[D2.2]

Implementation of selected release 1 PREDICT-6G MDP innovations.

PREDICT-6G

Funded by
the European Union

Revision v0.1

Work package	WP2
Task	T2.1, T2.2 and T2.3
Due date	31-10-2023
Submission date	30-11-2023
Deliverable lead	UC3M
Version	V2.2
Authors	Pablo Picazo, David Rico (UC3M) Sebastian Robitzsch, Chathura Sarathchandra, Renan Krishna (IDE) Tamás Kárász, Szabolcs Nováczki, Péter Szilágyi, Zoltán Vincze, Csaba Vulkán (NOK) Carla Fabiana Chiasserini (POLITO) Margarita Cabrera, Olga Muñoz, Josep Vidal, Javier Villares (UPC) Angelo Cenedese, Manuel Cheminod, Stefano Vitturi, Claudio Zunino (CNR and UNIPD) Rafael Rosales, Dave Cavalcanti (INT) Luis M. Contreras (TID) Marc Mollà Roselló (ERC)
Reviewers	Antonio de la Oliva (UC3M)

Abstract

This document includes the first implementations of the MDP innovations selected from D2.1. In this document, we provide the code to provide determinism in the data plane for different domains.

Keywords

Multi-domain, TSN, MDP, Data-plane, DetNet, 802.1Qbv, 3GPP

Document revision history

Version	Date	Description of change	Contributor(s)
V1	15-09-2023	TOC proposal	Pablo Picazo (UC3M)
V2	25-10-2023	First draft	Pablo Picazo (UC3M)
V2.1	10-11-2023	Draft for internal review	David Rico (UC3M)
V2.2	14-11-2023	Full Draft internal review	Pablo Picazo (UC3M)
V3	22-11-2023	Revision of Antonio's comments	David Rico (UC3M)
V4	06-06-2024	Revision of Reviewers comments	David Rico (UC3M)

Document information

Nature of the deliverable	R	
Dissemination level		
PU	Public, fully open. e.g., website	✔
CL	Classified information as referred to in Commission Decision 2001/844/EC	

Table of contents

List of figures

List of tables

Acronyms and definitions

MDP	Multi-Domain Process
TSN	Time-Sensitive Networking
DetNet	Deterministic Networking
UL	Uplink
DL	Downlink
gNB	Next Generation NodeB
BE	Best Effort
RRM	Radio Resource Management
AV/VR	Augmented/Virtual Reality
TS	Time Sensitive
TDD	Time Division Duplexing
QoS	Quality of Service
CSI	Channel State Information
QSI	Queue State Information
PI	Packets Information
RL	Reinforcement Learning
RB	Resource Block
SNR	Signal-to-Noise Ratio
URLLC	Ultra-Reliable Low-Latency Communications
RRC	Radio Resource Control

DAPS	Dual Active Protocol Stack
PDCP	Packet Data Convergence Protocol
SN	Sequence Number
UDM	User Data Management
NWDAF	Network Data Analytics Function
ADRF	Analytics Data Repository Function
NEF	Network Exposure Function
MQTT	Message Queuing Telemetry Transport
PFP	Packet Forwarding Protocol
MA	Measurement Agents
DPDK	Data Plane Development Kit
DS-TT	Device Side Time-Sensitive Networking Translator
NW-TT	Network Side Time-Sensitive Networking Translator
eBPF	extended Berkeley Packet Filter
XDP	Express Data Path
PREOF	Packet Replication and Elimination for Reliability
PHP	Packet Handling Protocol
PRF	Packet Replication Function
PEF	Packet Error Function
FWD	Forwarding

Table of partners

Short Name	Partner
UC3M	Universidad Carlos III de Madrid
NOK	Nokia Solutions and Networks KFT
ERC	Ericsson Espana SA
INT	Intel Deutschland GMBH
TID	Telefonica Investigacion y Desarrollo SA
ATOS	ATOS IT Solutions and Services Iberia SL
GES	Gestamp Servicios SA
NXW	Nextworks
COG	Cognitive Innovations Private Company
SIM	Software Imagination & Vision SRL
AUSTRALO	AUSTRALO Alpha Lab MTU
POLITO	Politecnico di Torino
UPC	Universitat Politecnica de Catalunya
CNR	Consiglio Nazionale delle Ricerche
UNIPD	Universita degli Studi di Padova
IDE	Interdigital Europe Ltd

1 Executive summary

This document presents the initial implementations of the Multi-Domain Process (MDP) innovations selected from Deliverable D2.1. The focus of these implementations is to introduce determinism in the data plane across various domains, contributing to the PREDICT-6G project.

The deliverable, led by Universidad Carlos III de Madrid (UC3M), encompasses advancements in dynamic scheduling for mixed traffic, development of extensions for the operation of 5G systems as Time-Sensitive Networking (TSN) bridges, and the implementation of IEEE 802.1Qbv for enhanced traffic management. It also addresses the challenges of flow monitoring in non-deterministic domains and the integration of Deterministic Networking (DetNet) with end-to-end data plane services. This deliverable is complemented with multiple implementations of the different innovations described herein, which are available (subject to constrains from the different partners) in different code repositories indicated in the respective section.

Key innovations include:

- **Dynamic Schedulers for Mixed Traffic**: Addressing the allocation of uplink (UL) and downlink (DL) resources in 5G networks, with a focus on optimizing the scheduling of time-sensitive and best-effort traffic flows.
- **Time-Based Schedulers on 3GPP**: Extending the capabilities of 5G systems to support TSN operations, ensuring time synchronization and reduced latency for critical applications.
- **TSN Redundancy**: Enhancing the resilience of networks to interference and attacks, ensuring reliable communication for time-sensitive applications.
- **IEEE 802.1Qbv Implementations**: Implementing traffic management mechanisms to support real-time and best-effort traffic, with a focus on offset synchronization and latency evaluation.
- **Flow Monitoring**: Developing methods for monitoring traffic flows in domains where determinism cannot be guaranteed, ensuring quality of service for diverse applications.
- **DetNet Implementation**: Integrating DetNet capabilities to provide deterministic services across the data plane, from Layer 2 to Layer 3, and ensuring end-to-end performance.

Keywords: Multi-domain, TSN, MDP, Data-plane, DetNet, 802.1Qbv, 3GPP

2 Introduction

This document stands as a pivotal guide in the journey of PREDICT-6G, specifically focusing on the initial implementation phase of Multi-Domain Process (MDP) innovations. Building upon the foundational work established in Deliverable D2.1, this report delves into the critical integration of determinism within the data plane, spanning a multitude of network domains. This endeavour is key to enhancing the efficiency and reliability of network operations in the context of 6G technologies.

The structure of the document is designed to facilitate a clear understanding and practical insight into the implementation of these innovations. It is divided into three main sections, each addressing a specific facet of the MDP innovations and their applications.

Section 3 is dedicated to exploring the fundamental research implementations, with a focus on dynamic scheduling. It examines various standardized tools and methods for extending protocol headers. These innovations are pivotal in driving a more deterministic and versatile behaviour of network nodes, thereby significantly enhancing the Quality of Service (QoS) within the network. The content here bridges the gap between theoretical research and practical applications, offering a thorough examination of the latest advancements in network technology.

In section 4, the document transitions to presenting standalone demonstrations, showcasing the practical application of the theories and strategies discussed earlier. The first demonstration covers localization and sensing across Time-Sensitive Networking (TSN) domains, while the second focuses on Wi-Fi TSN. This includes utilizing techniques such as path diversity, as introduced in IEEE 802.1CB, to bolster network resilience against interference or malicious attacks. Additionally, this section explores the enhancement of end-user mobility, aiming for a seamless roaming experience.

Section 5 presents integrated demonstrations. It considers the implementation of flow monitoring in network segments where the underlying technology does not natively provide such measurements. This includes an in-depth look at industrial cross-domain TSN use cases, featuring a DetNet implementation, an XDP switch implementation, and a preliminary insight into the deterministic end-to-end multi-domain use case. This section is particularly critical as it encapsulates different time-sensitive domains and their interoperability, highlighting the strides made towards a more cohesive and efficient network system.

3 Fundamental Research

3.1 Dynamic Scheduling

3.1.1 Dynamic schedulers for mixed traffic

In the wireless segment, multiple UE establish connections with a gNB to exchange diverse flows with the network, including time sensitive (with latency and possibly jitter constraints - TS) and best-effort (with no minimum rate or latency constraints - BE) flows. TS flows include isochronous, periodic traffic from control devices (e.g., sensors and actuators) and aperiodic bursty traffic from other time-sensitive applications such as AR/VR. BE flows carry aperiodic traffic that has less stringent or sometimes no specific requirements in latency, jitter, and throughput.

The **Radio Resource Management (RRM)** module is responsible for the dynamic allocation of OFDMA radio resources (e.g., power and resource blocks) and packet scheduling for flows sharing limited physical resources, as well as doing admission control mechanisms (aspects that are assessed at the system simulator level). All are essential to ensure QoS of flows in specific wireless channel conditions that may change over time. It also must manage situations where QoS drops to unacceptable levels in scenarios where congestion occurs. In such cases, pre-emption can be employed by acting on BE flows in the first place (either by dropping flows or degrading their QoS), and then less critical TS flows. All flows are identified at the scheduler level by its QoS class.

A hierarchical/modular approach is designed that involves two levels:

- an upper-level scheduler that dynamically decides the fraction of allocated time slots for UL and DL transmissions as well as the TDD transmission pattern,
- the UL and DL RRM modules operate independently with the goal of accommodating as many flows as possible with the required QoS.

