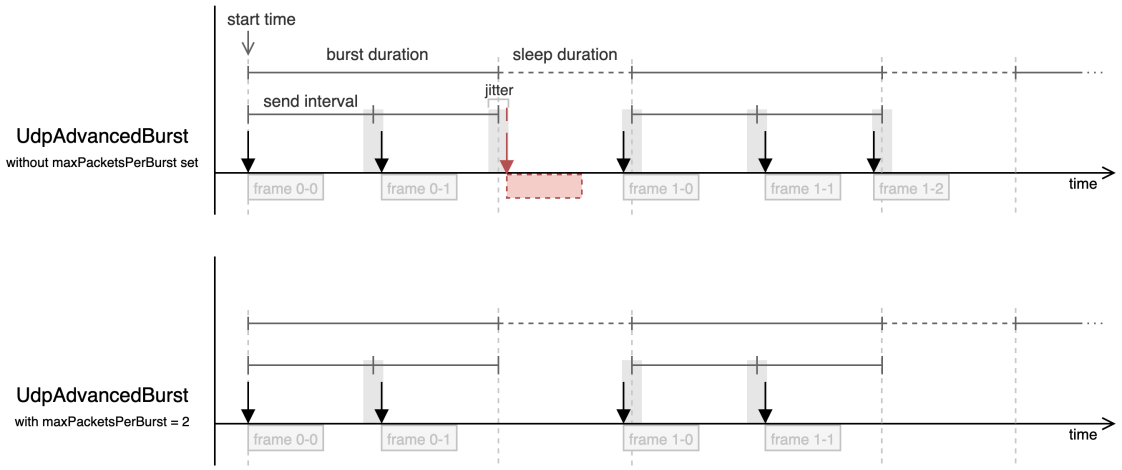


Figure 5.6: The jitter applied to the last send interval of the first burst pushes it beyond the burst interval and into the sleep phase, so that the corresponding frame is omitted in the first bursts but not the following. To ensure each burst has a constant size, the max. packets per burst parameter can be set.

sufficiently high to ensure that the final frames of the bursts are not dropped due to their eligibility times exceeding the configured MRT.

Notably, adjusting the CBS is not strictly necessary. If the sleep duration between bursts is long enough and the MRT is configured with an adequate margin, any short-term exceedance of the enforced traffic rate can still be balanced out over a longer period. This flexibility makes it possible to intentionally configure specific burst patterns while remaining compliant with the long-term bandwidth constraints enforced by the configured ATS scheduler.

5.8.4 Priority

Among the eight selected scenarios introduced in Section 5.5.5, six include an independent stream that competes for shared network resources with both the evaluation and the dependent stream. For each of these six scenarios, two configuration variants are tested:

- **Equal-Priority Variant (Default):** The independent stream is assigned the same priority as the other streams. Unless explicitly stated otherwise, this variant is used and referred to by default.
- **Multi-Priority Variant:** The independent stream is assigned a lower priority than the other streams. If applied, this variant is explicitly referred to as the multi-priority variant of a configuration.

This variation allows for a targeted analysis of the NETT strategies' behaviour in the presence of lower-priority traffic. Importantly, no other configuration parameters such as bandwidth allocations or timing characteristics are modified between variants. As a result, any differences in performance can be directly attributed to the priority assignment.

Since ATS scheduler groups are defined per priority level, a change in priority naturally results in a different group handling the independent stream, even when other traffic enters through the same ingress port. This makes the multi-priority configuration variants particularly useful for analysing the extent to which lower-priority ATS traffic is impacted by the different Non-ATS Eligibility Time Tagging strategies.

5.8.5 Selected Configurations

Although the theoretical implications of all configuration parameters have been outlined above, not every tested configuration will be examined in detail in this chapter. The full set of parameters and their assigned values is documented in Table A.2, providing a complete reference for all evaluated setups.

A detailed walkthrough of each individual configuration would add limited value at this stage and needlessly increase the complexity of this chapter. Instead, specific parameters will be revisited in the evaluation chapter when their influence is directly relevant for understanding the presented results. This approach maintains clarity and avoids redundancy, while still ensuring that all essential factors are addressed in context.

5.9 Analysis Software Suite

Executing simulations across all combinations of scenarios, setups and configuration variants results in over 2500 result files produced by OMNeT++, each containing thousands of individual data points. This substantial volume of data necessitated efficient and automated processing to enable meaningful analysis. To address this, a python-based software suite was developed to automate the execution of simulations across various configurations and to generate graphs that facilitate efficient result assessment [24].

Just like the system test suite introduced in Section 5.3.3, the software is designed to automatically execute all scenarios defined within an OMNeT++ network configuration file. While OMNeT++ supports parameter studies [23], its configuration language was deemed unsuitable for the complexity of the configurations required for this research. Therefore, the configuration process was managed by the python software, automatically replacing parameters in the OMNeT++ configuration file prior to each simulation run. This method allowed the parameters to be managed centrally in a single file, reducing the risk of errors and providing precise, automated control over the configuration parameters discussed in Section 5.8. Furthermore, this process allowed for simple specification of new simulation configurations.

Apart from substituting fixed parameters, the software suite also supports two mechanisms to structure simulation campaigns: repetitions and iterations. While similar in concept to features natively supported by OMNeT++, the implementation in this suite offers significantly greater flexibility with less configuration complexity. Repetitions allow the same configuration to be executed multiple times with different random seeds. This is particularly useful for scenarios involving randomised elements, such as jitter, where repeated runs help to average out stochastic effects and yield statistically robust results. Iterations, in contrast, enable the systematic variation of one or more parameters across a predefined set of values or according to a mathematical function, enabling parameter studies for efficient investigation of the impact of individual variables on system behaviour.

Like previously, upon completion of the simulation runs, the analysis software imports the results from the generated OMNeT++ output files. The data is parsed, while preserving all relevant scenario and configuration details, enabling traceability of each result to its specific simulation setup. This preservation of metadata facilitates the automatic generation of data plots, ensuring that axes are correctly labeled and titles are generated.

The automation of data visualisation drastically accelerates the analysis process. The manual work involved is limited to identifying significant findings, such as deviations between expected and actual results and creating specialised graphs to highlight these findings where necessary.

A key feature of the software is its ability to transform raw simulation data into well-structured data types. OMNeT++ records data in the form of vectors. For each measurement, such as the queue length of a particular TSN queue or the delay of a specific frame, an individual vector is recorded. Each entry of a vector includes three data points: the event number and simulation time at which the data was recorded and the value of the measurement at that time. As each measurement is recorded individually, correlating data between different measurements involves correlating data across multiple result vectors. This is quite challenging, as there is no information available such as the id of the frame an entry was recorded for and related data points are recoded at different simulation times throughout the processing of a frame.

The python software overcomes this challenge by utilising topology information encoded in the names of the networks configured in OMNeT++'s configuration file. These names encode critical information, such as the number of sources, sinks, stream types and the priority of the streams. By extracting this information, the software can infer which ATS scheduler and TSN queue in the switch handled which frames. Since each result recorded by OMNeT++ is linked to the specific module that generated it, as explained in Section 5.1, this enables automatic identification of the stream associated with the recorded result.

Once the stream associated with a given result is identified, this information is used to correlate multiple data points from different OMNeT++ modules. This approach allows for the creation of timelines that track the lifecycle of each frame. These timelines provide detailed information, including the frame's arrival time, assigned eligibility time, queuing delay and the time at which it was transmitted to the sink. Timelines are saved as images visualising the timing sequences, as well as in a text file format for manual inspection. These timelines are invaluable for understanding the behaviour of the NETT strategies under various configurations.

6 Evaluation

This chapter presents the experimental evaluation of the selected NETT strategies across various controlled configurations and scenarios. Building upon the concepts and methodology introduced in the preceding chapters, the evaluation aims to characterise the behavioural properties of the NETT strategies as well as to identify potential strengths and limitations. To this end, the most insightful configuration-scenario pairs are analysed with respect to average queuing delays and jitters, with supplementary observations on queue lengths and frame drops where relevant. Key findings are derived by comparing results from the NETT setups against the corresponding reference setup results, as explained in Section 5.6.

6.1 Evaluation under Nominal Conditions

The first evaluated configuration is designed to minimise variability and provide a predictable, stable environment for assessing the intrinsic behaviour of the NETT strategies. By applying all statistical effect mitigation techniques described in Chapter 5, the system is set up under nominal load levels and tightly controlled traffic characteristics. This allows the strategies to be analysed in isolation, without interference from stochastic variations or system stress phenomena.

6.1.1 Configuration Overview

Each frame is fixed at a size of 1250 B and transmitted at consistent intervals of 400 µs, resulting in a steady bandwidth utilisation of 25 Mbit/s per stream (see Section 5.8.1). This rate is also set as the CIR for all configured ATS schedulers, effectively modelling source-shaped streams with predictable transmission behaviour, as explained in Section 2.2.3.

All streams initiate transmission simultaneously and each source maintains identical send intervals, producing synchronous groups of either two or three frames, depending on the inclusion of an independent stream (see Section 5.5.1). In addition to countermeasures against statistical anomalies caused by harmonically related send intervals, as detailed in Section 5.8.2, further precautions are taken: As explained in Section 5.8.3, the very first frame of each simulation is sent without jitter. Since the initial frame ordering can influence the results, six iterations are executed per scenario, systematically covering all possible permutations. The individual results of each iteration are then averaged to obtain the final results.

The full list of parameters for this configuration CFG-2 is provided in Table A.2.

6.1.2 Results

The following sections will present the results gathered for CFG-2. First, the findings for scenarios with separate sources will be presented, afterwards results for shared sources and for the multi-priority variant of CFG-2 (see Section 5.8.4) will be evaluated. The expected behaviour of each particular configuration-scenario pairing will be discussed, followed by the evaluation of the actually observed behaviour.

Expected Behaviour Separate Sources: For the analysis of behaviour with individual sources and consequently separate ATS scheduler groups per stream, Scenarios 2 and 8 were selected, which are depicted in Figure 6.1.

Scenario 2, features only the dependent and the evaluation stream. As explained in Section 5.5.1, this is the minimum number of streams. Given the absence of sequentialisation due to the streams originating from separate sources (see Section 5.5.2), virtually identical queuing delays are expected across all setups. Furthermore, since the streams are transmitted as if source-shaped to their respective bandwidth allocations, no scheduler delays are anticipated.

Scenario 8, extends Scenario 2 by introducing an independent stream that terminates at a separate sink. Due to the separate source and sink, the independent stream remains isolated from the other streams. Consequently, no interference is expected, its traffic should not experience any queuing delay and the delay behaviour of the dependent stream and evaluation stream should remain unchanged compared to Scenario 2.

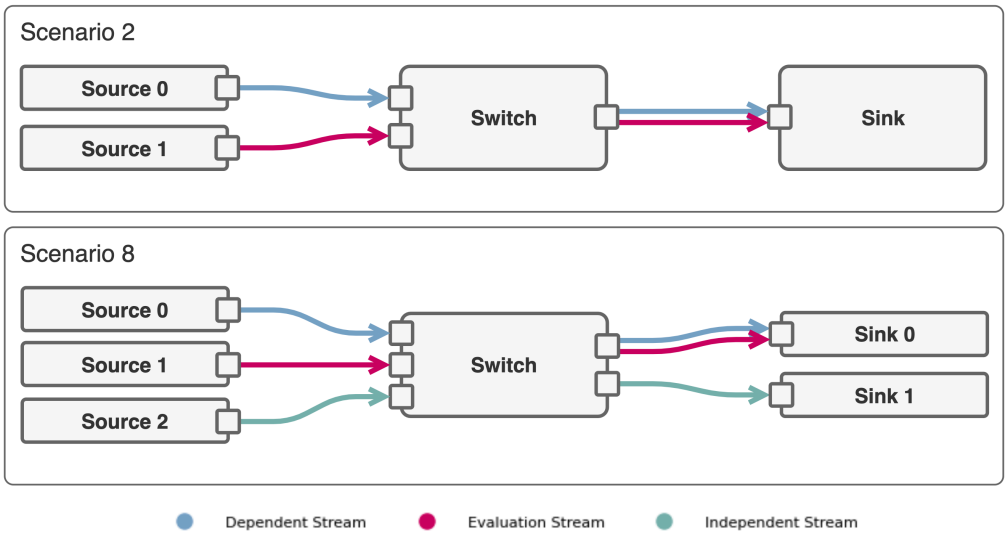


Figure 6.1: Topologies of Scenario 2 and Scenario 8. Both feature distinct sources for all streams. Scenario 8 extends Scenario 2 by adding an isolated independent stream terminating at a different sink.

Observed Results Separate Sources: As expected, TETT and SETT yield results identical to the reference setup, with average queuing delays of 50 µs, equating to half a frame's transmission time. This can be seen in Figure 6.2. The observed variation between 0 and 100 µs is consistent with the random ordering of simultaneously emitted frames, which is part of the statistical effect mitigation techniques described in Chapter 5. No scheduler delay occurs, confirming the shaping mechanisms remain inactive.

As anticipated, the independent stream is unaffected and exhibits no delay, due to its separation from the other streams. Since this also means that the results for the evaluation stream and the dependent stream are identical for Scenario 2 and 8, only the latter are shown in Figure 6.2 to avoid redundancy.

The only deviation is observed with GETT, which introduces a slight priority shift: Queuing delay for the dependent stream increases marginally by approximately 17 µs, while the queuing delay for the evaluation stream decreases correspondingly.