To make informed decisions, the following information regarding UL and DL should be collected:

1. Channel State Information (CSI): The measurement of the status of the channel should include attenuation and interference levels and should be monitored frequently.
2. Queue State Information (QSI): The status of queues for each QoS class should be monitored, including the number of packets waiting in the queues and their associated priorities.
3. Packets Information (PI): To prioritize packets based on their latency requirement, the timestamps of packets in the queues provide the required information about the packets' arrival rate. This timestamp is attached to the MAC layer PDU by the RLC.

3.1.1.1 Duplexing and access

In 5G we adopt time division duplexing (TDD) communications within each gNB, with the UL and DL intervals dynamically designed on a per-superframe basis. Within this allocation, a precise UL/DL TDD pattern is chosen using a reinforcement learning (RL) approach that takes decisions maximizing the reward of having an equal performance measure in UL and DL.

The scheduler dynamically allocates available radio resources to active flows. We will consider that the minimum indivisible unit to be allocated is a resource block (RB). An RB spans one slot in time and one sub-band in frequency with bandwidth B/J, with B being the total available bandwidth and J the number of sub-bands. The values for the slot duration, T_{slot}, and, if OFDMA is supported, the number of sub-bands, J, are selected as follows:

- The slot duration T_{slot} is assumed to be a multiple of the OFDM symbol duration.
- The selection of the number of sub bands J is a trade-off between performance and complexity. Depending on the coherence bandwidth of the propagation channel, it is important to note that, in terms of spectral efficiency, the performance improvement achieved by adopting higher values of J is expected to be marginal.

In summary, in each superframe, the scheduler has N_{slots} x J resource blocks available. As a baseline, we will assume that each RBs is assigned entirely to a single flow (orthogonal multiple access). If the gNB were equipped with multiple antennas, it would be possible to transmit multiple flows simultaneously within the same RB, thus increasing the system's capacity, see Figure 1.

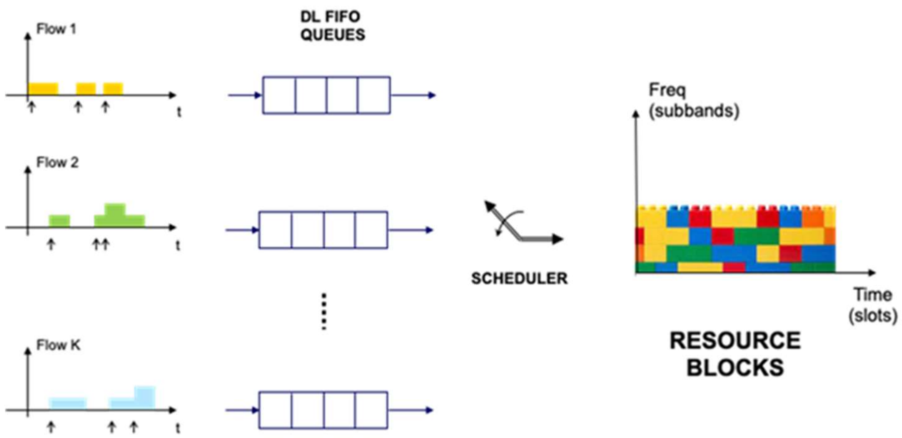


Figure 1: Asynchronous traffic queues and OFDMA resource allocation

The decision on RB and power allocation is taken by the scheduler at the beginning of each superframe. In a wireless channel, the transmission rate of available RBs is not constant, but it varies along the following dimensions:

1. **Time variability/diversity.** In an industrial indoor environment, the propagation channel is assumed to change slowly over time. The coherence time of the channel is around 10-30 ms [1]. Considering a fixed superframe duration of 1 ms implies that the channel remains approximately constant for 10-30 superframes. Consequently, the scheduler cannot exploit time diversity.
2. **Frequency variability/diversity.** In the industrial indoor environment, the channel coherence bandwidth is approximately 10-30 MHz [1]. To efficiently exploit spectral diversity, sub-bands should not exceed the coherence bandwidth of the wireless channel. For large bandwidths (B>40 MHz), the scheduler can leverage frequency diversity by utilizing multiple sub bands (J>1). Therefore, as a reference, the number of sub bands, J, can range from 2 to 32, depending on the total bandwidth, 20 MHz < B < 320 MHz Higher values of J can be chosen to increase the granularity of the scheduler and hence meet tight latency constraints.
3. **Spatial variability/diversity.** If the flows are intended for different user terminals, their propagation channels can be considered statistically independent due to their distinct locations and varying electromagnetic environments.

3.1.1.2 Allocation of UL and DL resources

Inspired by [2], to allow the system to adapt to changing traffic conditions by learning from past experiences and interactions with the environment, a Reinforcement Learning (RL) approach is adopted for dynamically adjusting the number of slots allocated for UL and the number of slots allocated DL in each superframe. In the RL framework, the state is defined as a function of observations such as the traffic load, network congestion, and experienced QoS, all of them related to the following measurable parameters:

- The aggregated length of the DL queues (in bits) at the end of superframe n, $q_{DL}(n)$
- The aggregated length of the UL queues (in bits) at the end of superframe n, $q_{UL}(n)$
- The average delay experienced by the DL packets served in superframe number n, $\underline{d}_{DL}(n)$
- The average delay experienced by the UL packets served in superframe number n, $\underline{d}_{UL}(n)$

In turn, the reward is a function that is defined in the statement of the RL problem, to align the main scheduling goal with the function that is maximized in the RL algorithm, i.e., the accumulated reward (Figure 2).

Figure 3 shows at the top the input TSN traffic throughout 100,000 5G frames (1000 sec.), measured in packets, and at the bottom the Q function learned.

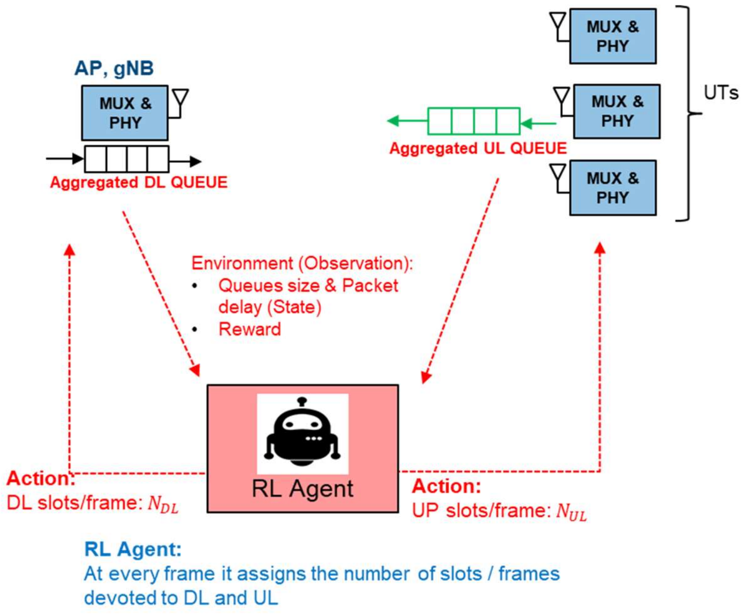


Figure 2: RL based adaptive TDD frame design.

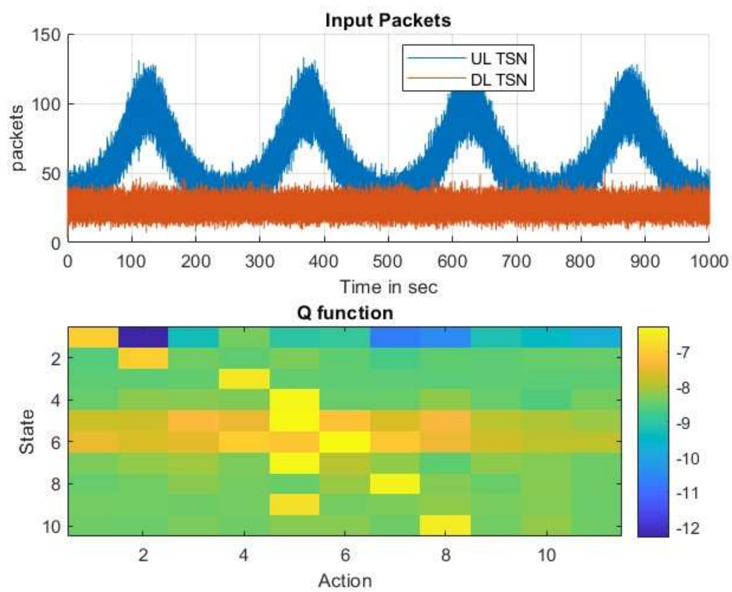


Figure 3: Top: Input TSN UL&DL traffic measured in packets. Bottom: The Q-function displays the best action the RL agent must take depending on the state of the traffic.

The state space considers the percentage of UL to DL buffered traffic. The action space is related to the number of UL/DL slots in the radio frame. Rewards account for a balance between UL/DL traffic: perfectly balanced buffered traffic generates higher return. In the plot, the system exhibits state 1, which must be connected to action 1, as the estimated Q-function correctly identifies in the lightest block of the first row. The rest of the states correspond to situations in which the size of the DL queues increases with respect to that of the UL, and the actions consequently shift towards a greater number of slots assigned to the DL.