This behaviour stems from how GETT assigns eligibility times based on arrival time when the affected stream is the only one in its ATS scheduler group, as defined by Equation 4.2 in Chapter 4. Unlike TETT and SETT, which strictly order Non-ATS frames to the end of the shared TSN queue, GETT permits earlier enqueuing of the Non-ATS evaluation frames and therefore occasional overtaking of ATS dependent frames.

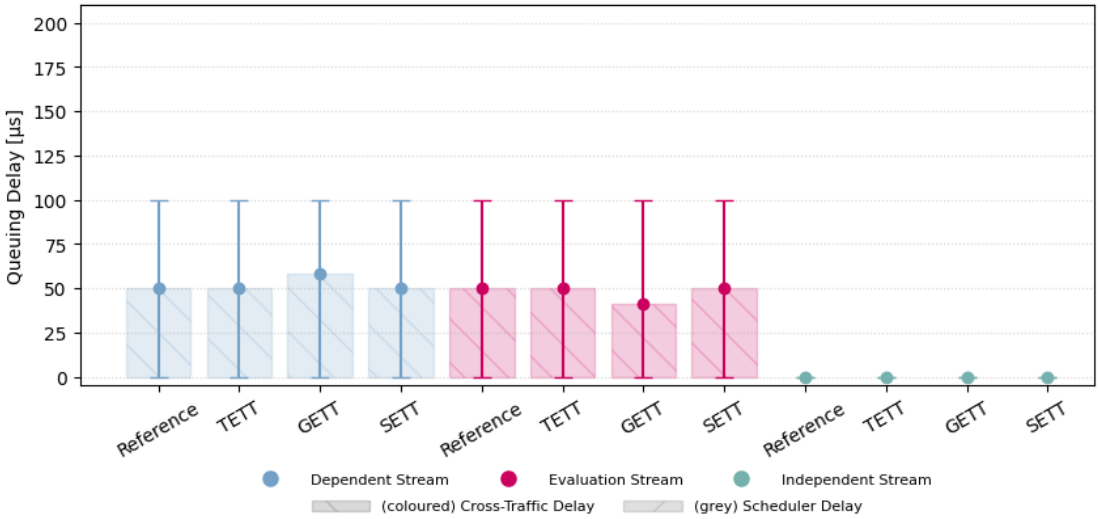


Figure 6.2: Average queuing delays and corresponding jitters for CFG-2, Scenario 8. Results are averaged over all initial send orders and depicted per stream and setup. The results for the evaluation stream and the dependent stream are identical for Scenario 2.

However, this effect does not accumulate across network hops, because once the streams are sequentialised after the first hop, they are grouped into a shared ATS scheduler group. As such, the delay caused by overtaking frames can no longer manifest in subsequent switches (exceptions apply in parallel-link topologies excluded from evaluation in Section 5.5.3).

Additionally, the relative magnitude of this effect diminishes with increasing path length and consequently increasing latency. The more hops a frame traverses, the higher its overall latency and the less the relative impact of this effect. Therefore, the behaviour does not fundamentally impair the applicability of GETT, particularly in typical deployments, where small jitters are tolerable within ATS-shaped traffic.

These results confirm that TETT and SETT avoid introducing any additional delay under idealised conditions, meaning conditions without temporary bandwidth exceedance, virtually no jitter, no drift and with fully repetitive transmission patterns. They neither defer Non-ATS frames, nor impact other ATS streams, regardless of queue sharing or physical separation.

In contrast, GETT introduces a minor, non-accumulative timing advantage for the Non-ATS stream. This behaviour is context-dependent and seems to not represent a critical concern.

Expected Behaviour Shared Sources: For the analysis of behaviour with shared sources and consequently shared ATS scheduler groups, Scenarios 1 and 7 were selected. Both are illustrated in Figure 6.3.

In Scenario 1, both the dependent stream and evaluation stream originate from the same source. Given the synchronised traffic configuration, each pair of frames will be sequentialised at the source before transmission. Since the previous results showed cross-traffic delay only, which should be eliminated by this sequentialisation, one might expect the queuing delays to approach zero.

Just like it was the case with Scenarios 2 and 8, Scenario 7 expands on Scenario 1 by introducing an additional independent stream, which again is entirely isolated from the others. Consequently, the independent stream should remain unaffected and results for the dependent stream and evaluation stream should be identical to those of Scenario 1.

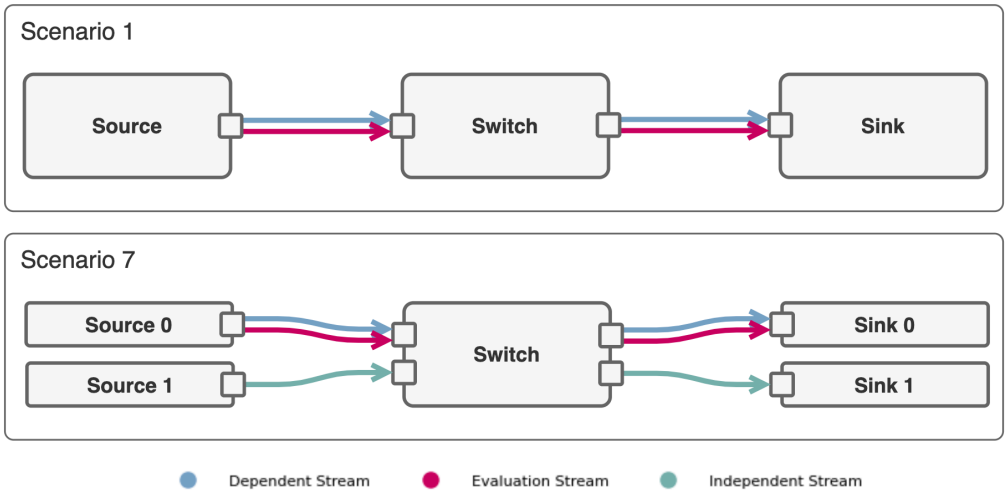


Figure 6.3: Topologies for Scenarios 1 and 7. Both feature a shared source for the dependent and the evaluation stream. Scenario 7 builds upon Scenario 1 by adding an independent stream, originating from a separate source and terminating at a separate sink. The results for the evaluation stream and the dependent stream are identical for Scenario 1.

Observed Results Shared Sources: As previously observed and expected, the results for the evaluation stream and the dependent stream are identical between Scenarios 2 and 7. Therefore, only the latter are shown in Figure 6.4, representatively for both.

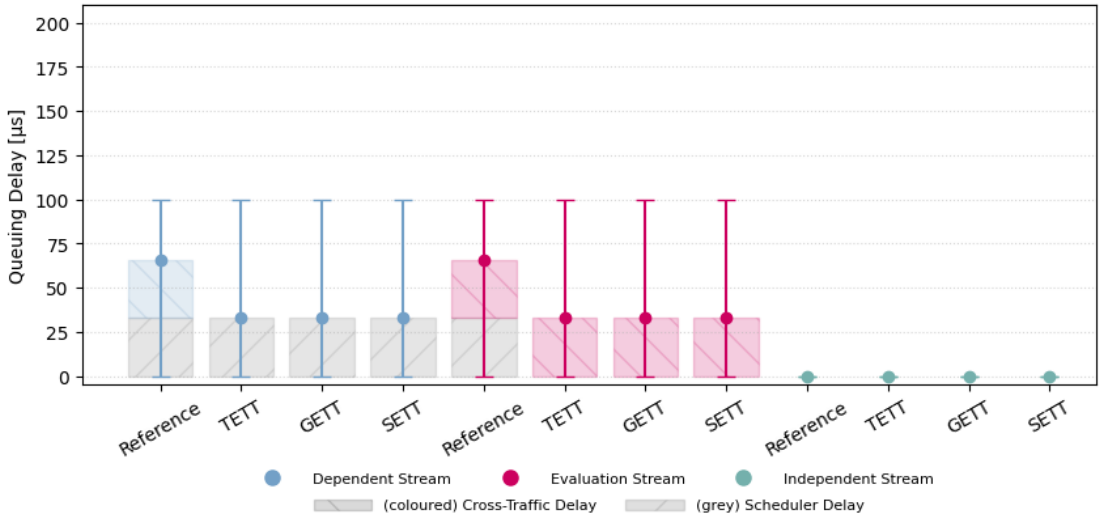


Figure 6.4: Average queuing delays and corresponding jitters for CFG-2, Scenario 7. Results are averaged over all initial send orders and shown per stream. Coloured areas of the bars indicate the portion of cross-traffic delay, grey bars indicate scheduler delay.

Contrary to expectations, both the dependent stream and the evaluation stream experience non-zero queuing delay on average. Moreover, as seen in Figure 6.4, all three NETT strategies outperform ATS in the reference setup on average by approximately 50 %.

In the NETT setups, the dependent stream experiences scheduler delay only, without additional cross-traffic delay. The inverse is true for the evaluation stream which solely faces cross-traffic delay. In contrast, in the reference setup, the dependent stream incurs additional cross-traffic delay on top of the scheduler delay present in both setups, caused by the ATS scheduled nature of the evaluation stream.

These observations can be explained by considering all possible initial transmission orders in each simulation run. In the reference setup, both the dependent stream and evaluation stream are ATS-shaped. The tightly configured ATS schedulers are sensitive to the first arrival sequence, which primes the credit counters and influences subsequent transmission behaviour.

Figure 6.5 illustrates this effect. In the first frame pair, the evaluation frame is buffered at the source while the dependent frame is being transmitted. When the evaluation frame arrives at the switch at 200 µs, in the reference setup, it is scheduled to be eligible for transmission immediately by the ATS scheduler, consuming credit.

By chance, in the next frame pair, the evaluation frame is transmitted first by the source and arrives at the switch at 500 µs. However, in the reference setup, the credits have not yet fully recovered since only 300 µs have passed and 400 µs are needed for full replenishment. As a result, the frame is delayed by 100 µs, leading to additional scheduler delay which in turn causes a cross-traffic delay for the following dependent frame.

As there is no ATS scheduler used for the evaluation stream in the NETT setup, this specific delay-chain does not occur. In the NETT setups, this chain reaction occurs only if it is the dependent frame that is delayed by its ATS scheduler. The evaluation frame, being unshaped, is always immediately eligible for transmission and therefore cannot introduce cross-traffic delay for the dependent frames.

This explains the results in Figure 6.4. In the reference setup, both dependent frames and evaluation frames experience queuing delay, caused on average by a combination of scheduler delay and cross-traffic delay. In the NETT setups, only the dependent frames are subject to scheduler delays and only they can influence evaluation frames by causing cross-traffic delay, therefore cutting the average queuing delay in half for both.

These results demonstrate that employing NETT strategies can reduce inter-stream interferences occurring due to sequentialisation at the previous node. These interferences are scheduling side effects that manifest when using ATS for two streams from a shared upstream node, meaning two streams arriving through the same link.

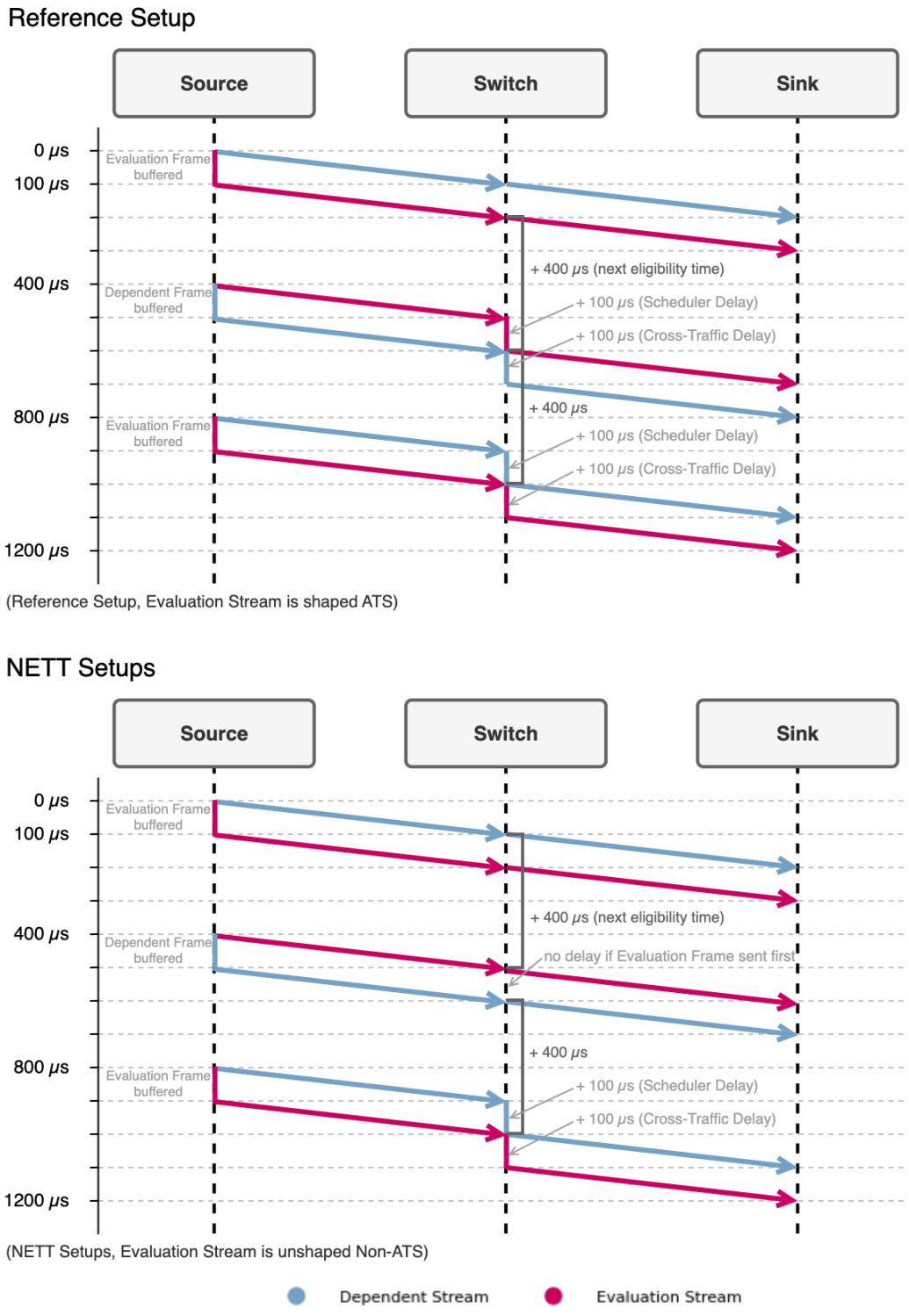


Figure 6.5: Transmission timings of the dependent and the evaluation stream in Scenario 1 CFG-2, compared for the reference setup and any of the three NETT setups.

Expected Behaviour Multiple Priorities: When employing shapers for rate limiting, they create gaps between successive frame transmissions to reduce their bandwidth consumption and smooth out bursts, as explained in Chapter 2. These gaps can be utilised by lower priority traffic for their transmissions. By using NETT strategies, less shaping gaps are created, as the Non-ATS frames are either queued with the same eligibility time as other frames or become instantly eligible for transmission. This should in theory lead to higher queuing delays for lower-priority streams.

To test this hypothesis and see if these effects are measurable for CFG-2, one of the multi-priority configuration variants can be examined, involving a lower-priority independent stream as introduced in Section 5.8.4. While this configuration variant could be tested for Scenario 7 to compare it directly to the results examined previously, this would yield no meaningful insight. In Scenario 7, the independent stream is forwarded using a separate egress port, isolating it from the other streams regardless of its priority. Due to this isolation, changing its priority would not yield different results.

Instead, results from the multi-priority variant tested with Scenario 3 are more insightful. On the source side, the scenario is identical to Scenario 7: The dependent stream and evaluation stream originate from the same source, while the independent stream originates from another, as depicted in Figure 6.6. The difference lies at the sink: All three streams terminate at the same destination and thus share the same egress port.

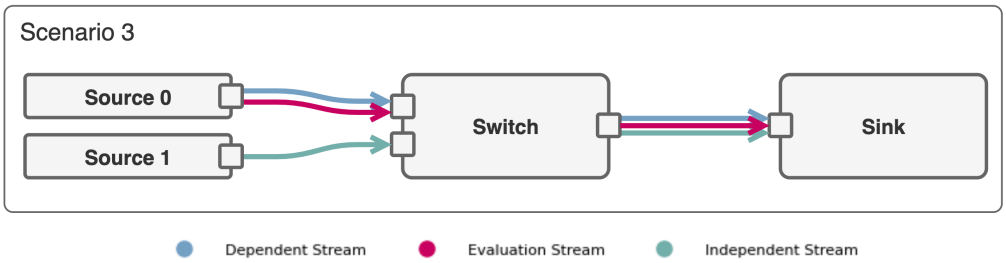
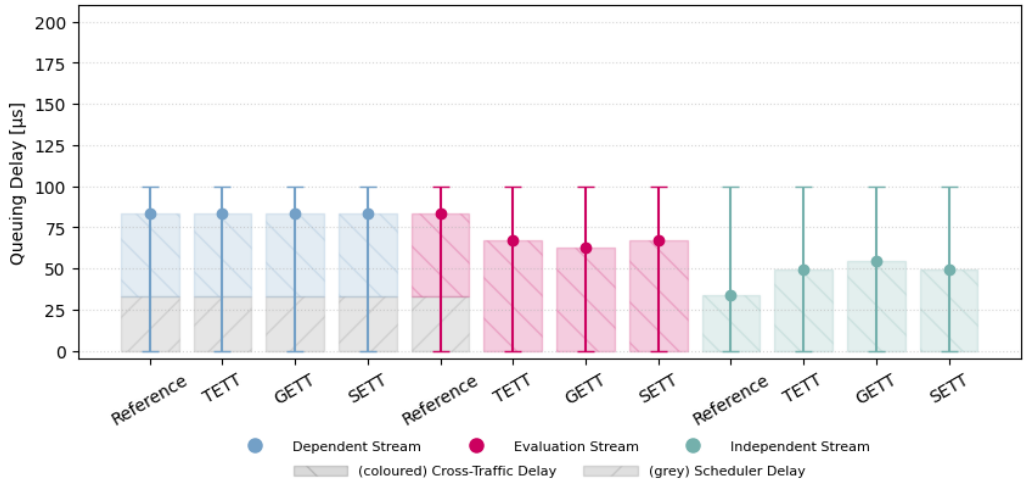


Figure 6.6: Topology for Scenario 3. It features a shared source for the dependent and the evaluation stream, as well as an independent stream, originating from a separate source. All three streams terminate at a common sink.

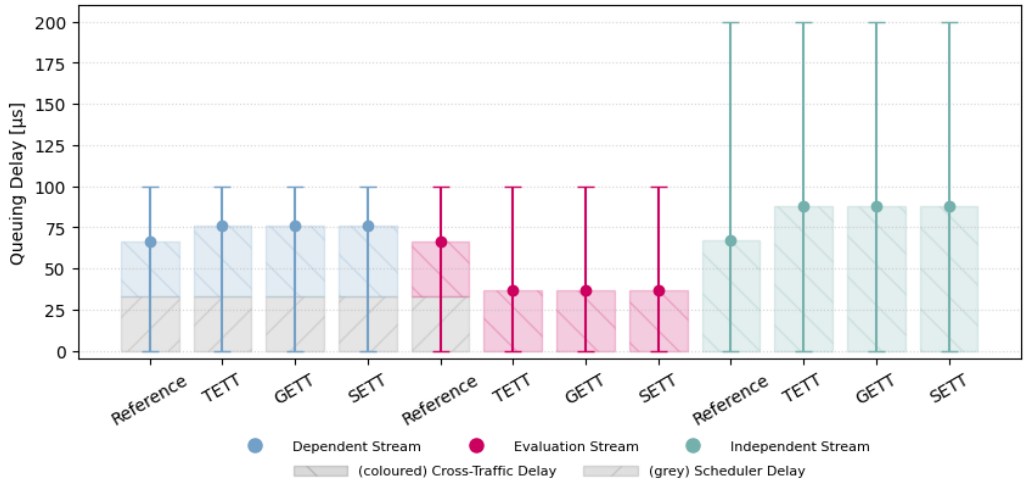
Now that the lower-priority independent stream is transmitted over the same link as the other streams, it is susceptible to cross-traffic delay. Therefore any impact Non-ATS traffic has on lower-priority traffic can be measured.

Due to the aforementioned elimination of shaping gaps, an increase in queuing delay for independent frames specifically is expected in the multi-priority variant.

Observed Results Multiple Priorities: The results from the equal-priority configuration for Scenario 3, displayed in Figure 6.7a, align with the previously observed trends. The queuing delay of the dependent stream remains unaffected by different NETT strategies applied to the evaluation stream. This behaviour is expected, as the dependent and evaluation frames are sequentialised at the source. In contrast, the independent stream, being unsequentialised, exhibits an increase in average queuing delay. This is due to the fact that since evaluation frames are not delayed as much anymore, they tend to cause a little more cross-traffic delay to the independent frames.



(a) Equal-Priority Configuration



(b) Multi-Priority Configuration

Figure 6.7: Average queuing delays and corresponding jitters for CFG-2, Scenario 3.

Interestingly, the independent stream displays a comparatively low average queuing delay in the reference setup for the equal-priority configuration (Figure 6.7a). This is attributed to its dedicated source, which isolates it from the scheduling effects associated with shared ATS scheduler groups, as discussed in Section 6.1.2.

When comparing the results of the equal- and multi-priority configuration variants for Scenario 3, shown in Figure 6.7, both average and maximum queuing delays for the independent stream nearly double when its priority is reduced, aligning with expected behaviour.

However, this negative impact is not significantly amplified by the use of NETT strategies, which can be seen by the relatively minor change from the reference setup-results to the NETT setup-results for the independent stream, shown in Figure 6.7b. The observed difference is approximately 20 µs, about one-fifth of a single frame transmission time. More significant is the average improvement for the evaluation stream: NETT strategies nearly halve the queuing delay, primarily by removing the scheduler delay caused by the scheduling side effects, when using ATS for multiple streams from a shared upstream node, as detailed in Section 6.1.2.

From these observations, it can be concluded that under the nominal traffic conditions of CFG-2, the impact of NETT strategies on lower-priority streams is less pronounced than initially anticipated. At the same time, clear benefits are observed for higher-priority Non-ATS traffic. However, these findings should not be interpreted as a definitive dismissal of the hypothesised issue of closed shaping gaps. Rather, the results suggest that the effect is not as pronounced in applications characterised by repetitive, low-demanding traffic patterns.

6.2 Evaluation at Maximum Load

Configuration CFG-10 serves as a stress test for the NETT strategies, featuring full link saturation to simulate peak network load conditions. Evaluating system behaviour under such conditions is beneficial for assessing whether the NETT strategies introduce unintended performance limitations or whether they can uphold the latency guarantees even under maximum load. Therefore, this type of evaluation contributes to a deeper understanding of the potential risks associated with these strategies, enabling a more informed assessment of their applicability in real-world deployments.

6.2.1 Configuration Overview

Configuration CFG-10 features a relatively small frame size of 125 B for the evaluation stream, resulting in transmission times of 10 µs. The remaining streams are configured with frame sizes ten times larger. Despite this, the evaluation stream is configured to consume as much or even more bandwidth than the other streams combined. This asymmetry creates a setup in which the system is exposed to a constant load generated by frequently transmitted evaluation frames, interspersed with sporadic cross-traffic from the larger frames of other streams.

Both the independent stream and the dependent stream are allocated a bandwidth of 25 Mbit/s each, enforced via their respective ATS schedulers. The evaluation stream is configured to utilise the remaining unallocated bandwidth.
In order to ensure that the link between the switch and the shared source of the evaluation stream and the dependent stream is fully saturated, two configuration variants are defined. This is because in some configurations, this link is shared by all three streams but in others the independent stream is forwarded using a different link to a separate source. Depending on this, the remaining bandwidth to be used by the evaluation stream is either 75 Mbit/s or 50 Mbit/s. If the independent stream shares a common sink with the other streams, the variant with 50 Mbit/s for the evaluation stream is used. For all others, 75 Mbit/s is utilised by the evaluation stream, as the dependent stream is the only other stream sharing the link between the switch and their sink.

Lastly, the start times of the streams are offset from another to spread out the load more evenly and to avoid unintended bursts due to aligned send intervals.

6.2.2 Results

Expected Behaviour: The configuration ensures that the arrival times of the frames match the configurations of the ATS schedulers, so that the credits of the ATS schedulers should have replenished just as the next frame arrives. Therefore, no scheduler delays are expected for scenarios with separate sources and, based on previous results, very limited queuing delays for all streams can be expected for scenarios with shared sources.

Furthermore, while assumed to be low, the queuing delay for the evaluation stream is expected to be higher than that of the other streams. This is due to the different frame sizes: Since evaluation frames have a transmission time of 10 µs and are not expected to

accumulate at the switch, the average cross-traffic delay induced by evaluation frames on frames of the dependent and independent streams is expected to be in the order of tens of micro seconds.

The other frames have transmission times of 100 µs per frame. Because both other streams have these large frames and because the evaluation stream transmitts frames frequently, the evaluation stream is likely to face higher queuing delays.

In contrast, the send intervals of the dependent and the independent stream are spaced out so that their frames are less likely to interfere with another.

Observed Results Shared Source: Figure 6.8 shows the results for Scenario 3 (topology illustrated in Figure 6.6). As expected, the average queuing delay of the evaluation stream is about one transmission time of one frame of the other streams. Furthermore, the cross-traffic delay in the independent stream, that arrives at a separate ingress port, is higher when employing the NETT strategies than when the evaluation stream is ATS-shaped in the reference setup.

In the reference setup, the evaluation stream experiences significant scheduler delay, which is due to the scheduling side effects associated with multiple ATS streams from a shared upstream node, as discussed in Section 6.1.2.