3.1.1.3 DL/UL Radio Resource Allocation

Once the split of DL/UL resources has been decided by the TDD pattern selector, the DL scheduler distributes the radio resources among the active Isochronous TS traffic, Asynchronous TS traffic, and best-effort traffic flows. In the DL, on each scheduling period, the downlink transmitted power is minimized with the following constraints:

- Each user can be allocated any number of resource blocks (RB)
- Each resource block is allocated to a maximum of one user.
- The transmission rate allocated to each user follows the Shannon transmission rate formula (affected by an SNR penalty) or the Polyanskiy's finite block length regime transmission rate with the proper approximation to make it a convex function.

The solution to the convex problem is semi analytic and includes a low complexity sub gradient step on the multipliers associated to the transmission rate. For the UL, the objective function is the minimization of the maximum transmitted power among the terminals.

The approach followed can encompass different traffic types just by selecting the target transmission rate for every flow. For the types discussed we have:

- Best effort traffic. Several flows can be accommodated in a slice of resource blocks which is characterized by either a total sum rate or a maximum number of used RB. Flows are admittedas long as does not exceed a certain prefixed lumped rate or a total power limitation. As this traffic has no restrictions in QoS, it can be accommodated in non-committed resources. Note that, under the approach followed, the minimization of the total used power entails a complete use of all RB so the only non-committed resource is the difference between the power obtained from the optimization and the maximum transmitted power available at the transmitter. Therefore, we propose to include BE flows with a tentative rate constraint and to reduce (or increase) this rate if the required power exceeds (or is smaller than) the available power.
- Isochronous traffic. Short packets are generated periodically with strict delay constraints, so they must be served as soon as they arrive at the queues, with a target maximum probability of error.

Knowing the packet size L and the scheduling period T_s the target transmission rate is $r_o = L/T_s$. A specific QoS identifier is used for this traffic, defining loss rate, delay budget and maximum jitter.

- Asynchronous URLLC traffic. It is characterized by short packet lengths, a target outage delay $\Pr\{D > D^{max}\} < 1 - g$ and possibly jitter (the variance of D) constraints. For M/M/1 models, the relation of $\Pr\{D > D^{max}\}$ to the required r_o and to the packet arrival rate a for exponential traffic model is

$$r_o = a - \ln(1 - g) / D^{max}$$

If the flow has jitter constraints $\mathrm{var}\{D\} < J$ we can use the variance expression for the exponential distribution: $\mathrm{var}\{D\} = 1/(r_o - a)^2$. Therefore, a joint delay and jitter bound for a flow generates the constraint:

$$r_o = a + \max(J^{-1/2}, - \ln(1 - g) / D^{max})$$

For model-free traffic, Markov's and Little's theorem provides a looser target for the outage delay constraint related to the length of queues over time. A bound for instantaneous jitter $\Pr\{(D > D^{max})^2 > J\}$ can also be defined based on Chebyschev's inequality.

3.1.1.4 Admission control

In case the required transmitted power obtained from the solution of the optimization problem for all flows may surpass the available power at the transmitter, as a result of additional incoming non-BE flows, or a degradation in the propagation channel conditions. If this is the case, pre-emption can act by reducing the transmission rate of BE-flows (or dropping flows if needed) in the first place, and then less critical TS flows. This can be efficiently monitored through the Lagrange multipliers associated to every user computed in the sub gradient step of the RRM optimization algorithm described in section 2.1.1.3.

3.1.1.5 Dual Active Protocol Stack Handover

In 3GPP TS 38.300 [14], Dual Active Protocol Stack (DAPS) Handover is defined as a handover procedure that maintains the source gNB connection after reception of RRC message (HO Command) for handover, and until releasing the source cell after successful random access to the target gNB. Until 5G NR release 15, the UE typically releases the connection from the source cell before the connection is established with the target cell (Hard Handover). Due to this, UL and DL transmission is finalized at source cell before the UE starts to communicate with the target cell results an interruption of a few tens of milliseconds in the communication between the UE and the base station. This interruption is very critical for the Predict6G envisioned applications.

3GPP has proposed a solution to overcome this problem named as Dual Active Protocol Stack (DAPS) where UE connection with source cell remains active for Rx and Tx of user data, until it is able to send and receive user data in the target cell. This puts a new requirement at UE side to simultaneously receive and transmit data at both source and target gNB for a short time period during the handover procedure.

The dual active protocol stack (DAPS) handover characteristics follow:

- UE maintains Tx/Rx with the source gNB after receiving the handover (HO) request.
- UE performs simultaneous reception of user data from source and target cell in the DL.
- UE maintains packet data convergence protocol (PDCP) packet connection to the source gNB until the handover.
- UE maintains PDCP UL packet connection to the source gNB until completion of the RACH. procedure with the target gNB, then it switches UL transmission to target gNB.
- DAPS handover is possible over either interface Xn or N2.
- A DAPS handover can be used for an RLC-AM or RLC-UM bearer.

This way, the DAPS HO reduces interruption times during handover close to 0 ms, by maintaining the source gNB radio link (including data flow) while establishing the target gNB radio link.

It is assumed that radio resource blocks used by the source and target gNB are orthogonal, so the UE receives user data simultaneously from both the source and target cell, the PDCP layer is reconfigured to a common PDCP entity for the source and target user plane protocol stacks. To secure in-sequence delivery of user data, PDCP Sequence Number (SN) continuation is maintained throughout the handover procedure. For that reason, a common (for source and target) re-ordering and duplication function is provided in the single PDCP entity.

The simultaneous reception of DL traffic requires the joint allocation of radio resources (transmit power and resource blocks) at the source and target gNB, as well as the identification of the feasibility of both simultaneous connections during DAPS handover and the single connection to the target gNB after the handover is completed. This is approached by implementing the minimization of the maximum transmit power between the source and the target gNB with the rate constraints as described in section 2.1.1.3 In case the solution is not feasible, dropping flows using a priority policy as described in 2.1.1.4 is adopted.

Repository available **here**.

3.1.2 Aligning rTWT and IEEE 802.1Qbv

Following previous deliverable [4], our proposal involves an improvement that integrates Time-Sensitive Networking (TSN) by synchronizing channel access through rTWT and aligning it with the 802.1Qbv scheduling mechanism. The researchers in [18] have demonstrated a practical example of TWT used for time-sensitive scheduling, effectively segregating high-priority traffic from regular traffic, yielding encouraging outcomes in terms of reduced latency and jitter. Building on this, we use network calculus analysis, as noted in [17], to validate that TSN's required bounded latencies can be attained through this method, accommodating both high and low priority packets in a Station (STA) environment, with a certain probability of violation contingent on the channel's quality. Our work includes modeling the 802.1Qbv's gate operation mechanism using Network Calculus and the impact of wireless errors through stochastic scaling as per [19].

3.1.2.1 Modelling 802.1Qbv as TDMA curves

The 802.1Qbv's Time-Aware Shaper (TAS) controls the opening and closing of each queue through a Gate Control List (GCL), which sets the time cycle T_q for each queue and the duration L_q it remains open for transmission. By utilizing Time Division Multiple Access (TDMA), a specific user queue q is assigned a transmission service curve, denoted as $\beta_{L,T}(t)$, formulated in following equation.

$$\beta_{L,T}(t) = C \cdot \max\left\{\left\lfloor \frac{t}{T} \right\rfloor L, t - \left\lceil \frac{t}{T} \right\rceil (T - L)\right\}$$

Here, C represents the transmission rate, and T, L denote the periodicity and duration of TDMA transmission slots. In 802.1Qbv, the service curve for a queue q is represented by $\beta_{L_q,T_q}\left[t - \delta_q\right]^+$, where δ_q > 0 marks the initial opening time of the queue q gate.

It's important to note that in 802.11Qbv, it's possible for the gates of multiple queues to be open simultaneously. To manage this concurrency of gates, we apply an affine lower bound to the TDMA service curve.

$$\beta_q(t) = \frac{C \cdot L_q}{T_q + L_q}[t - \delta_q]^+.$$

3.1.2.2 Concurrent 802.1Qbv opened gates.

When high priority queue q and low priority queue q' are opened concurrently, packets of q are transmitted ahead of those from q'.

Given two queues q, q' whose gates are opened concurrently; the high priority queue q has a strict service curve.

$$\beta_q^*(t) = R_q\left[t - \left(\delta_q + \frac{l_{\max}^{q'}}{R_q}\right)\right]^+$$

Where $R_q = \frac{C \cdot L_q}{T_q + L_q}$.

Similarly, it is also possible to obtain the service curve for the traffic in the low priority queue q when such queue is opened concurrently with the higher priority queue q.

Given two queues q, q' whose gates are opened concurrently; if the flow at q has a strict affine arrival curve γ_{C_q, b_q} , then the low priority queue q' has a strict service curve:

$$\beta^*_{q'}(t) = (R_{q'} - C_q)\left[t - \frac{R_{q'}\delta_{q'} + b_q}{R_{q'} - C_q}\right]^+$$

Proofs to these two statements are given in [17].

3.1.2.3 Retransmissions in 802.11

Until now, we've established the service curves $\beta^*_q(t)$ and $\beta^*_{q'}(t)$ for high and low priority traffic relative to the open and closed states of the 802.1Qbv gates. By synchronizing these gate openings with the rTWT sessions, as we can avoid channel contention. This is because the STA is allocated a specific Resource Unit (RU) for 802.11ax Multi-User transmission (MU) Tx, and access to the rTWT sessions is limited.