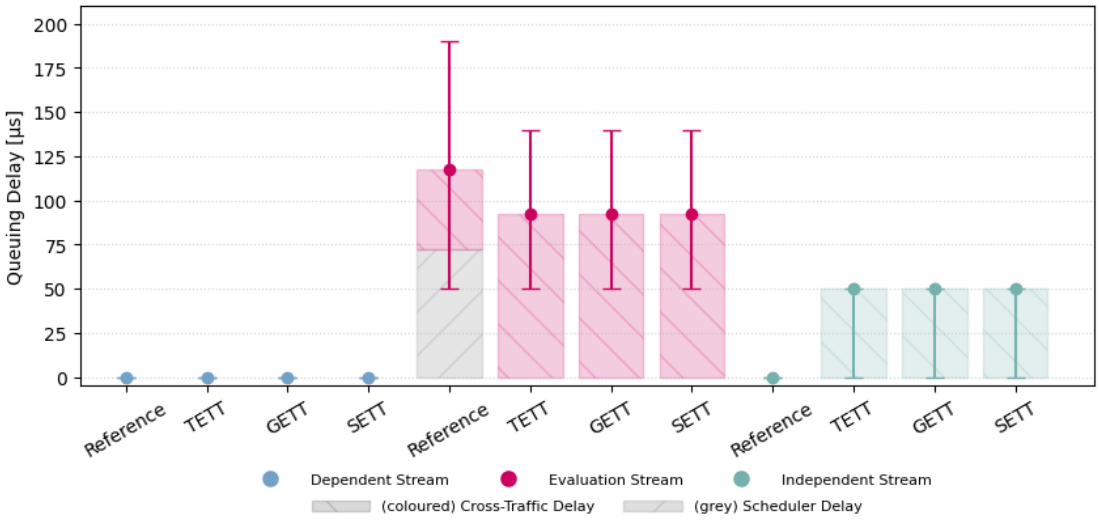


Figure 6.8: Average queuing delays and corresponding jitters for CFG-10, Scenario 3.

Interestingly, the average queuing delay of the evaluation stream does not change much when switching from it being ATS-shaped to being unshaped in the NETT setups, as can be seen in Figure 6.8. This is noteworthy since on average, more than half of the queuing delay in the reference setup is due to scheduler delay. Therefore, more significant reduction of queuing delay could be expected.

This shows that although the evaluation frames are delayed by their ATS scheduler in the reference setup, they would have been delayed by cross-traffic due to independent frames either way. The slightly lower queuing delay of the evaluation stream in the NETT setups and the higher queuing delay in the independent stream indicate that although some evaluation frames are queued before the independent frames, most of them are not. Each NETT strategy exhibited some form of indirect rate limitation in this configuration.

The queuing delay of zero for the dependent stream can be explained by the sequentialisation of dependent frames and evaluation frames in their shared source in conjunction with the deterministic ordering of frames in CFG-10. Furthermore, the configuration spaces out dependent frames and independent frames by misaligning their send intervals. Therefore, they can only impact another, if they are delayed significantly in the switch. This observation is therefore configuration dependent and of no further interest.

Observed Results Separate Sources & Multiple Priorities: Figure 6.10 shows the results for Scenario 6 (see Figure 6.9) with the multi-priority configuration variant of CFG-10. In Section 6.1.2 it was concluded that under low-demanding conditions, the impact of NETT strategies on lower-priority streams was less severe than expected.

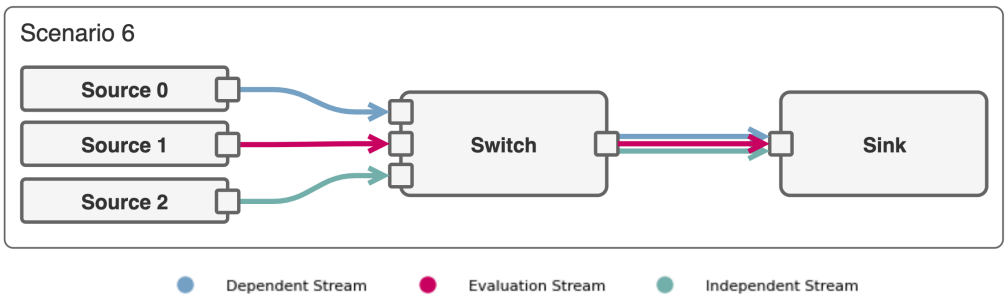


Figure 6.9: Topology for Scenario 6. It features a separate sources and a common sink for all three streams.

For CFG-10 this statement holds true. Since in Scenario 6 each stream originates from a separate source, the streams are not subject to the aforementioned ATS scheduling side effects associated with shared upstream nodes, discussed in Section 6.1.2. Since the configuration spaces out the send intervals, minimal interference between streams are measured.

Because the same queuing delays are measured for all setups, it can be concluded that, despite the high volume of traffic, NETT strategies do not cause unexpected side effects, as long as the ATS shaping mechanisms remain idle, supporting the results from CFG-2 presented in Section 6.1.2.

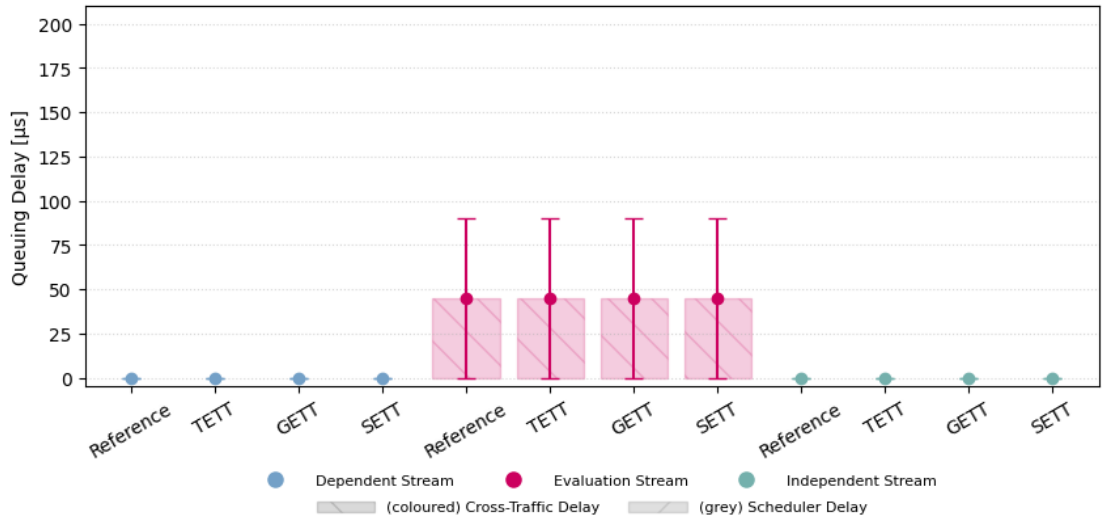


Figure 6.10: Average queuing delays and corresponding jitters for CFG-10 (multi-priority variant), Scenario 6.

6.3 Evaluation under Overload

Configuration CFG-10F is employed to assess system behaviour under slight bandwidth over-utilisation. In TSN networks, latency guarantees presuppose compliant system behaviour such that the enforced bandwidth limitations are not exceeded long term by the transmitted network traffic. Any violations of these assumptions and can lead to unbounded queuing delays and lost determinism [2, 10]. The measurements obtained under CFG-10F therefore serve solely to provoke potential failure modes of the NETT

strategies and to contrast their behaviour with ATS, thereby enabling a more informed evaluation of associated risks and limitations for real-world deployments.

6.3.1 Configuration Overview

As suggested by their designators, the configuration CFG-10F is a variation of CFG-10, introduced in Section 6.2.2. Both have identical parameters with the exception of the evaluation stream's bandwidth utilisation. The source of the evaluation stream is configured to transmit frames that exceed the configured maximum traffic rate by 1 Mbit/s, leading to an attempted link utilisation of either 76 Mbit/s or 51 Mbit/s.

The CIR parameter remains unchanged from the configured limits in CFG-10, disallowing bandwidth exceedance. The MRT is set to 1 ms for all ATS schedulers. Any frames assigned with a higher eligibility time are discarded, as explained in Section 2.2.3.2.

6.3.2 Results

Expected Behaviour Shared & Separate Sources: Due to the configured bandwidth limits, in the reference setup, the ATS scheduler of the evaluation stream enforces that traffic remains within the 100 Mbit/s link capacity, preventing congestions and protecting concurrent streams from the misbehaving evaluation stream.

In contrast, the NETT setups offload this responsibility to the NETT strategies, which are not designed to actively enforce bandwidth constraints, as stated in Section 4.2.3. As a result, total traffic should exceed the physical link capacity between the switch and the shared sink, resulting in excessive queuing delays.

Just like in CFG-10, the goal of CFG-10F is to saturate the link between the switch and the shared source of the dependent stream and evaluation stream. To induce an overload within the switch, the two streams can only share a source, if the evaluation stream is not configured to use 76 Mbit/s. Otherwise, their combined traffic would already exceed 100 Mbit/s, surpassing the link capacity between the source and the switch. In such a case, the traffic would be physically rate-limited before reaching the switch, keeping it within the bandwidth budget upon arrival. Although this over-utilisation increases

the end-to-end delay, the delay would occur at the source, not within the switch. Consequently, these setups do not effectively test the behaviour of NETT strategies under actual switch overload conditions.

Both scenarios evaluated under CFG-10 adhere to this restriction. The following section will present the results for CFG-10F for the same scenarios.

Observed Results Shared Sources: In Scenario 3, the independent stream originates from a separate source and converges in the switch with the dependent stream and evaluation stream, as shown in Figure 6.6, creating an over-utilisation and consequently a congestion within the switch.

The results shown in Figure 6.11 demonstrate that all three NETT strategies fail to shield concurrent streams from congestion caused by the evaluation stream. As a result, all streams experience significant cross-traffic delay in exceedence of 10 ms.

In contrast, the ATS scheduler used in the reference setup delays the rate-exceeding evaluation frames and drops them once the configured MRT is exceeded. Due to this mechanism, in the reference setup the maximum TSN queue length measured is limited to 10 frames. For the reference setups, the queue length exceeded 550 frames at the end of the simulation, continuously rising. This clearly highlights the shortcomings of NETT strategies in handling over-utilisation, while underlining the capabilities of ATS.

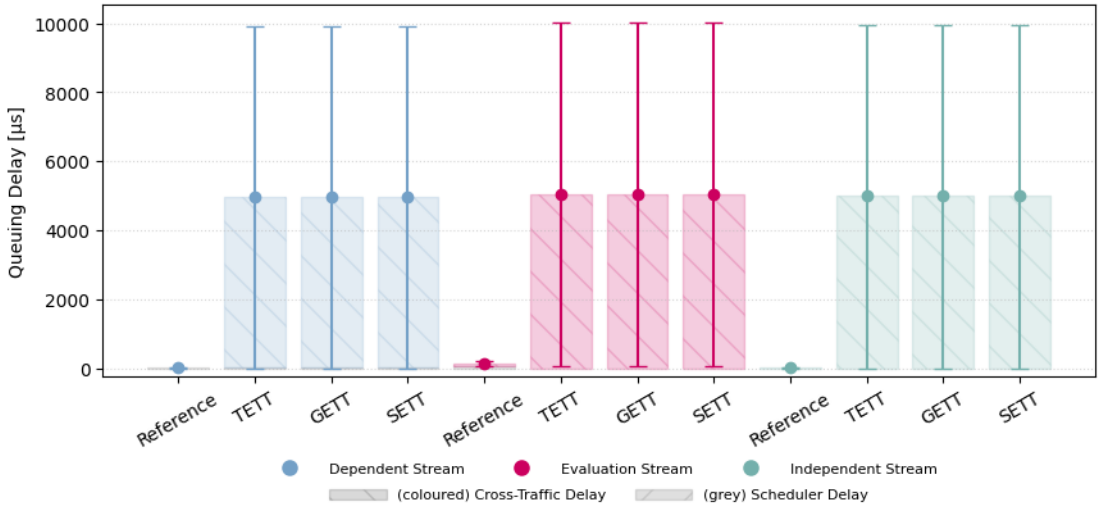


Figure 6.11: Average queuing delays and corresponding jitters for CFG-10F, Scenario 3.

That being said, while ATS drops the rate-exceeding frames before reaching the TSN queue, if CBSa would have been employed in the same setup, the lengths of the TSN queue would have been similar to those measured with NETT, since CBSa has no pre-queue rate limiting as explained in Chapter 2.

Observed Results Separate Sources: The results for Scenario 6 (Figure 6.9) with the multi-priority configuration variant are presented in Figure 6.12. Both Section 6.1.2 and 6.2.2 concluded that the impact of NETT strategies on lower-priority streams was less severe than expected. However, under overload, the situation changes significantly. As shown in Figure 6.12, the lower-priority independent stream experiences queuing delays exceeding 40 ms, which is approximately 400 times the transmission time of its frames. This delay exceeds acceptable thresholds by more than two orders of magnitude and can be interpreted as starvation: The stream is effectively deprived of timely access to network resources.

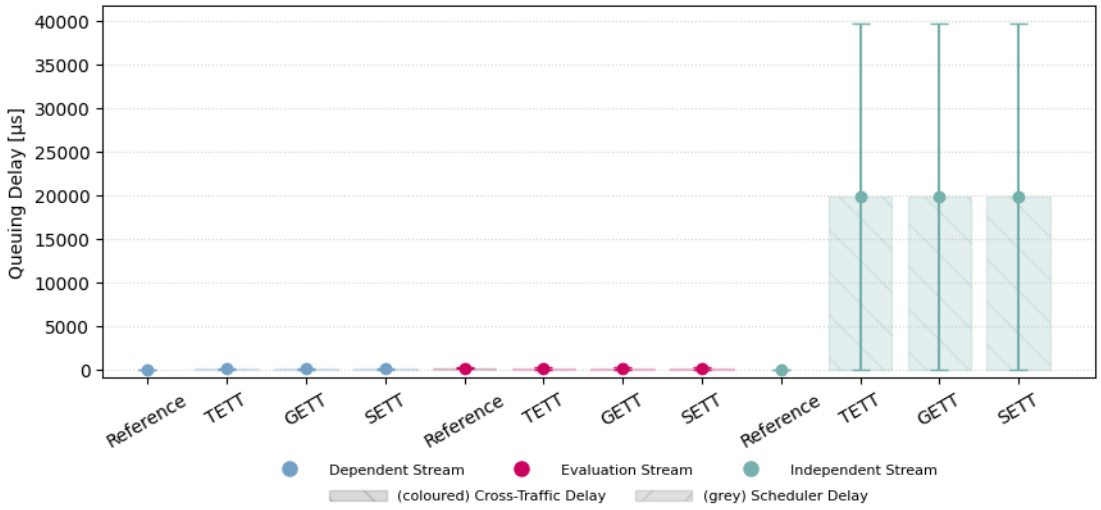


Figure 6.12: Average queuing delays and corresponding jitters for CFG-10F (multi-priority variant), Scenario 6.

In contrast, the reference setup, using an ATS scheduler, successfully delayed or dropped rate-violating evaluation frames, allowing lower-priority independent frames to pass without delay. This highlights the superior traffic isolation and delay guarantees of ATS compared to NETT strategies, which are unable to prevent starvation in overloaded conditions. However, these situations must be avoided by design and should not be a primary reason of avoiding the NETT strategies all together.

6.4 Evaluation under Bursty Conditions

Now that multiple configurations without any major randomness to the traffic have been presented, a configuration with less monotonous send behaviours is shown. It is meant to give an insight into how the NETT strategies perform for more realistic traffic patterns, involving jitters and drift. It is evaluated, whether the effects seen in previous results translate to less predictable situations.

6.4.1 Configuration Overview

For each stream, CFG-9 features bursts of five to six frames in immediate succession. The frame sizes of 625 B equate to transmission times of 50 µs per frame. Each burst is followed by a break in which no frames are sent, so that the average bandwidth usage of each stream equates to roughly 25 Mbit/s.

The ATS schedulers are configured with a CIR of the same 25 Mbit/s and a CBS of double the frame size. This allows for two frames of the same stream to become eligible for transmission right after another, which reduces the maximum scheduler delays experienced by each stream while potentially increasing the cross-traffic delay they create for concurrent streams. The MRT is set to account for the maximum scheduler delay.

As mentioned, each burst consists of five or six frame. This variation is due to a jitter that is applied to the burst duration, explained in Section 5.8.3. This jitter is configured to cause drift, so that the bursts are sent in different relative timings to the other streams throughout the course of the simulation.

Although the jitters and drifts are randomly applied without a bias, to mitigate possible statistical errors, the average results of three random repetitions are presented. This is sufficient given the high number of variations of relative timings tested in each repetition.

6.4.2 Results

Expected Behaviour: Anticipating network performance in higher complexity configurations is nearly impossible. However, deriving from previous results, the queuing delays are not expected to differ much between the reference setup and the NETT setups. Most

results presented, apart from the overload configuration, showed at most minor deterioration when employing NETT strategies while some even showed significant improvements for streams of shared TSN queues.

Observed Results: The results show multiple effects of interest. The four most significant will be presented in this section.

Observed Results Separate Sources & Separate Sinks: Firstly, the results for Scenario 8 mostly align with previous results and conform to the expected behaviour. The scenario features three separate sources and two sinks, as shown in Figure 6.1. The results presented in Figure 6.13 show that the queuing delays of the dependent and the independent stream are very similar throughout the four setups. For the independent stream this is to be expected, as the stream is isolated from the others. For the dependent stream the results are of more significance, as they indicate that the difference of delay imposed on concurrent streams is relatively minor for ATS and Non-ATS streams.

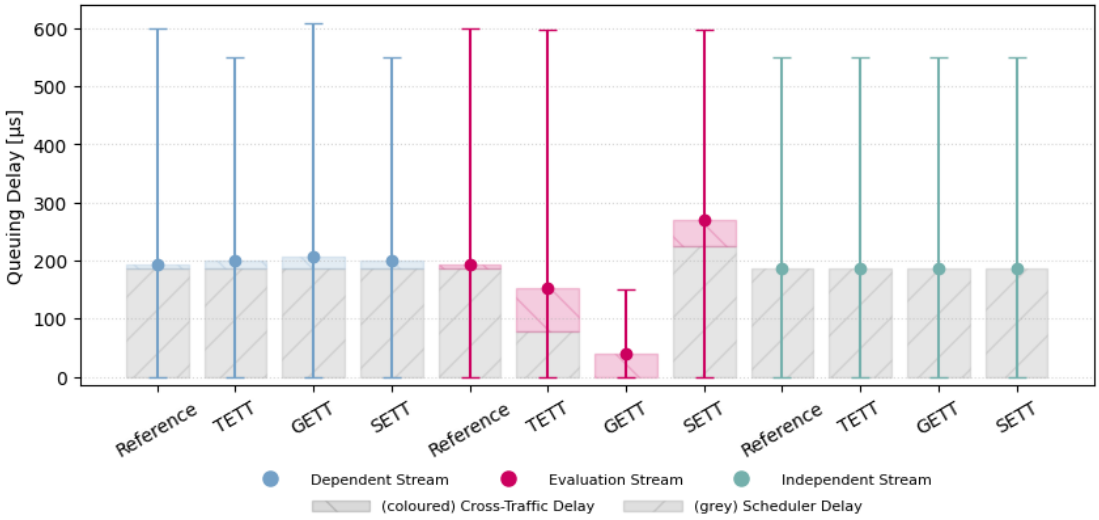


Figure 6.13: Average queuing delays and corresponding jitters for CFG-9, Scenario 8.

The most obvious difference is within the queuing delays of the evaluation stream, when comparing the NETT strategies themselves. As each stream is part of another ATS scheduler group, the evaluation stream experiences no scheduler delay when using GETT. In contrast, when using SETT, the evaluation stream experiences more scheduler delay on average than any of the other streams. The average delay TETT causes to the evaluation

stream is almost exactly the intermediate between the other two strategies, while still outperforming ATS.

Using TETT and SETT also decreases the maximum queuing delay for the dependent stream, which aligns with previous results.

Observed Results Separate Sources & Common Sinks: The results for the multi-priority variant tested with Scenario 6 (Figure 6.9), shown in Figure 6.14, confirm most previous findings as well. They show the same lower delay for the evaluation stream when using GETT and overall little variation in average delays between the setups. However, the maximum queuing delay for the lower-priority independent stream increases by 25 % when using TETT and by 50 % when using SETT, compared to GETT and the reference setup.

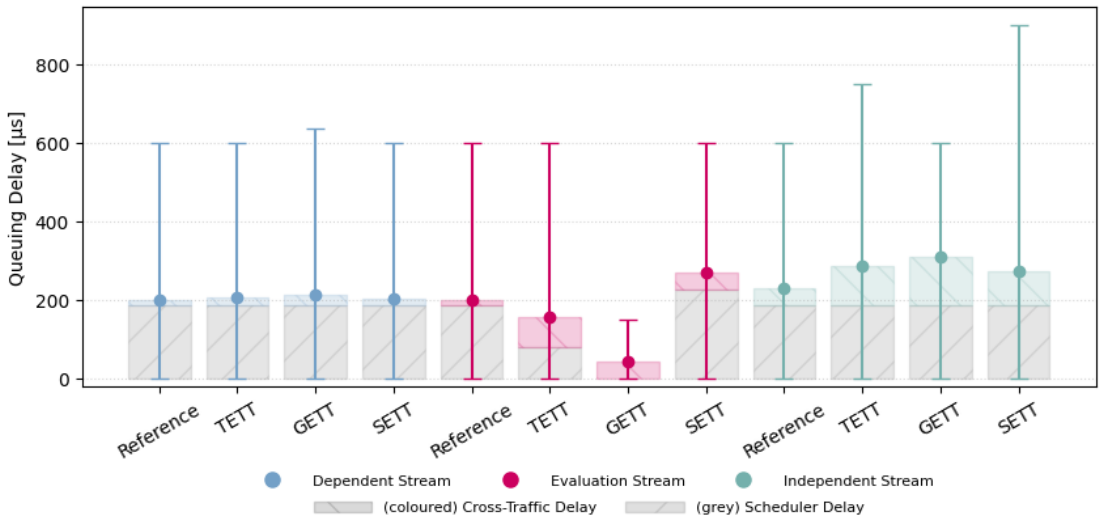


Figure 6.14: Average queuing delays and corresponding jitters for CFG-9 (multi-priority variant), Scenario 6.

For SETT this can be explained intuitively. Suppose a lower-priority independent frame is the last to become eligible for transmission. Now, all evaluation frames arriving at the switch are assigned the same eligibility time as this independent frame, making them become eligible at the same time. Due to their higher priority, the evaluation frames will be selected first by SPQ, causing further cross-traffic delay for the lower-priority independent frame. This helps to understand why lower-priority ATS streams may experience higher queuing delays with SETT than with any other NETT strategy.

Note that the shown results are just random samples. The presented percentages may not be used as constants for any calculations and the differences in maximum queuing delays between NETT strategies differ depending on traffic patterns and topologies.

The higher maximum queuing delay measured when using TETT is likely due to a higher number of accumulated frames compared to GETT and ATS. As long as some other frame is queued in the same shared TSN queue, any incoming evaluation frames are queued behind the last frame and become eligible for transmission at the same time. This is similar to the effect described above for SETT. However, with TETT, evaluation frames are not affected by lower-priority frames since they are not part of the same TSN queue. Therefore, there is no correlation between the eligibility times of evaluation frames and independent frames when using TETT, reducing the maximum queuing delay for the independent frames compared to SETT. Again, these results depend on the specific traffic patterns and are subject to change under different conditions.

Observed Results Shared ATS Scheduler Group: Until now, the results were consistent with expectations derived from earlier findings. However, this consistency does not hold for Scenario 5. As illustrated in Figure 6.15, the scenario features a shared source for both the dependent stream and the independent stream. With that, using the equal-priority variant of CFG-9, both ATS streams are placed in the same ATS scheduler group.

The results for Scenario 5, shown in Figure 6.16, reveal significant queuing delays for both the dependent and independent stream across all configurations. This includes the reference setup, where the evaluation stream is ATS-shaped. These findings strongly suggest that the delay originates from ATS itself, rather than from any particular NETT strategy.

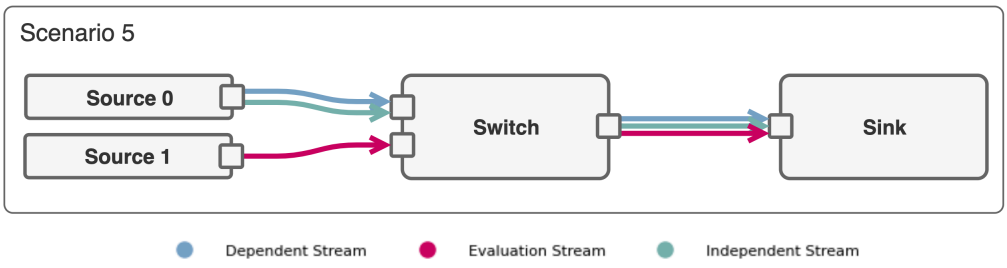


Figure 6.15: Topology of Scenario 5. It features a shared source for the dependent and the independent stream and a common sink for all three streams.

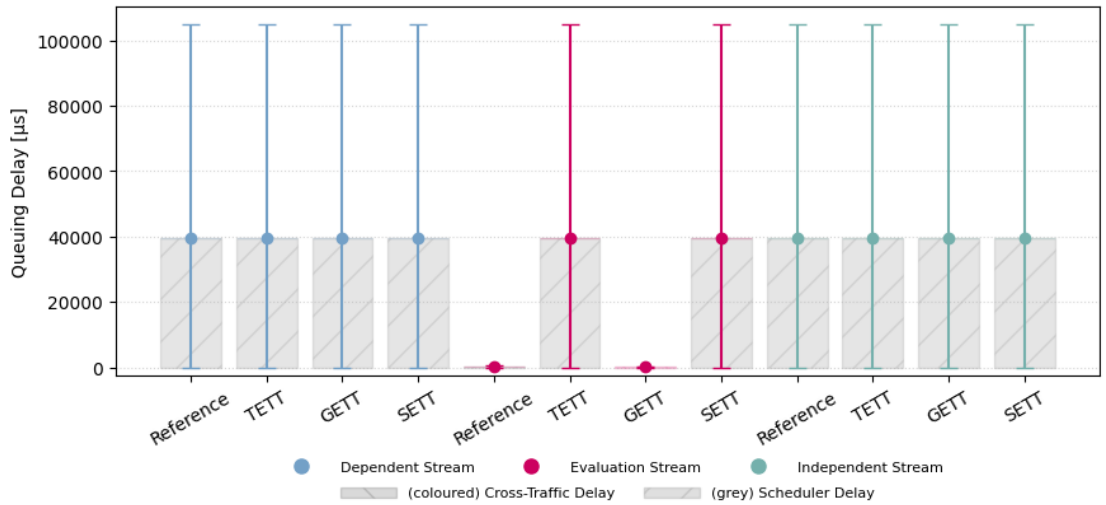


Figure 6.16: Average queuing delays and corresponding jitters for Scenario 5 using CFG-9. The dependent and independent stream share the same ATS scheduler group, resulting in substantial queuing delays due to scheduler interference. The effect significantly impacts SETT and TETT, while GETT remains unaffected.

This behaviour is closely related to the scheduling effects associated with shared upstream nodes, as previously discussed in Section 6.1.2. There, frames could experience scheduler delay due to varying transmission orders at the shared source.

In this case, the issue originates from the shared group eligibility time between the dependent stream and the independent stream. Generally, the longer a burst, the longer the last frame in that burst is delayed by ATS, as explained in Section 2.2.3.2.