Nonetheless, transmission errors due to channel issues might still occur, leading to the need for retransmissions. To address this, we adopt the stochastic scaling method proposed in [19]. This technique involves transmitting the outgoing traffic of 802.1Qbv, represented as $D_q(t)$ and $D_{q'}(t)$, over a wireless channel characterized by a scaling process S, as depicted in Figure 4. In cases where the transmission is unsuccessful, the STA will wait a maximum duration of W before attempting to retransmit the i-th packet. We model this waiting period using the scaling curve δ_W.

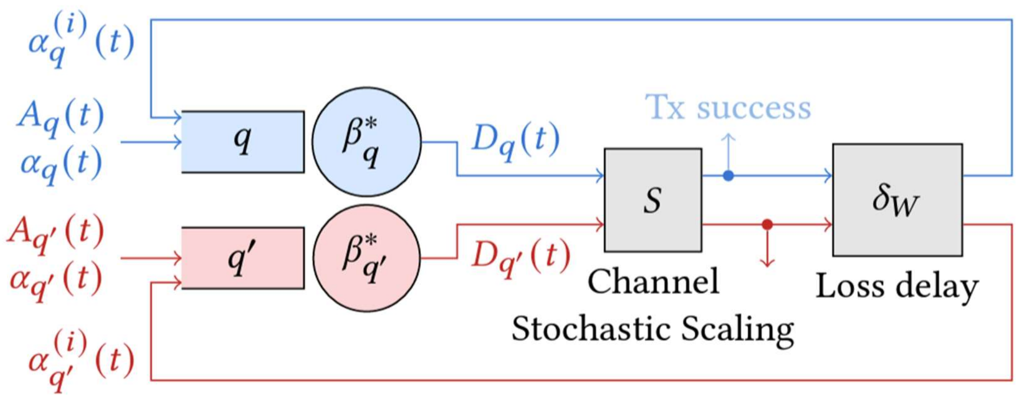


Figure 4: When the q, q' are opened, packets D(t) traverse the stochastic wireless channel S. On Tx error, delta W represents the time it takes to realize the packet loss. Then, packets are fed-back for retransmission leading to re-entrant arrivals.

When the q, q' are opened, packets $D(t)$ traverse the stochastic wireless channel S. On Tx error, δ_W represents the time it takes to realize the packet loss. Then, packets are fed-back for retransmission leading to re-entrant arrivals.

Following [4], we consider a binary symmetric channel modelled with stochastic scaling process $S(b) = \sum_i^b X_i, X_i \sim^{iid}$ Bernoulli(p), with p the loss probability. We now define a stochastic scaling curve $S^\varepsilon(b) = pb + 1 - \varepsilon$ that, according to[19], satisfies:

$$\mathbb{P}\left(\sup_{0 \leq a \leq b} \left\{S(b) - S(a) - S^\varepsilon(b-a)\right\} \leq 0\right) \geq 1 - \varepsilon, \quad \forall b \geq 0$$

Once the system is solved, we define the aggregated arrival curve for the queue q. At this point, given the aggregated arrival curve and the service curve, Network Calculus provides bounds to obtain the maximum delay experienced by queue q using the following expression:

$$h(\alpha, \beta) := \sup_{s \geq 0} \left\{\inf\{u \geq 0 : \alpha(s) \leq \beta(s+u)\}\right\}$$

Note that previous curves are derived from the stochastic scaling process that model the channel, hence, they are not deterministic. In fact, as [19] highlights, the reliability of the delay bound (i.e., the probability that it holds) will be:

$$\mathbb{P}\left(d(t) \leq h(\alpha^{(Tot)}, \beta)\right) \geq (1-\varepsilon)^N \qquad \forall t \in \mathbb{R}^+,$$

3.1.2.4 Results

To highlight the importance of these results, we analyze two scenarios: with (i) T = 3, L = 1 and (ii) T = 3, L = 2. In both, we calculate the aggregated arrival curves for one non-overlapping queue and two overlapping queues (one with higher priority than the other). The rest of the parameters are C = 10, α (0) = γ0.1,0.001 and N = 3. Additionally, the violation probability of the stochastic scaling curve ε is 3.3344e−4.

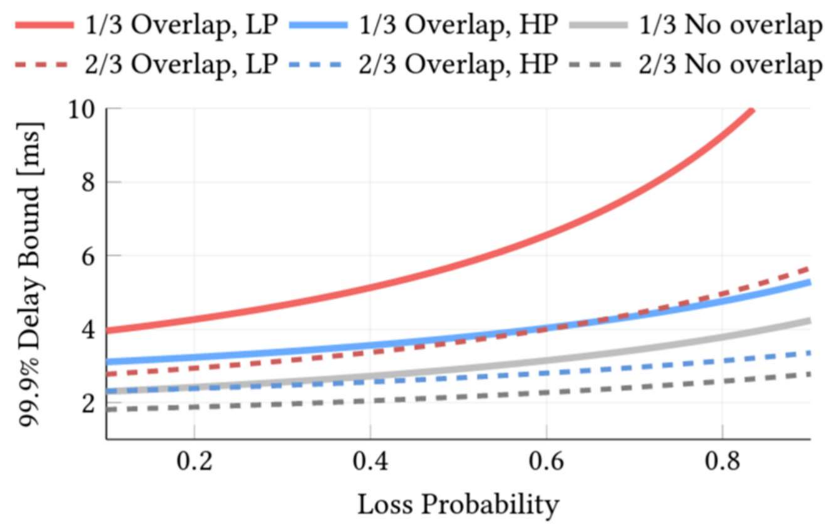


Figure 5: Delay bound vs. wireless loss probability for different L/T ratios. The delay bound holds with a 99.9% prob. We test non-overlapping setups with single 802.1Qbv queues (gray), and overlapping setups with concurrent high (HP) and low-priority (LP) queues (blue and red)

Figure 5 shows the maximum delay experienced by the queues as a function of the loss probability p. Note that these bounds hold with a 0.999 probability. We observe that the delay is reduced with greater overlap. The high priority queue is less impacted compared to the low priority queue. Moreover, comparing the L/T = 1/3 non-overlapping case with 2/3 overlapping case, the high priority queue in the latter case experiences less delay at the expense of the low priority queue.

Repository available **here**.

3.2 Data Unit Groups for Deterministic Communications over TSN and 3GPP Networks

In today's networks QoS treatments and (time-sensitive) packet scheduling is performed primarily at the granularity of the packet with a notion of flows and traffic classes. While there are advantages to this, the lack of knowledge and/or the consideration of information carried in the payload packets, prevents today's networks from optimizing procedures for application layer data/information units (e.g., treating packets that belong to the same video frame collectively, may allow networks to avoid retransmissions due to packet loss, end-to-end).

The ability to extend existing protocol headers allows for a more versatile and programmable behavior of compute nodes, such as switches and routers. Especially, when operating beyond a single computer network (e.g. 3GPP-based mobile network traversing into the Data Network), protocol headers that allow to carry information for one computer network but not useable/applicable in another one may positively impact the versatility of such protocols. For instance, one of the advantages of IPv6 over IPv4 is the requirement of routers to only check mandatory IPv6 header fields. Extension headers must follow a specific sequence in the header but are recommended to be ignored if not needed allowing the router to operate at higher speeds. On the contrary, IPv4 header fields must all be parsed by a router making it slower when adding more information to the header. For security reasons, packets with unknown (but optional) headers can be treated as a thread and rejected though. Therefore, when introducing new header fields, special care must be taken to ensure that the added fields do not interfere with the normal operations of the protocols when traversing two or more computer networks.

In Releases 16 and 17 3GPP specified the support of TSN by defining functionalities and procedures of a TSN AF and the support of Device-Side and Network-Side Translators acting as TSN switches. TSN follows the concept of identifying and classifying continuous streams of incoming packets and applying traffic engineering methods for time-sensitive applications. However, the applicability of the PDU Sets functionality is limited to the 3GPP domain only and does extend to the TSN domain. Thus, this section addresses the question of how to extend the concept of PDU Sets to the TSN domain, enabling the computer networks to offer application-layer protocol independent grouping of packets carrying an application data unit.

3.2.1 3GPP’s Packet Data Unit Set Feature

3GPP Rel.18 introduced support for XR sessions and PDU sets allowing the system to serve different QoS-sensitive service flows to a single or multiple UEs that are collectively participating in a single application. For instance, sensing data formats can range from pure text-based periodic short bursts of numeric values to actual video streams for machine-based processing. All sensing data collectively forms the input to the algorithm that determines the sensing result, while the core network is chartered with processing the sensing data, following 3GPP’s TR 22.837 proposition. Each sensing data type (numeric vs video) will have its own QoS Flow with different QoS requirements (5QIs) and the set of these QoS Flows form the sensing service.

Figure 6 illustrates the Rel.18 features PDU Sets and Multi-Modal Data Flows, as described in 3GPP’s TR23.700-60 and considered for implementation in the Localization and Sensing Proof-of-Concept. PDU Sets can give the 5G network the ability to understand which PDUs belong to an application-related data unit which can only be used for further processing (by the application) if received in full. For instance, if the PDU Set forms a video frame or a sensing data result, the 5GS can ensure the correct delivery of the

PDU Set and apply potential re-delivery of PDUs within a PDU Set within the RAN via HARQ retransmissions. In some cases, if for any reasons the 5GS must drop packets to deliver on the indicated 5QIs, it would be advisable if entire PDU Sets are dropped to free up resources for other PDU Sets to be delivered in their entirety.

Moreover, TR23.700-60 also introduces the concept of Multi-Modal Data Flows, which allows the 5GS to identify inter-related QoS flows distributed across multiple UEs that are serving the same application. For instance, if QoS Flow 1 is radar sensing data and QoS Flow 3 video sensing data, the 5GS can ensure that both flows are treated as a Multi-Modal Data Flow for combined consumption of an ML-based analytics engine that processes the sensing data.

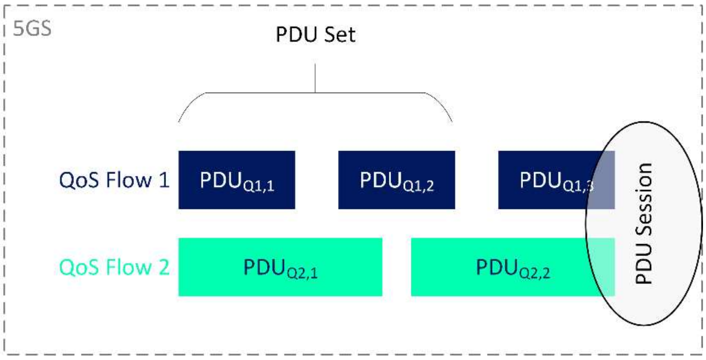


Figure 6: Packet Data Unit Set Concept Overview

The table below provides the PDU Set markings specified by 3GPP to fill out PDU Set Markers, as specified in TS26.522.

Table 1: Packet Data Unit Set Markings

PDU Set Marking	Length	Description
End PDU of the PDU Set	1 bit	If set, it indicates the last PDU of the PDU Set
End of Data Burst	3 bits	While not specified yet in detail, the 3 bits indicate the end of a data burst within a PDU Set
PDU Set Importance	4 bits	This field allows to indicate the importance of this PDU Set compared to other PDU Sets within the same QoS flow. The lower the value the higher the importance. At the moment, the application layer codec level aspects are used to define the importance, i.e. video, audio, text/metadata, image

PDU Set Sequence Number	10 bits	Sequence number that identifies the PDU Set against other PDU Sets
PDU Sequence Number within PDU Set	6 bits	Sequence number of a PDU within a PDU Set
PDU Set Size	24 bites	The total size of all PDU in the PDU Set. Note, this field is an optional field according to 3.2.6

To mark packets as a PDU Set requires the detection of packet streams using header information. 3GPP studied the dominant protocols used to stream media content in TS23.700-60 and covered protocols such as RTP and HTTP. And while RTP targets video delivery only, a more widely applicable option is presented herein, i.e. utilizing the IP header, and proposing a new header extension/option.

3.2.2 Data Unit Groups

Modern applications often send an application data unit which does not fit into a single packet on Layer 2 or 3. As a result, a single Application Data Unit (ADU) is fragmented into a smaller chunks with a maximum length of each chuck determined by the underlying computer network and its Maximum Transmission Unit (MTU), as illustrated in Figure 7 (for example, UHD video frame may be carried by more than one IP packet). The set of packets that carry the entirety of a single ADU are hereby defined as a Data Unit Group (DUG).

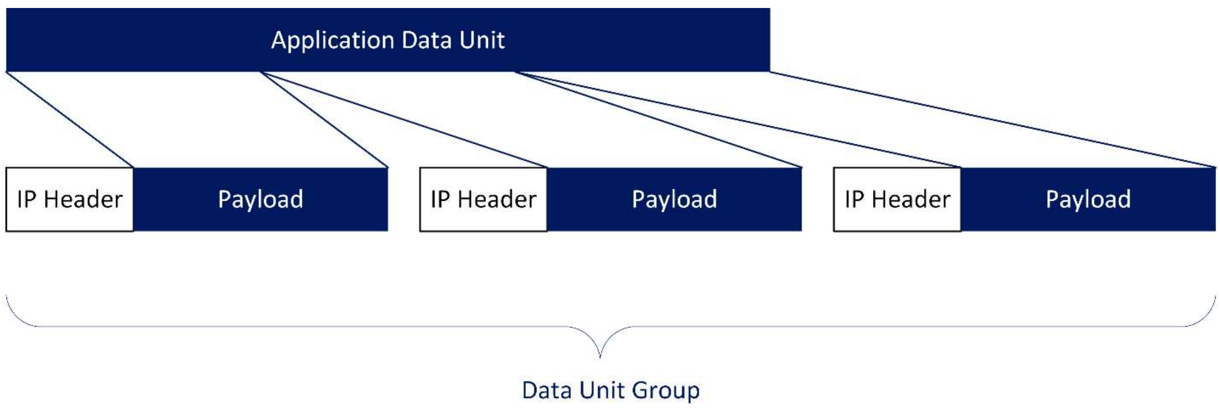


Figure 7: Data Unit Group and its Relationship with Individual Packets Carrying an Application Data Unit

Consequently, when the entire DUG is successfully exchanged between the sender and receiver, the receiving application will be able to retrieve the ADU as provided by the sender. In contrary, if a single packet went missing between the sender and receiver, the receiving application will not be able to read the ADU in its entirety.

3.2.3 Internet Protocol Version 6

The Internet Protocol in its both versions (Version 4 and Version 6) must be seen as the common lowest OSI layer denominator across a wide range of computer networks, making this protocol suite – and the IP protocol itself – an extremely significant component.

Note, there has been an effort in the IETF Transport Services Working Group (TSVWG) to define a User Datagram Protocol (UDP) solution to convey PDU Set information for application layer protocols that rely on UDP, e.g. HTTP/3 or QUIC. The published draft [29] defines a group of packets that should be handled similarly (e.g. all packets of a video I-frame) as a Media Data Unit (MDU). The draft defines a range of meta parameters as an extension header to UDP which can be directly mapped to PDU Set Markings.

As for the RTP header, the device which issues the actual packet will need to populate the correct information into the packet header(s). It is expected that any device which supports the Data Unit Group protocol extension (IPv4 or IPv6) will either implement their own IP packet header packet generation in user space and operates on raw sockets or comes with a modified kernel library and modified I/O socket libraries to allow the exchange of DUG-related information when handing over ADUs.

When looking at suitable protocol candidates for possible extension to support Data Unit Groups and allow the signaling of DUG information from generic internet applications to lower layers, the following constraints were identified. The chosen protocol:

- Allows extension by IETF without demanding backwards compatibility challenges for devices that do not support any proposed extension.
- Is supported by most communication devices.
- Does not prevent information access by switches and routers using encryption.

This document proposes the introduction of DUG as an extension to the IP header for both dominant IP versions, i.e. IPv4 and IPv6. As extending IPv4 and IPv6 headers follow different procedures set out in IETF, the solutions are presented in separate sub-sections.

IPv6 addresses the challenge to allow routers to perform faster in comparison to its earlier Version 4. IPv6 achieves this by allowing network nodes to only parse the IPv6 headers it requires to check to perform its actions. In order to comply with this requirement and to follow the guidelines on how to design IPv6 header extensions [20][21], i.e. using Type Length Values (TLVs), new IPv6 TLVs are defined, as illustrated in Figure 8.

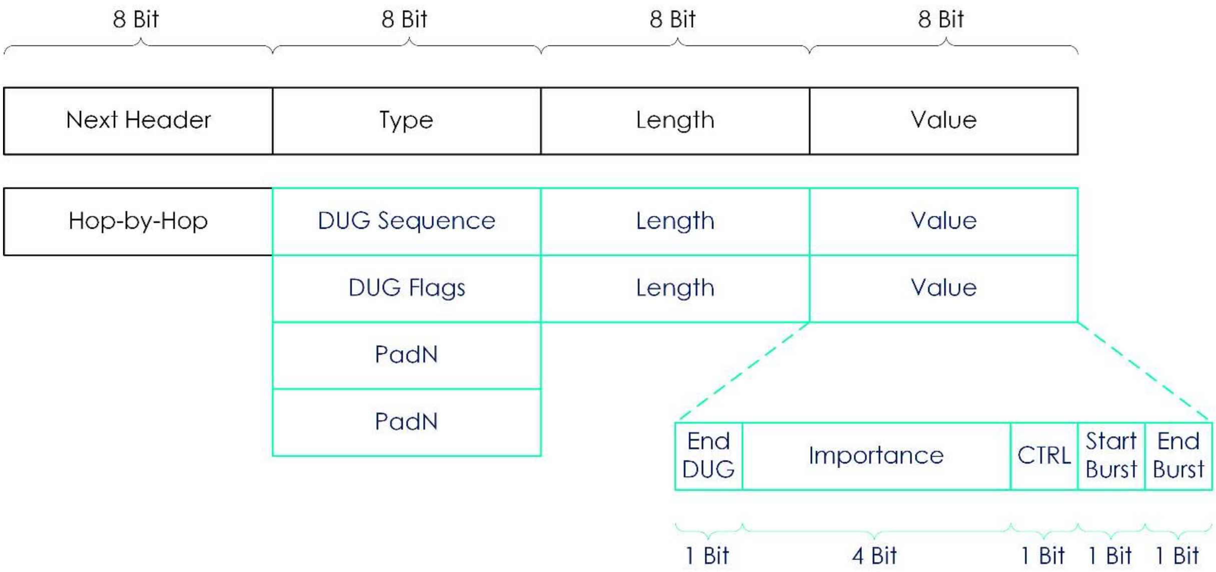


Figure 8: Proposed IPv6 Header Protocol Extension for Data Unit Groups

As the order of IPv6 extension headers is strongly recommended to follow a fixed order [21]in IPv6, the DUG-related information is added to an appropriate IPv6 header extension. The main purpose of the proposed DUG TLVs is for intermediate node to read the values and perform any sort of traffic engineering upon them. Thus, the "Hop-by-Hop" options header is identified as the most appropriate one, as intermediate nodes – such as 3GPP UEs/UPFs or TSN switches – shall process the DUG-related information. The configuration of intermediate IPv6 nodes may follow [30] and therefore the processing of the proposed DUG TLVs. The two new DUG TLVs defined are described in detail in Table 2.