Suppose a burst from one of the ATS streams (burst-0) is transmitted to the switch, immediately followed by a burst from the other stream (burst-1). In CFG-9, the last frame of burst-0 is delayed by 550 µs (see Section B). Just 50 µs later, the first frame of burst-1 arrives. However, due to the delay caused by burst-0, the shared group eligibility time is set 500 µs into the future, preventing the first frame of burst-1 from becoming eligible immediately and instead delaying it and the entire burst-1.

This delay affects more than just burst-1's transmission. It postpones when the ATS credit counter can begin recovering. By the time the next burst arrives, the credits may not have fully replenished, causing additional delays. A similar, albeit smaller, cascading effect was previously discussed in Section 6.1.2.

In theory, a sufficiently long idle period between bursts would allow the credit counter to recover. However, such idle times do not exist in CFG-9. As seen in Figure 6.16, maximum queuing delays exceed 100 ms, and queue lengths span several hundred frames. This occurs even though both the dependent and independent stream remain within their configured bandwidth limits.

Reducing the MRT could lower the maximum scheduler delay, but would lead to dropped frames without resolving the root cause. While network calculus confirms that worst-case latency guarantees for the overall priority class remain intact, the experienced maximum latency of individual streams still increases [25]. For more complex topologies, the described behaviour can lead to unbounded latencies, necessitating careful design [26].

Whether this issue affects a Non-ATS stream handled by a NETT strategy depends on both the topology and the specific strategy used. With SETT, it is sufficient for the problem to occur within the same node. For TETT, it must occur within the same shared TSN queue, and for GETT, within the same ATS scheduler group. This explains why, in Scenario 5, only the evaluation stream processed by GETT is unaffected, as evident by its significantly lower queuing delays shown in Figure 6.16.

From this, the NETT strategies can be ranked by their susceptibility to scheduling effects stemming from shared ATS scheduler groups. SETT is most susceptible, as it considers all ATS schedulers within a node, followed by TETT, which covers all traffic of the same TSN queue and finally GETT, which restricts scope to ATS streams sharing both priority and ingress port.

Although GETT is statistically the least vulnerable, there are situations where it is affected while TETT is not. Furthermore, when the issue does manifest, it may destabilise the system through delayed or dropped ATS frames. As such, the presence of this effect is concerning regardless of whether it impacts a Non-ATS stream or not.

6.5 Evaluation of a Worst-Case Scenario

While the previous sections analysed situations where delays arose due to exceeding the physical link capacity or due to ATS anomalies, the following section presents a case in which comparable delays occur due to the NETT strategies themselves, despite all traffic adhering to long-term bandwidth restrictions.

Figure 6.17 illustrates the issue by showing the timeline for CFG-12 in Scenario 3 (Figure 6.6), including the arrival times, eligibility times and corresponding departure times.

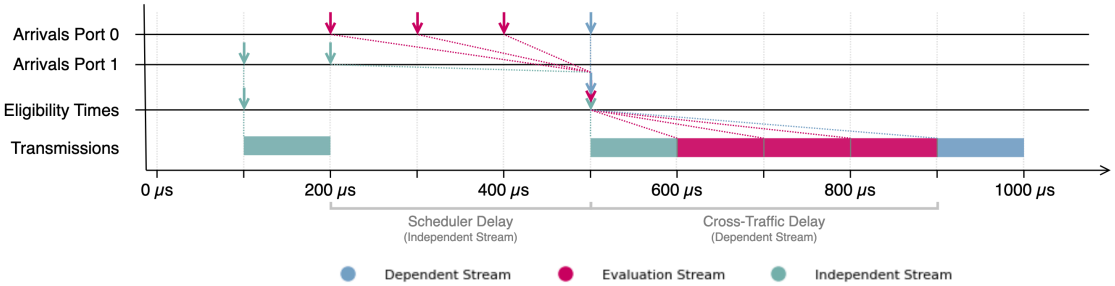


Figure 6.17: Arrival, calculated eligibility times and transmissions for CFG-12, Scenario 3 using TETT, illustrating queue build-up and how frames are released in a burst due to simultaneous eligibility.

The configuration is set-up so that two independent frames arrive at the switch in quick succession, momentarily exceeding the CIR enforced by their ATS scheduler. As a result, the second independent frame experiences a scheduler delay of 300 µs. During this delay, three additional evaluation frames arrive and are placed in the TSN queue behind the scheduler-delayed independent frame, in accordance with the TETT strategy. All four frames become eligible for transmission simultaneously. A dependent frame then arrives just as this burst is released, causing it to be queued last and incurring a cross-traffic delay of 400 µs.

In the reference setups where ATS is used for the evaluation stream, this problem does not occur. With ATS, the evaluation frames are spaced out against another and which consequently limits the delay for the dependent frame to between one or at most two transmission times.

This configuration illustrates how NETT strategies can lead to accumulation of numerous frames. Rather than spacing frames evenly over time like other traffic shaping algorithms,

these strategies can cause many frames to become eligible for transmission at once, leading to sharp spikes in delays across affected streams, as well as creating bursts that may negatively affect traffic in downstream nodes, as discussed in Section 2.2.3. These delays can cascade and impact seemingly unrelated streams.

The size of the maximum burst depends on the triggering ATS stream, which is the independent stream in this case: Reducing the CIR of its ATS scheduler increases the shaping gap inserted between successive independent frames linearly. A longer scheduler delay naturally increases the amount of time evaluation frames have to accumulate. And the more evaluation frames accumulate, the higher is the cross-traffic delay experienced by the dependent frame arriving last.

Consequently, when a NETT strategy is employed, lowering the allocated bandwidth of a single ATS stream can delay an unrelated stream that belongs to a different ATS scheduler group. The results of this correlation are illustrated in Figure 6.18.

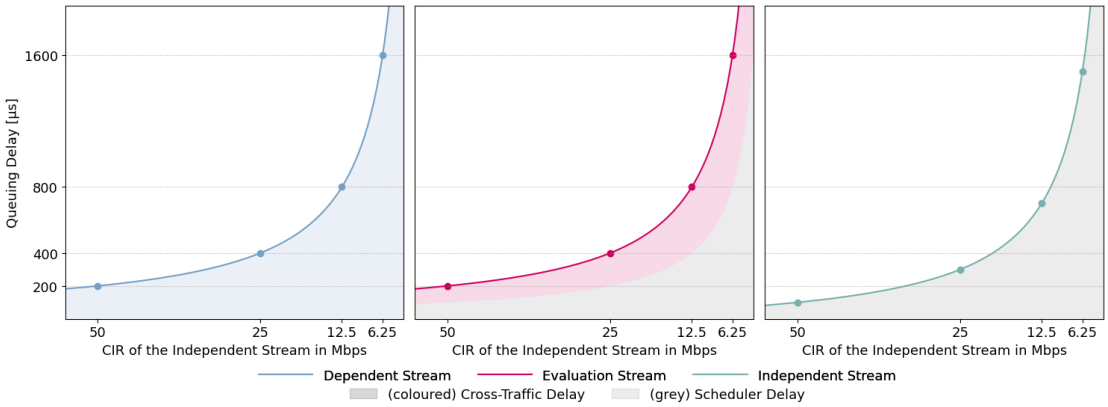


Figure 6.18: Maximum queuing delay observed in CFG-12 Scenario 3 under varying CIRs for the independent stream, illustrating the proportional increase in delay for all streams with reduced bandwidth allocated to the independent stream.

Similar to previously, whether this issue arises does not only depend on the specific traffic patterns but the employed NETT strategy and the network topology as well. For SETT, mere co-location of the Non-ATS and the triggering ATS stream on the same node suffices for the problem to arise. This allows such interference in multi-priority configurations, even for Scenarios 7 or 8 (Figure 6.3, Figure 6.1), where the interfering independent stream is both isolated and of lower priority, effectively creating a form of priority inversion: The lower-priority independent stream causes the higher-priority dependent stream to be delayed, due to the SETT-processed evaluation stream.

It is important to emphasise that this is primarily a theoretical issue. The degree of frame accumulation is inherently bounded by the configuration of the ATS schedulers and the burst characteristics of the triggering ATS stream, in conjunction with the link capacity and frame size and thereby maximum number of Non-ATS frames that can accumulate in a given time. In the example provided, the MRT was deliberately configured more permissively than would typically be used in a practical deployments for similar streams. Furthermore, if a stream is expected to transmit two frames in close succession followed by a long pause, it is reasonable to adjust the CBS to accommodate such small bursts without introducing undue delay. Either reducing the MRT or tuning the CBS to match the independent stream's bursting behaviour would mitigate this issue significantly.

With that being said, the issue may also be triggered by any other situation where a frame is significantly delayed. NETT strategies clearly have the tendency to amplify and contribute to congestion. Therefore, certain configurations using NETT strategies can result in significant and seemingly unpredictable delays. While bounded under well-designed configurations, this illustrates an intrinsic vulnerability of NETT strategies in preserving deterministic latency guarantees.

7 Conclusion

This thesis investigated strategies for combining ATS and Non-ATS traffic in shared TSN queues, addressing the so-called Non-ATS Conflict. This conflict occurs when unscheduled frames, lacking an eligibilityTimeTag, are introduced into queues governed by an ATS TSA. Such configurations, being neither explicitly defined nor prohibited by the TSN standard IEEE 802.1Q, result in undefined system behaviour.

Summary of Contributions: To address this ambiguity, three candidate strategies were proposed: TETT, GETT, and SETT. These approaches resolve the conflict by assigning eligibility times to Non-ATS frames, derived from either concurrent ATS traffic or active ATS schedulers. This makes the Non-ATS traffic compatible with standard ATS queues.

Using a microbenchmarking methodology, the behaviour and limitations of each strategy were evaluated under controlled conditions.

Results and Limitations: Results showed that all three strategies can successfully mitigate the Non-ATS Conflict, often with minimal performance degradation to concurrent ATS streams, and in some cases outperform standard ATS shaping by avoiding scheduling anomalies.

Despite these promising results, several limitations were identified. Firstly, none of the proposed strategies actively enforce bandwidth constraints for Non-ATS traffic. They rely on compliant behaviour, making them vulnerable to congestion or starvation of lower-priority streams. While this is a known issue even in other mechanisms like CBSa, it remains a relevant operational risk.

Secondly, pathological cases showed severe frame accumulation and synchronised eligibility times, which led to large bursts and latency spikes, occasionally exceeding expected bounds by several orders of magnitude. Although external factors might theoretically

bound such effects, they pose a significant challenge to maintaining latency guarantees. Unlike standard TSN mechanisms, which are supported by formal guarantees, the NETT strategies currently lack deterministic worst-case assurances. While it may be possible, albeit difficult, for future work to derive analytical latency bounds for networks employing NETT, such proofs are currently unavailable. Consequently, these strategies are unsuitable for safety-critical applications at this stage.

That said, even standardised ATS-based systems require careful configuration to avoid unintended interactions, as became evident by scheduling anomalies among ATS streams within the same ATS scheduler group.

Strategy Comparison: Should the use of NETT strategies be considered for non-critical applications, extensive testing is strongly recommended. The evaluations demonstrated that their behaviour is highly sensitive to network topology and traffic patterns.

Each strategy has its distinct strengths and weaknesses. TETT and SETT introduced no additional delays under near-ideal conditions, whereas GETT did. This drawback of GETT is relatively minor and offset by its superior implementability. Unlike TETT and SETT, GETT may be realised on existing hardware through appropriate configuration, avoiding the need for architectural changes. Even if that is not feasible, GETT will most likely require the smallest change for implementation among the NETT strategies.

SETT and TETT showed similar performance in most benchmarks, which could favour SETT due to its simplicity regarding implementation. However, SETT carries a risk of priority inversion. Its dependence on the eligibility times of lower-priority frames can affect the scheduling of higher-priority streams within the same shared TSN queue. Consequently, SETT should be avoided.

Considering these findings, the most promising overall candidate is GETT.

Future Work: The findings in this thesis are unlikely to be exhaustive. Further investigation is needed to understand how GETT interacts with high-priority streams, Time-Aware Shaping (TAS) traffic and environments with many concurrent streams and diverse scheduling algorithms.

Hybrid approaches, incorporating external rate-limiting mechanisms such as PSFP, could improve system robustness. Alternatively, preventive mechanisms, such as the Non-ATS

Avoidance strategies discussed in Chapter 4, may help eliminate the risk of misconfigurations leading to Non-ATS Conflicts. These too require further validation and must be designed carefully with robust logging mechanisms in order to not conceal mistakes but rather strengthen system resilience.

Final Remarks: While NETT strategies, and GETT in particular, offer a viable path toward resolving the Non-ATS Conflict, they are not without limitations. They provide an effective interim solution for non-critical systems, but long-term deployment will require mechanisms with stronger formal guarantees and enforcement capabilities. Continued exploration of alternative or hybrid approaches remains an important direction for future work. Alternative mechanisms may ultimately prove to be both robust and reliable solutions for the Non-ATS Conflict.

Glossary

A

Asynchronous Traffic Shaping A traffic shaping algorithm that determines the eligibility times of frames based on a credit counters and a group eligibility time. ATS consists of two individual parts, where the ATS schedulers realise the computational part of the overall traffic shaping operation and the ATS TSAs realise the execution based on the assigned eligibility times. Since ATS schedulers operate independently of TSN queues, a single ATS scheduler can feed frames into multiple different TSN queues

ATS scheduler The first part (computational part) of the Asynchronous Traffic Shaping process. For each arriving frame, ATS schedulers calculate and assign the eligibility time of that frame, based on their credit counter and the group eligibility time of the ATS scheduler group. It attaches the assigned eligibility times to the frame for the ATS TSA in the form of an eligibilityTimeTag

ATS scheduler group ATS schedulers are organised in ATS scheduler groups. There is one ATS scheduler group per ingress port and priority. The purpose of a group is to constrain the assigned eligibility times of all ATS schedulers by the same group eligibility time

ATS TSA The second part of the ATS shaping process which releases the frames at their predetermined eligibility times calculated by the ATS scheduler, which is stored in the eligibilityTimeTag. This is a Transmission Selection Algorithm

ATS queue A specific TSN queue that is configured to use the ATS TSA and queues frames that contain eligibilityTimeTags assigned by an ATS scheduler

B

bandwidth The amount of data that can be transmitted over a communication channel per unit of time, typically measured in bit/s or its multiples (e.g., Mbit/s). In practice, bandwidth may refer to the actual usage or to the theoretical maximum depending on context

burst A sudden spike or increase in the rate of data transmission over a network within a short period. Bursts can lead to temporary congestions or buffer overflows and can have a significant impact on the delay of concurrent streams

C

CommittedBurstSize A parameter of ATS schedulers that determines the initial value of the ATS's credit counter in bit. It also acts as the upper limit of credits which cannot be exceeded. This means that the CommittedBurstSize must be at least equal to the size of the largest frame scheduled by the ATS scheduler

CommittedInformationRate A parameter of ATS schedulers that determines the constant rate at which an ATS's credit counter accumulates credits in bit/s

congestion Congestions occur when the rate of data traffic exceeds the available network resources, such as bandwidth and the number of frames buffered increases continuously, as the node cannot keep up with the incoming frames. This leads to increased delays and overall performance degradation and in the worst-case packet loss due to buffer overflows

credit counter Credit counters are used by traffic shaping algorithms, specifically by CBSa and Asynchronous Traffic Shaping. They track the available credits [bit]. In CBSa, when the credit counter is positive or zero, the CBSas becomes eligible for transmission if not empty. In ATS, the eligibility time of frames are calculated by the ATS scheduler based on the credit counter

Credit-Based Shaper algorithm A traffic shaping that determines the eligibility of TSN queues based on a credit counters. The Credit-Based Shaper algorithm is a Transmission Selection Algorithms. It operates solely on a single TSN queue

cross-traffic delay A component of queuing delay. It represents the time between a frame becoming eligible for transmission and its actual transmission start. This

delay is caused by other frames already in theTSN queue being transmitted first, including those from the same stream

D

delay A general term for the time it takes for data to be processed or transmitted within a system. In networking, delay can refer to various components such as physical signal propagation, transmission, queuing and processing delay

dependent frame A frame of a dependent stream

dependent stream An ATS stream that shares its TSN queue with the evaluation stream. Its behaviour is affected by the presence of the evaluation stream and therefore by the NETT strategies behaviour

destination A specific type of sink that is explicitly addressed as the final recipient in the header of a frame. The destination represents the intended endpoint of a transmission as determined by addressing information within the frame

downstream The direction of a stream from a node towards the destination(s). It may refer to all intermediate nodes a frame has to pass through in order to reach its destination, starting at the next node relative to where the frame currently is

drift A gradual deviation of a stream's nominal schedule, defined by its start time and send interval. Drift occurs when jitters accumulate over time, particularly if they exhibit a consistent bias

E

egress port A physical network port through which frames exit a node towards their destinations. A physical port may be bidirectional and support both ingress and egress functionalities. The egress port of a TSN node contains the TSN queues

eligibility time The specific moment in time when a frame or TSN queue becomes eligible for transmission. For ATS shaped frames this time is calculated by an ATS scheduler before queuing them. TSN queues managed by other TSAs become eligible for transmission based on the operations of the Transmission Selection Algorithm

eligibilityTimeTag The eligibilityTimeTag is a node-local field assigned to any frame subject to ATS shaping. It stores the eligibility time of the frame

eligible for transmission A frame or TSN queue is considered eligible for transmission when it meets all necessary conditions defined by the specific Transmission Selection Algorithm supported by the TSN queue. For CBSa, a TSN queue is eligible if it contains one or more frames and the credit counter is non-zero. For ATS, a frame is eligible if the assigned eligibility time is earlier than or at the current time

evaluation frame A frame of an evaluation stream

evaluation stream The stream that is scheduled using an ATS scheduler in the reference setups and left unscheduled in the NETT setups. It is the stream to which the NETT strategies are applied and thus serves to evaluate their effectiveness

F

frame The protocol data unit for Ethernet at the data link layer (layer 2) of the ISO/OSI reference model

G

group eligibility time All ATS schedulers associated with the same ATS scheduler group share a common state variable called group eligibility time. ATS schedulers may not assign eligibility times lower than their group eligibility time. ATS schedulers update this shared group eligibility time variable if they assign an eligibility time higher than the one currently stored in the variable, which means the group eligibility time always holds the highest eligibility time assigned by any ATS scheduler within the ATS scheduler group. This process ensures a non-decreasing order of eligibility times of successive frames associated with a single group, which permits frames of the same group to be forwarded in FIFO order

Group Eligibility Time Tagging A specific NETT strategy. When a Non-ATS frame arrives at a TSN node, it is tagged with an eligibilityTimeTag containing the current group eligibility time, as determined by the ATS scheduler group, responsible for the ingress port and priority combination of the Non-ATS frame. If the group eligibility time is earlier than the current local system time of the TSN node, the local system time is used instead

H

hop A single segment along the path of a frame, representing its transmission between two directly connected nodes. A hop occurs each time a frame is forwarded from one node to another on its way downstream toward a sink

I

IEEE 802.1Q The base standard for TSN, revision 2022 [2]

independent frame A frame of an independent stream

independent stream An ATS stream that competes for shared network resources but is not required to share immediate contention points with the evaluation stream. It is used to evaluate indirect interference and prioritisation effects

ingress port A physical network port through which frames enter a node. While typically the counterpart to egress ports, a physical port can be bidirectional and support both ingress and egress functionalities

Internal Priority Value A node-local priority value that can differ from the priority indicated by the frame's PCP header field. It can be used to alter the forwarding behaviour of the current node without effecting any downstream nodes

J

jitter The difference between the maximum and minimum latency or delay experienced by any frame of a specific stream

L

latency The total time required for a frame to travel from the source to the destination. Latency typically includes all forms of delay

link A physical or logical connection between two directly connected nodes, where data flows from an egress port on the transmitting node to an ingress port on the receiving node

link capacity The maximum achievable data rate of a link, typically expressed in Mbit/s. It defines the upper bound for bandwidth of the link

M

MaximumResidenceTime A parameter of ATS schedulers that limits the maximum delay the ATS scheduler can add to a frame. Any frames where the associated eligibility time exceed this value are dropped

N

NETT setup A specific setup where the evaluation stream configured as Non-ATS traffic and processed by a NETT strategy. The results from these setups are compared to those of the corresponding reference setup

node Any device within a network that can send, receive, or forward data. The term node can be used to describe all devices such as routers, switches, servers, or any other hardware participating in network communication

Non-ATS Refers to any traffic that is not scheduled by an ATS scheduler but, based on its priority and egress port, is to be queued together with ATS frames in a single TSN queue. These frames typically do not carry an eligibilityTimeTag but can be assigned one by the use a NETTs to work with regular ATS queues

Non-ATS Conflict A condition in which an unscheduled frame, based on its priority and egress port, is directed to an ATS queue despite lacking an eligibilityTimeTag, resulting in undefined queuing behaviour. Such frames are called Non-ATS frames

Non-ATS Eligibility Time Tagging A specific subcategory of Non-ATS Handling strategies where the Non-ATS frames are tagged with an eligibilityTimeTag without the use of a ATS scheduler. The NETT strategies include TETT, GETT and SETT

Non-ATS Handling strategy There are multiple different possible strategies to mitigate the Non-ATS Conflict and combine ATS and Non-ATS traffic in shared TSN queues. All of these strategies are referred to as Non-ATS Handling strategies. There are three subcategories to this, one of which, namely NETT, is explored in this thesis

P

Per Stream Filtering and Policing A mechanism in IEEE 802.1Q that manages and prioritises network traffic on a per-stream basis, allowing the enforcement of policies like traffic rate limits

priority The importance level assigned per stream. In TSN networks, priority is stored in the PCP of the frames

Priority Code Point A 3-bit field in the IEEE 802.1Q VLAN header that specifies the priority of a frame, with values ranging from 0 (lowest priority) to 7 (highest priority)

Q

Quality of Service The service provided to network traffic which may include guarantees for delay, bandwidth, jitter or packet loss

queuing delay The total time a frame spends in a TSN queue of a node. It consists of both the time a frame waits to become eligible for transmission (scheduler delay) and the time it waits after becoming eligible due to the transmission of other queued frames (cross-traffic delay)

R

Real-Time Computer System A computing system, comprising both hardware and software, that is designed to meet strict real-time constraints

Real-Time System A system that must meet pre-defined timing constraints known as deadlines. The correctness of operations may depend not only on logical results but also on the time at which results are produced

reference setup A specific setup where the evaluation stream is assigned an ATS scheduler, turning it into an ATS stream. Reference setups avoid the Non-ATS Conflict and do not incorporate NETT strategies. They serve as a performance baseline, enabling comparison with NETT setups by maintaining identical conditions aside from the scheduling mechanism

S

scenario A simulation network topology designed to enable evaluation of the effectiveness of a specific NETT strategy in resolving the Non-ATS Conflict in a specific circumstance

scheduler A component that determines the point in time when frames are to be transmitted. The most important scheduler in the context of this thesis is the ATS scheduler

scheduler delay A component of queuing delay. It refers to the time a frame spends waiting in the TSN queue before it becomes eligible for transmission, as determined by the applied scheduling mechanism

shaper A mechanism that enforces the maximum transmission rate for streams or for all traffic of a specific priority. The most important traffic shaping algorithm discussed is ATS

shared TSN queue TSN queues that contain both ATS and Non-ATS frames

sink A node, or a logical component within a node, that consumes data from the network. It serves as the endpoint for transmitted packets or frames. In most contexts, "sink" and "destination" can be used interchangeably when referring to the final recipient of a frame, even though "sink" more generally describes any component that consumes data, regardless of if being the explicitly addressed destination or not

source A node, or a logical component within a node, that generates and transmits data into the network. The source of a frame describes the frame's origin

source shaping Traffic shaping that takes place either in the sending node itself, or at the ingress point of the stream into the network

starvation A condition where a stream is deprived of timely access to network resources, often because other processes are continuously prioritised over it

stream Describes a unidirectional set of packets or frames, typically identified by header fields (e.g. source MAC/IP/port, destination MAC/IP/port). In the context of this thesis, the term *stream* is used universally to refer to both scheduled and unscheduled traffic flows, regardless of whether they are explicitly managed by a scheduler and assigned a priority and Quality of Service or not

stream handle A node-local, numerical tag of a frame that indicates which stream it belongs to. It is assigned by the stream identification function upon arriving at the TSN node and is removed before passing the frame on to the next node

stream identification A filter that aims to identify which stream a frame belongs to. It comprises a set of rules that map identifying header information of frames such as the source and/or destination addresses to a stream handle. If no rule applies, an empty stream handle is added to the frame

Strict Priority Queuing The algorithm used for transmission selection. It selects the next frame to be transmitted from the highest-priority TSN queue that is eligible for transmission

super-group eligibility time The latest group eligibility time among all group eligibility times for a TSN node. This is used by the SETT

Super-Group Eligibility Time Tagging A specific NETT strategy. When a Non-ATS frame arrives at a TSN node, it is tagged with an eligibilityTimeTag containing the super-group eligibility time

T

Tail-Element Eligibility Time Tagging A specific NETT strategy. When a Non-ATS frame is queued in a shared TSN queue, it is tagged with the same eligibilityTimeTag as the current tail element of the shared TSN queue. If the shared TSN queue is empty, the eligibilityTimeTag is set to the time of arrival of the frame

Time-Sensitive Networking A collection of standards defined by the IEEE 802.1 Task Group for ensuring deterministic communication over Ethernet networks

traffic A summarising term referring to all frames and all streams present in a network

transmission The process of sending a frame from one node to another over a link. It includes all physical and logical operations required to move data across the link

Transmission Selection Algorithm Each TSN queue may use a specific transmission selection algorithm. These include CBSa and ATS. Frames are only selected from TSN queues where the associated TSA determines, that there is a frame eligible for transmission

transmission time The time required to complete the transmission of a frame over a link. It depends on the link capacity and the size of the frame

TSN node A network node that implements the TSN standard IEEE 802.1Q

TSN queue Refers to one of the eight outbound queues present at each egress port of a TSN node. There is one queue per priority, separating frames of different priorities from one another. Each TSN queue is configured to use a specific Transmission Selection Algorithm that determines the eligibility of that TSN queue

U

upstream The direction of a stream from a node towards its source. It may refer to all intermediate nodes a frame has passed through already

Bibliography

[1] *IEEE Standard for Ethernet*, IEEE Std., Rev. 2022, 2022.

[2] *IEEE Standard for Local and Metropolitan Area Networks – Bridges and Bridged Networks*, IEEE Std., Rev. 2022, Dec. 2022.

[3] R. Oshana, "Chapter 1 - Software Engineering of Embedded and Real-Time Systems," in *Software Engineering for Embedded Systems*, R. Oshana and M. Kraeling, Eds. Oxford: Newnes, 2013, pp. 1–32.