Table 2: Type Length Values for the Proposed Data Unit Group IPv6 Extension Header

Type	Length	Description
Sequence	8 bit	An unsigned integer number indicating the order of packets which a Data Unit Group. With each new packet, the sequence number is iterated by 1.
Flags	8 bit	A set of flags providing more contextual information about the application payload this IP packet carries. A detailed list of all flags are provided in Table 3.

PadN	16 bit	Two padding fields of a total length of 2 bytes to make the DUG extension header a multiple of 8 bytes

The flags TLV has a set of values which are presented and described in the table below.

Table 3: Details of the Type Length Value Flag for the Proposed Data Unit Group IPv6 Header Extension

Flag Name	Length	Description
End of Data Unit Group	1 bit	This bit indicates the end of the DUG.
Importance of Data Unit Group	4 bits	These bits indicate the importance of this packet against other packets in the same DUG. Lower values indicate a higher importance and allow to prioritise packets in different DUGs. Importance may be derived from the application type or the sensitivity of applications when not receiving a DUG in their application.
Control	1 bit	When reliable transport protocols are in use, e.g. TCP, or upper-layer control procedures take place, e.g. establishment of a TLS session, there is no ADU exchanged. However, these control packets are equally important to the delivery of an ADU and depending on their functionality in the communication sometime even more important than the ADU itself. This bit allows to tag these packets.
Start Burst	1 bit	This field indicates the start of a burst of packets within a DUG.
End Burst	1 bit	This field indicates the end of a burst of packets within a DUG.

3.2.4 Internet Protocol Version 4

The Options field in the IPv4 header allows the addition of information to the IPv4 header and is variable in length with a maximum possible length of 40 bytes. Adding a new option requires to define the option by a 1 octet-long “option-type”, an “option-length” octet and the actual data in a multiple of 8 bits (1 octet) again. Furthermore, the “option-type” field comes in a pre-defined three-field octet indicating:

- 1 bit: Will the option been copied into all fragments in case IP fragmentation takes place.

- 2 bits: Identification of the classes the value represents (0: control; 1: reserved for future use; 2: debugging and measurement; 3: reserved for future use).
- 5 bits: The option number helps IP modules that read the header to interpret the meaning and implement a certain usage based on it.

All known option-type values are defined by IANA [23].

To extend the IPv4 header with DUG-related information, a single 2 octet-long option may be used. The option-type will be set to:

- Copy: 1
- Class: 0
- Value: To be decided based on IANA

The option-length field will indicate 16 bits with the structure of the option-value field as illustrated in Figure 9. The meaning of each field is identical to the IPv6 fields, as provided in Table 2.

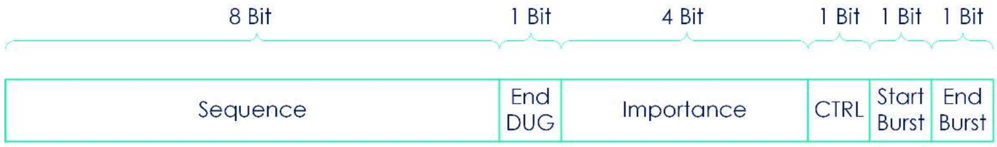


Figure 9: Proposed Data Unit Group Header Option for Internet Protocol Version 4

To indicate the end of the options list, a 1 octet of 0s is added to the end of the options list.

3.2.5 Data Unit Group Rules

The PCF may send Policy and Charging Control (PCC) Rules for a PDU Session to the SMF. The PCC Rules may indicate to the SMF that one or more IP Flows of the PDU Session are expected to carry packets with the DUG extension header.

During a PDU Session Establishment or PDU Session Modification procedure, the SMF may provide DUG Rules to the UE. The DUG Rules may indicate which IP/UDP Flows of the PDU Session are expected to carry packets with the DUG extension header. The DUG Rules may be sent to the UE in a PDU Session Establishment Accept message or a PDU Session Modification Command message. The DUG Rules may be part of the QoS Rules, or they may be part of any rule that lists IP Flows and allows the SMF it indicate for which flows the DUG extension header is enabled.

Alternatively, the DUG Rules may be sent to the UE by the PCF as part of a URSP Rule. For example, the indication maybe part of a traffic descriptor of a URSP Rule. In other words, the PCF may indicate that any traffic that matches the traffic descriptor has the extension header is enabled.

The DUG Rules may indicate certain actions that the UE should take when it detects a packet that matches the rule. One example action is whether to forward the matching packet to a local application or a local network port. Another example action is to determine PDU Set information from the packet header and forward the PDU Set Information to the upper-layer (e.g., SDAP or PDCP layer) of the RAN protocol stack. Another example action is to remove the DUG header option before forwarding the packet.

DUG Rules are categorized into detection and action rules for processing packets. DUG Detection Rules indicate the protocol header to check for, e.g. IPv4 or IPv6, and the exact header field, e.g. DUG header extension. The action then defines what to do with the packet once a Detection Rule found the referred header. Exemplary actions are “drop” or “remove DUG header”. Below, an exemplary representation using JSON.

```
[
        {
        “detection”: {
                “header_type”: “ipv4”,
                “header_field”: “extension”,
                “type”: “dug”
                },
        “action”: {
                “remove_extension”: “dug”
                }
        }
]
```

3.2.6 Example of DUG Operations in Uplink Scenarios for 3GPP UEs.

This section provides the procedures for identifying a DUG using the proposed IP header extensions with the outcome to map it to PDU Set Markings allowing the 3GPP User Plane to take advantage of the PDU Set capabilities.

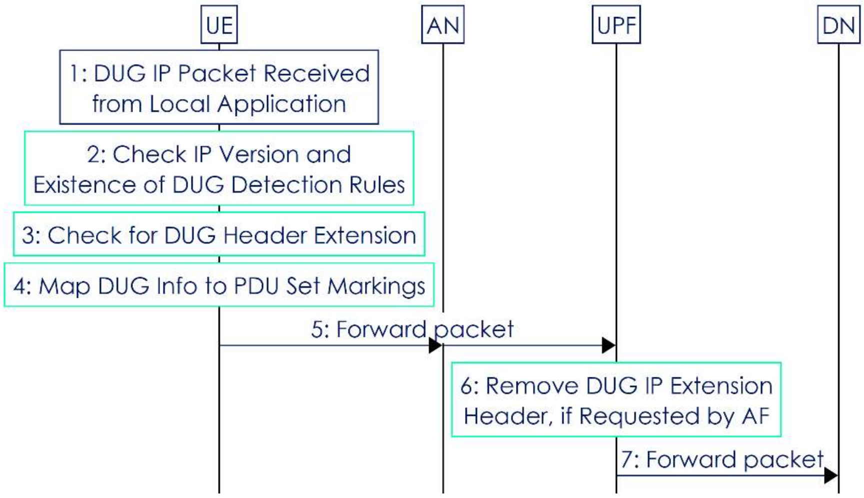


Figure 10: 3GPP Uplink Procedures for Data Unit Group IPv4 and IPv6 Packets

1. The UE receives an IP packet from an application with the DUG IP header extension being used.
2. The UE checks for the IP version of the packet from the local application based on the configured DUG Rules. The UE identifies the packet as an IP Version 4 or IP Version 6 packet.
3. If the IP packet is identified as Version 4, the UE checks for the existence of the IPv4 DUG TLVs in the options field. If found, Step 4 applies. If the UE identified the IP packet as Version 6, the UE checks for the hop-by-hop header extension and the presence of DUG TLVs. If found, Step 4 applies. If the UE cannot find any DUG-related header information, the packet is not identified as a member of a DUG and no further steps apply under this set of DUG Rules.
4. The UE identifies the DUG extension header information and writes the PDU Set Markings, as indicated in Table 4 for IPv4 and in Table 4 for IPv6. Note that the PDU Set Markings are currently standardized in 3GPP, and the tables below merely serves as an example on the DUG to PDU Set Markings mapping based on the current information.

Table 4: IPv4 DUG Header Extension Fields to PDU Set Marking Mapping

IPv4 Header Field	PDU Set Marking
Identifier	PDU Set Sequence Number
DUG Extension Header: Sequence	PDU Sequence Number within a PDU Set
DUG Extension Header: Flags::End of DUG	End PDU of the PDU Set

DUG Extension Header: Flags::End of Burst	End of Data Burst
DUG Extension Header: Flags::Importance	PDU Set Importance

Table 5: IPv6 DUG Header Extension Fields to PDU Set Marking Mapping

IPv6 Header Field	PDU Set Marking
Flow Label	PDU Set Sequence Number
DUG Extension Header: Sequence	PDU Sequence Number within a PDU Set
DUG Extension Header: Flags::End of DUG	End PDU of the PDU Set
DUG Extension Header: Flags::End of Burst	End of Data Burst
DUG Extension Header: Flags::Importance	PDU Set Importance

5. The UE sends the IP packet to the AN where the GTP-U information is written so that the AN can send the IP packet to the UPF. The AN forwards the packet to the UPF via GTP-U signaling.
6. To increase the compatibility with routers in the DN that do not implement the proposed DUG IPv4 or IPv6 header extension, the UPF may have received a configuration from the AF that indicates that the UPF should remove this extension header for each uplink packet that arrived via N3 before it leaves on N6. Also, if the packet traverses to another UPF via N9, the UPF may remove the DUG header extension (for IPv6)/option (for IPv4).
7. The UPF forwards the IP packet towards the DN or alternatively to another UPF.

3.2.7 Operations in TSN Networks

The extension of the DUG concept to the TSN domain is described in this section. In 802.1cb IEEE defines how packet streams can be identified and besides Layer 2-related header fields, e.g. source/destination MAC address or VLAN identifiers, IP header fields are considered already [24]. Any DUG-related configuration of TSN switches to identify packets belonging to the same information stream should be understood as an extension to 802.1cb.

A TSN switch first assess the priority of incoming packets before it is performing packet (re)scheduling tasks across its priority queues. visualizes in which part of the TSN operations the DUG information could be used to perform appropriate scheduling tasks and in case of timing constraints or time-related congestion, make decisions which packets have priority based on the knowledge of multiple packets belonging to a DUG. The impacted components are the:

- Priority Filter: The TSN switch allows to be configured which DUG Flags TLV value is used to place a packet into which queue. Depending on how many queues are available and what algorithms are available in the re-ordering and pre-emption engine, the mapping of DUG packets may me a 1:1 to a queue based on the importance field; alternatively, the CTRL bit or start/end of burst fields might be used to prioritize packets of the same DUG differently.
- Re-ordering/Pre-Emption Engine: As the information is available which packets belong to a DUG and their respective importance, the TSN engine may re-schedule or pre-empt all DUG packets within the cycle, or of deemed possible drop an entire DUG to free up resources.

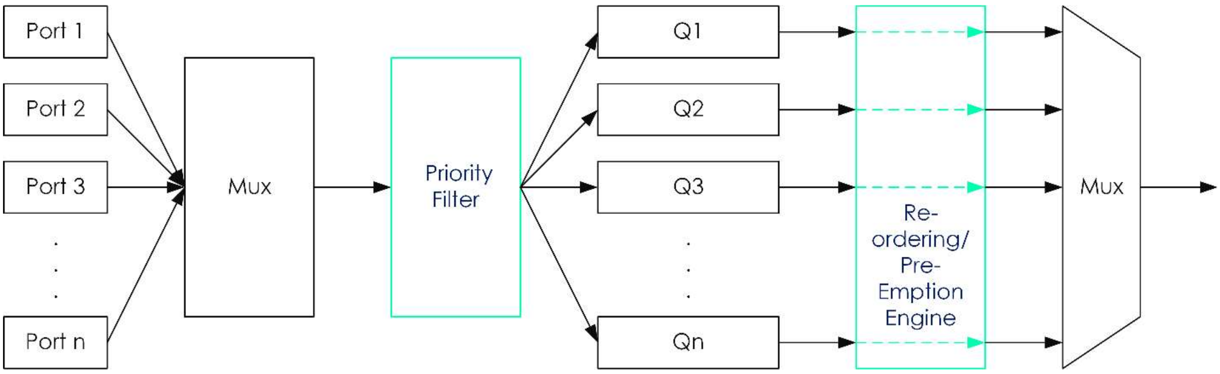


Figure 11: Internal TSN Switch Flow Operations for Data Unit Groups

4 Standalone Demos

4.1 Localization and Sensing

This subsection describes the development efforts for the localization and sensing use case [3] towards the implementation of the WP2 innovation on TSN over a PDU Set-Enabled 3GPP User Plane [4]. The implementation effort so far focused on:

- Extract sensing data from non-3GPP sensors
- Implement novel sensing framework compliant with future 3GPP systems (based on TS22.837 and recently established ETSI ISAC ISG [25]) that provide ISAC capabilities to coordinate sensing capabilities and allow the exposure of capabilities and resources via domain-specific management services for this demo
- Design and create local TRL 3 testbeds to test implementation for localization and sensing use case

The current testbed topology is provided in Figure 12and depicts physical nodes as black rectangles with their logical name and in brackets their hostname. Furthermore, all nodes have their network interface names and IP address assignments depicted with light blue as the DMZ network for remote management access (e.g., SSH) and the dark blue and pink interfaces as the User Plane interfaces. On the left, three UEs are illustrated which are connected to the gNB wirelessly (represented by a Wi-Fi-based station <> access point). The UE with hostname “gopigo202” is a vehicle (GoPico-based robotic [5]) with two distance and one video sensor and two rotors to move in any direction (forwards/backwards/left/right). The other two UEs (hostname pi203 and nuc151) are equipped with a Texas Instruments IWR1843 [6] radar sensor each allowing to detect objects in a configurable field of 90x90 degrees maximum. The nuc233 node on the middle top hosts selected Core Network services for managing the UEs, i.e., a User Data Management (UDM), Network Data Analytics Function (NWDAF), Analytics Data Repository Function (ADRF) and Network Exposure Function (NEF) service. The vertical applications on the right (VR Headset, Video Server and Xbox) are connected to the gNB via a UPF and separated in terms of IP subnets from the UEs/gNB. While the Video Server reads the video stream and prepares it to be displayed on any device (e.g., the VR Headset), the Xbox node has an Xbox gamepad connected to control the two rotors of the vehicle via an InterDigital-internally developed proprietary HTTP-based control framework [7] supporting a range of input and output devices.

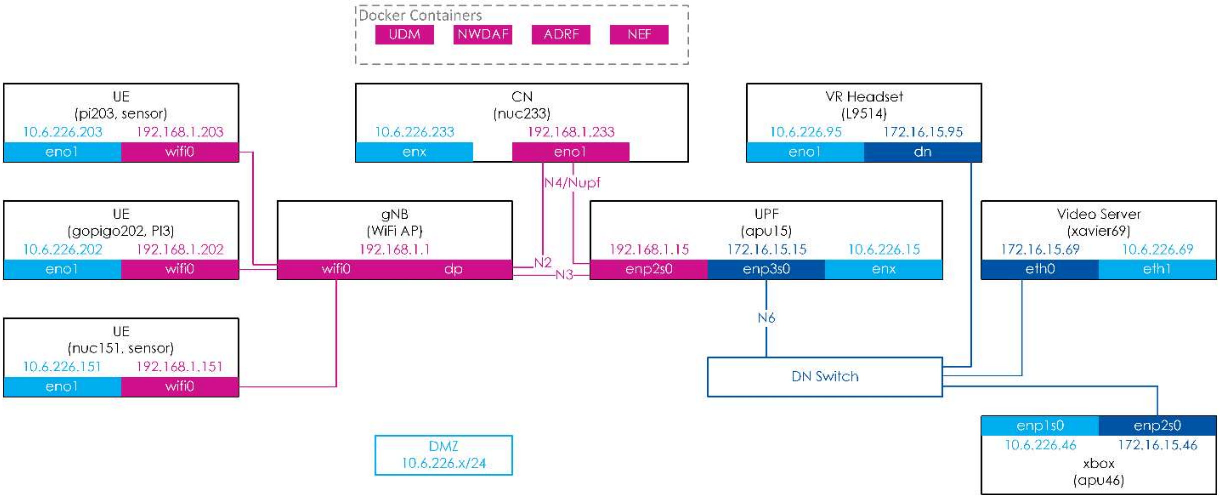


Figure 12: Network Topology of Localization and Sensing Testbed at InterDigital's Premise

To date, the sensing framework allows to collect and transfer sensing data from the distance and radar sensors to the ADRF using a proprietary software component on the UEs and InfluxDB 2.0 inside the ADRF. The NWDAF implements an analytics function to detect where the vehicle is in the field based on the radar information and sends the results periodically via the NEF (RabbitMQ (MQTT) message broker) to any vertical application (e.g., the VR Headset).

Besides publishing the sensing results via an MQTT broker, the NWDAF also stores all values in an InfluxDB, allowing to plot the sensing results in a web-based dashboard, as provided in

Figure 13. While the two row provides the horizontal and vertical distance to the first detected object by the radar sensors, the second row provides the measured distances to the first object by the vehicle's distance sensors.

Figure 13: Network Data Analytics Function Dashboard for Sensing

In addition to the dashboard to visualize the sensing results, an Operations Support System (OSS)-like dashboard has been realized using Telegraf, InfluxDB and Grafana. The following Telegraf plugins were enabled for this OSS dashboard: .cpu, .disk, .diskio, .kernel, .mem, and .net. The dashboard illustrated in Figure 14 allows to glance over the performance of the POC and categorizes all nodes into User Equipment (vehicle and the two sensors), network nodes (UPF and the host for all other Core Network functions), and the nodes in the Data Network (video server and Xbox controller host). This state-of-the-art collection of tools form the basis for the domain-specific Measurement Management Service of this demonstration, allowing to integrate with and end-to-end Measurement Management Service.

Figure 14: OSS-like Dashboard of the Sensing Proof-of-Concept Infrastructure

The software implementing the sensing framework is hosted on an InterDigital internally hosted Github Enterprise instance and all software is under a non-public InterDigital exclusive license. For the integration of this framework into 5TONIC, the software will be licensed to 5TONIC under an experimental royalty-free agreement for the duration of the PREDICT-6G project, where all software is provided as binaries.

Repository available **here**.

4.2 TSN on Wi-Fi

4.2.1 TSN redundancy to interference/attacks

4.2.1.1 Background

Path diversity can play an important role in providing deterministic communication services. The standard IEEE 802.1CB defines the TSN redundancy capability, to allow to transmit information through multiple paths. This redundancy can be used to increase the resilience of the network against interference and/or malicious attacks, as well as to improve the mobility of end-users towards a seamless roaming experience.