[4] D. Abbott, "Linux for Embedded and Real-time Applications (Third Edition)," ser. Embedded Technology, D. Abbott, Ed. Oxford: Newnes, 2013, pp. 3–11. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978012415996900001 0

[5] G. Bernat, A. Burns, and A. Liamosi, "Weakly hard real-time systems," *IEEE Transactions on Computers*, vol. 50, no. 4, pp. 308–321, 2001.

[6] T. Steinbach, "Ethernet-basierte Fahrzeugnetzwerkarchitekturen für zukünftige Echtzeitsysteme im Automobil," Ph.D. dissertation, 2018.

[7] Semiconductor Engineering, "Automotive Ethernet, Time Sensitive Networking (TSN)," 2021, accessed: 2025-04-22. [Online]. Available: https://semiengineering.com/knowledge_centers/automotive/automotive-networking/automotive-ethernet-time-sensitive-networking-tsn/

[8] A. Wallibai and R. G. Deshpande, "Automotive Ethernet: High-speed in-vehicle networking for next-generation electronics," *World Journal of Advanced Research and Reviews*, vol. 8, no. 2, pp. 353–368, 2020, accessed: 2025-04-22. [Online]. Available: https://wjarr.com/content/automotive-ethernet-high-speed-vehicle-networking-next-generation-electronics

[9] "Time-Sensitive Networking (TSN) Task Group," accessed: 2025-01-29. [Online]. Available: https://1.ieee802.org/tsn/

[10] J. Walrand, "A Concise Tutorial on Traffic Shaping and Scheduling in Time-Sensitive Networks," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 3, pp. 1941–1953, 2023.

[11] *IEEE Standard for Local and Metropolitan Area Networks – Bridges and Bridged Networks - Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams*, IEEE Std., Rev. 2009, Jan. 2010.

[12] D. Medhi and K. Ramasamy, "Chapter 3 - Routing Protocols: Framework and Principles," in *Network Routing (Second Edition)*, second edition ed., ser. The Morgan Kaufmann Series in Networking, D. Medhi and K. Ramasamy, Eds. Boston: Morgan Kaufmann, 2018, pp. 64–113.

[13] J. Specht and S. Samii, "Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks," in *2016 28th Euromicro Conference on Real-Time Systems (ECRTS)*, Jul. 2016, pp. 75–85.

[14] *IEEE Standard for Local and Metropolitan Area Networks – Bridges and Bridged Networks - Amendment 34: Asynchronous Traffic Shaping*, IEEE Std., Rev. 2020, Nov. 2020.

[15] L. Thomas and J.-Y. Le Boudec, "Network-calculus service curves of the interleaved regulator," *Performance Evaluation*, vol. 166, p. 102443, 2024.

[16] E. Mohammadpour, E. Stai, M. Mohiuddin, and J.-Y. Le Boudec, "Latency and Backlog Bounds in Time-Sensitive Networking with Credit Based Shapers and Asynchronous Traffic Shaping," in *2018 30th International Teletraffic Congress (ITC 30)*, vol. 02, Sep. 2018, pp. 1–6.

[17] *IEEE Standard for Local and Metropolitan Area Networks – Bridges and Bridged Networks - Amendment 28: Per-Stream Filtering and Policing*, IEEE Std., Rev. 2017, Sep. 2017.

[18] "OMNeT++ Discrete Event Simulator," accessed: 2025-03-31. [Online]. Available: https://omnetpp.org/

[19] "INET Framework," accessed: 2025-03-31. [Online]. Available: https://omnetpp.org/download-items/INET.html

[20] "CoRE4INET Framework," accessed: 2025-03-31. [Online]. Available: https://omnetpp.org/download-items/Core4INET.html

[21] "CoRE Research Group," accessed: 2025-03-31. [Online]. Available: https://core-researchgroup.de/

[22] "Commit 80c9a23 Asynchronous Traffic Shaper with Scheduler Groups," accessed: 2025-03-31. [Online]. Available: https://github.com/inet-framework/inet/commit/80c9a230ca8035ab9c554cb9575f4e7d40445b6a

[23] "OMNeT Manual," accessed: 2025-03-31. [Online]. Available: https://doc.omnetpp.org/omnetpp/manual/

[24] "Python programming language," accessed: 2025-03-31. [Online]. Available: https://www.python.org/

[25] J.-Y. Le Boudec, "A Theory of Traffic Regulators for Deterministic Networks with Application to Interleaved Regulators," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2721–2733, Dec. 2018.

[26] T. Lübeck, P. Meyer, T. Häckel, F. Korf, and T. C. Schmidt, "Asynchronous traffic shaping and redundancy: Avoiding unbounded latencies in in-car networks," *arXiv preprint arXiv:2504.01946*, Apr. 2025.

[27] "inet-fork used for Simulations," accessed: 2025-03-31. [Online]. Available: https://github.com/CoRE-RG/inet/tree/core/nids

A Experimental Setup

Software

The following software was used for conducting simulations and analysing results:

Table A.1: Software used for conducting simulations and analysing results

Software	Version	Usage
OMNeT++	v6.0.3	Discrete event simulation environment for network behaviour modeling [18]
	v4.5.4	Protocol stack and traffic generator support for OMNeT++ [19]
INET Framework	CoRE-RG/inet nids	Protocol stack and traffic generator support including TSN-specific extensions for simulating real-time Ethernet in OMNeT++ [27]
Python	v3.13	Post-processing of simulation output, data analysis, and automation scripts [24]
Jupyter Notebook	v8.6.3	Interactive data analysis and visualisation
MacOS (intel)	Sequoia 15.4.1	OS of computer used for running simulations

Simulation Configuration Parameters

(see Table A.2 on the next page)

Note 1: The seed of each simulation is set to i, where i refers to the current iteration number, starting from 0.
Note 2: The table shows UDP-packet payload sizes, not frame sizes. The total frame size includes the Ethernet physical header and Fcs (12 B), the Ethernet MAC header including VLAN tag (18 B), the IPv4 header (20 B), the UDP header (8 B), and, for the purposes of this thesis, the IFG (12 B) (total overhead: 58 B+12 B=70 B).
Note 3: CIR and CBS are calculated for total frame sizes including IFG.

Table A.2: Simulation Configuration Parameters

Configuration		CFG-2	CFG-9	CFG-10	CFG-10F	CFG-12
Sim-Time Limit		1 s	1 s	1 s	1 s	1 s
Repetitions		1	3	1	1	1
Iterations (i)		6	1	2	2	4
Payload Size	Dependent S.	1180 B	555 B	1180 B	1180 B	1180 B
	Evaluation S.	1180 B	555 B	55 B	55 B	1180 B
	Independent S.	1180 B	555 B	1180 B	1180 B	1180 B
Start Time	Dependent S.	$[0; 0; 1; 2; 1; 2]$ ps	uniform$(0, 600)$ µs	200 µs	200 µs	$200 \cdot 2^{\mathrm{i}} + 2 \cdot 10^{-6}$ µs
	Evaluation S.	$[1; 2; 0; 0; 2; 1]$ ps	uniform$(0, 600)$ µs	100 µs	100 µs	100.000 001 µs
	Independent S.	$[2; 1; 2; 1; 0; 0]$ ps	uniform$(0, 600)$ µs	0 s	0 s	0 s
Send Interval	Dependent S.	400 µs	10 µs	400 µs	400 µs	1 ps
	Evaluation S.	400 µs	10 µs	$[13.334; 20]$ µs	$[13.158; 19.608]$ µs	1 ps
	Independent S.	400 µs	10 µs	400 µs	400 µs	1 ps
Send Interval Jitter	Dependent S.	±1 ps	-	-	-	-
	Evaluation S.	±1 ps	-	-	-	-
	Independent S.	±1 ps	-	-	-	-
Burst Duration	Dependent S.	0 s	uniform$(45, 55)$ µs	0 s	0 s	0 s
	Evaluation S.	0 s	uniform$(45, 55)$ µs	0 s	0 s	0 s
	Independent S.	0 s	uniform$(45, 55)$ µs	0 s	0 s	0 s
Sleep Duration	Dependent S.	0 s	1200 µs	0 s	0 s	0 s
	Evaluation S.	0 s	1200 µs	0 s	0 s	0 s
	Independent S.	0 s	1200 µs	0 s	0 s	0 s
Max. Packets per Burst	Dependent S.	-	6	-	-	1
	Evaluation S.	-	6	-	-	$2 \cdot 2^{\mathrm{i}} - 1$
	Independent S.	-	6	-	-	2
Committed Information Rate	Dependent S.	25 Mbit/s	25 Mbit/s	25 Mbit/s	25 Mbit/s	25 Mbit/s
	Evaluation S.	25 Mbit/s	25 Mbit/s	$[75; 50]$ Mbit/s	$[75; 50]$ Mbit/s	50 Mbit/s
	Independent S.	25 Mbit/s	25 Mbit/s	25 Mbit/s	25 Mbit/s	$50/2^{\mathrm{i}}$ if i > 0 else 50 Mbit/s
Committed Burst Size	Dependent S.	1250 B	1250 B	1250 B	1250 B	1250 B
	Evaluation S.	1250 B	1250 B	125 B	125 B	1250 B
	Independent S.	1250 B	1250 B	1250 B	1250 B	1250 B
Maximum Residence Time	Dependent S.	1 ms	1 s	1 ms	1 ms	1 ms
	Evaluation S.	1 ms	1 s	100 µs	1 ms	10 ms
	Independent S.	1 ms	1 s	1 ms	1 ms	1.5 ms

B Auxiliary Calculations

CFG-9 Maximum Anticipated Scheduler Delay

Using the parameters specified in Table A.2 for CFG-9, the maximum anticipated scheduler delay corresponds to the delay incurred by the last frame of a burst, assuming the ATS scheduler was initially idle upon scheduling the burst's first frame.

In CFG-9, all streams are set-up with the same parameters. They are as follows:

Frame size:	625 B (including IFG)
Burst Duration:	50 µs ± 5 µs
Send Interval:	10 µs
MaxPacketsPerBurst:	6
CIR:	25 Mbit/s
CBS:	1250 B

The maximum number of frames transmitted during each burst is given by

$$min\left(\left(\frac{\text{Burst Duration}}{\text{Send Interval}}+1\right),\ \text{MaxPacketsPerBurst}\right)$$

The +1 stems from the fact that the first frame is sent at the start of the burst interval, as shown in Figure 5.5.

Therefore, the maximum number of frames sent during a burst of CFG-9 is:

$$min\left(\left(\frac{50\,\mu s+5\,\mu s}{10\,\mu s}+1\right),6\right)=min\left((5+1),6\right)=6\,\text{frames}$$

Per burst, 6 frames are sent at most. The transmission time of one frame is:

$$t_{\text{tx}} = \frac{625\,\text{B}}{100\,\text{Mbit/s}} = 50\,\mu\text{s}$$

Since the transmission time of 50 µs is longer than the send interval of 10 µs, the frames of a burst are transmitted to the switch back-to-back without a gap.

The CIR is set to 25 Mbit/s. According to Equation 5.1, this equates to gaps of:

$$T_{\text{send}} = \frac{50\,\mu\text{s} \cdot 100\,\text{Mbit/s}}{25\,\text{Mbit/s}} = 200\,\mu\text{s}$$

The CBS is set to twice the frame size, meaning the ATS scheduler does not insert this gap of 200 µs between the first two transmissions. If we assume that all frames are scheduled at the same time, the last frame would experience a scheduler delay of $200\,\mu\text{s} \cdot (6-2) = 800\,\mu\text{s}$.

From this, we have to subtract the time the ATS scheduler had to recover credits. The first frame arrives at the switch at T_0. The last frame arrives at $T_0 + 50\,\mu\text{s} \cdot 5 = T_0 + 250\,\mu\text{s}$. This means that when the last frame is scheduled, the ATS scheduler has effectively had 250 µs to recover. Subtracting this from the calculated scheduler delay of 800 µs gives us the final maximum delay:

$$800\,\mu\text{s} - 250\,\mu\text{s} = \underline{\underline{550\,\mu\text{s}}}$$

C Definitional Impurities in Stream Filters & ATS Scheduler Groups

During the research conducted for this thesis, an additional issue emerged. Section 8.6.5.3 of the IEEE 802.1Q standard specifies that each Stream Filter can allocate an ATS scheduler instance based on a combination of the stream handle and priority. Importantly, it allows the use of wildcards in place of specific priorities, which means that frames of differing priorities may be handled by the same ATS scheduler instance. This behaviour is explicitly illustrated in Figure 8-15 of the standard.

Moreover, Section 8.6.5.6 states that each ATS scheduler is assigned a fixed ATS scheduler group identifier. It further states that there should be one ATS scheduler group per upstream traffic class and ingress port.

Since the standard does not mandate that a single instance processes frames of only one priority, frames of different priorities can be processed by an ATS scheduler linked to any ATS scheduler group, regardless of the frame's actual priority.

Some implementations, such as [22], diverge from this approach by dynamically selecting the ATS scheduler group based on each frame's header information. While this ensures that frames are assigned to the correct ATS scheduler group, it still permits a single ATS scheduler instance to process frames of multiple distinct priorities.

As discussed in Section 6.4, ATS scheduler group can cause undesirable scheduling effects [25, 26]. Since these issues stem from the group eligibility time, the definitional ambiguity described above could potentially be leveraged to alleviate such problems. However, alternative mitigation strategies have also been proposed and explored in related work [26].

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original