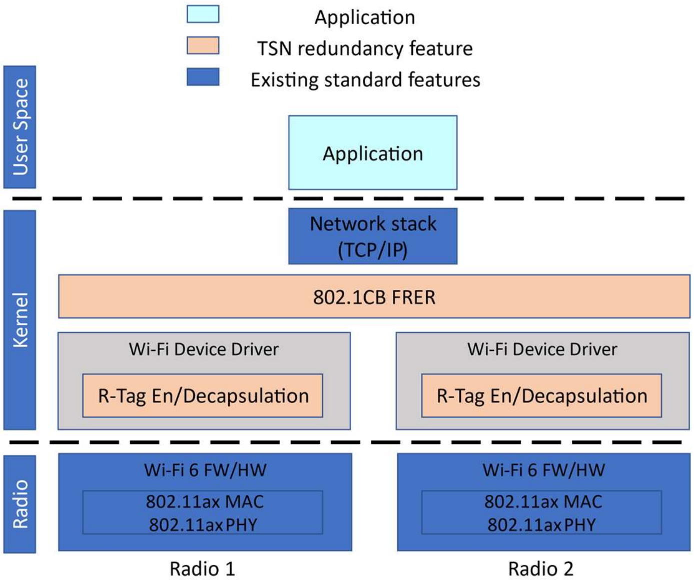


Figure 15: Path diversity (802.1CB) in Wi-Fi 6 TSN using two radio NICs

4.2.1.2 Results

We have published a study [8] on how to take advantage of the TSN redundancy capability (as defined in the IEEE 802.1CB standard) to eliminate outages or delays due to events like roaming and interference in a mobile robot use case enabled by Wi-Fi 6 TSN. Demonstrating the roaming performance with no delay impact on the applications though simulations of a mobile robot in a factory scenario and experimental results with a mobile robot connected via multiple Wi-Fi 6 radios in a warehouse environment. Figure 15 illustrates the communication stack, and Figure 16 shows the results in latency when using redundancy vs when not using redundancy.

Repository available **here**.

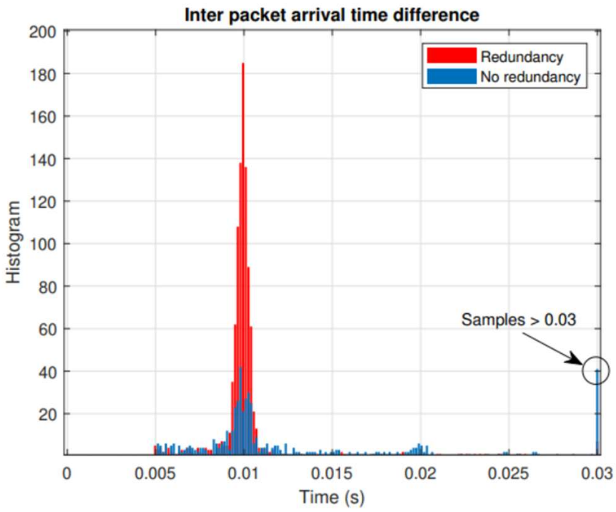


Figure 16: Inter packet arrival times w/o redundancy enabled. We observe most packets arriving near 10 ms intervals with redundancy, however a large group of packets arrive with significant delays greater than 30 ms when redundancy is not enabled.

4.2.2 Implementation with real-time and best effort, offset synchronization and latency evaluation.

802.1Qbv [15] represents an interesting opportunity to achieve deterministic communication, thanks to the scheduled traffic schemes it allows to implement. In practice, for a network in which all the nodes are synchronized (through the 802.1AS protocol) the slotted approach introduced by 802.1Qbv [15] ensures exclusive network access to the nodes. This is of paramount importance when wireless networks are used since such a technique allows to achieve an ordered access to the physical medium that avoids contentions.

In order to assess the potential of 802.1Qbv [15] over Wi-Fi, a main secondary protocol has been realized and tested on a collaborative robot application via simulations [9].

The application is described in Figure 17. It represents a semiautomated warehouse characterized by the contemporaneous presence of mobile devices and humans working in collaboration. The plant includes two distinct workspaces. The first one, highlighted in red, is the loading bay where both the unloading and loading of goods from trucks by human personnel take place manually. The second workspace, highlighted in green, is a fully automated area where mobile robots move goods to/from storage areas for further processing. In this workspace, the access of human personnel is forbidden.

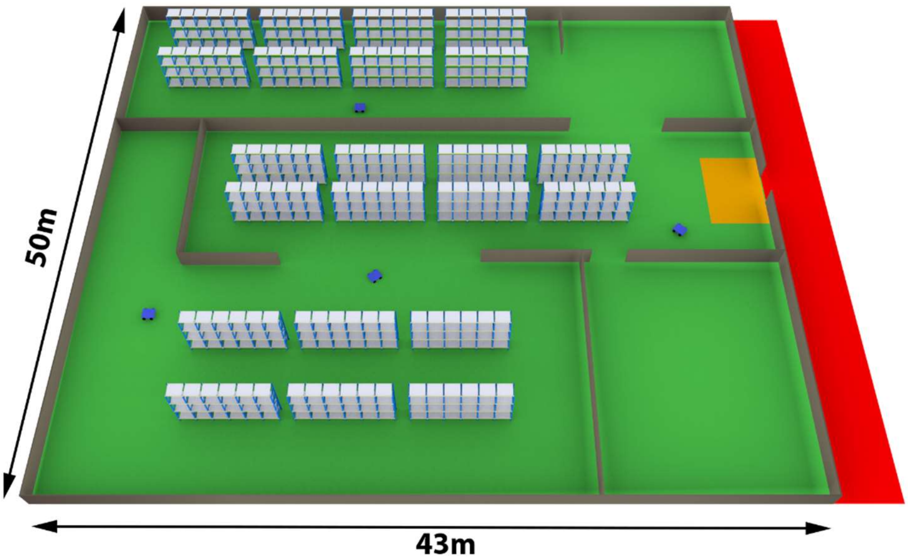


Figure 17: Semiautomated warehouse with mobile devices and human collaboration

Four autonomous mobile robots operate in the automated area. They represent mobile secondaries connected to a main located at the center, 5 m above the ground, of the automated area to provide an effective coverage. A further secondary, in this case static, is represented by the barriers placed at the entrance to the automated area.

The protocol assigns slots of fixed duration to main and secondary. The data exchange is organized in cycles. The main starts a cycle by sending a unique frame to all the secondaries in its slot. Subsequently, each secondary is granted to transmit its frame in the slot assigned to it. When the slot of the last secondary has expired a new cycle is started with the slot of the main. Clearly, TSN features are particularly helpful to implement such a technique since they allow (i) strict synchronization of the nodes and (ii) assigning of the time slots to the nodes.

The addressed performance index is the Percentage of Failed Polling (PFP), that accounts for the effectiveness of the protocol. Indeed, a polling is considered as failed when, within a cycle, either a secondary does not receive the frame sent by the main or the main does not receive the frame from a secondary.

To assess the improvements introduced by the adoption of TSN, a comparison with a non TSN based protocol has been carried out. This protocol, referred to as "continuous" implements a continuous polling, in which the main starts polling the first secondary and then subsequently moves to the next one. When the last secondary has been polled, a new cycle immediately starts with the first secondary.

The first set of simulations has been carried out for slot durations of, respectively, 5 ms for the main and 1 ms for each secondary. The failed polling statistics are reported in Table 6.

Table 6: Failed polling statistics.

Polling	**Failed Polling**	**Total Cycles**	**Total *PFP* (%)**
Continuous	670	12522453	0.005350
Slotted	0	613417	0.0

As can be seen, the benefits of TSN are evident, since no failed polling were detected when the slotted TSN-based protocol was adopted. Conversely, this was not the case of the continuous approach that shows a high number of failed polling.

In a second set of simulations, a best effort (BE) traffic was introduced in the network. The BE traffic comprises five TCP streams at a constant rate of 5 Mbps each, simulating video streams transmitted by the secondaries to the main during each cycle. The percentage of failed polling is reported in Table 7.

Table 7: Failed polling statistics with BE traffic

Polling	**Failed Polling**	**Total Cycles**	**Total *PFP* (%)**
Continuous	17629	1506478	1.170213
Slotted	1753	613417	0.285776

As expected, the behavior of the continuous technique results worsened since the access to the physical medium is completely unregulated, and the main-secondary traffic is mixed up with the BE one.

Conversely, the slotted TSN-based technique performs better. However, surprisingly, a non-negligible percentage of failed polling was observed for this technique as well. A more in-depth analysis revealed that this problem was due to a lack of harmonization between 802.1Qbv and Wi-Fi. In practice, it has been detected those typical events related to the Wi-Fi protocol behavior, such as retransmissions of formerly queued frames, random backoff times, and beacons, interfere with the main-secondary data exchange causing delays that lead to failures in the polling.

5 Integrated Demos

5.1 Non-deterministic devices and networks

5.1.1 Flow monitoring for non-deterministic domains

This section provides an overview of the implementation of a flow monitoring capability suitable for producing in-line measurements on network segments where the underlying technology does not natively provide such measurements, or at least not with the granularity and accuracy required to collect flow level performance related to deterministic service requirements.

The implementation consists of Measurement Agents (MA), where each MA is implemented as a packet processing software module using Intel's Data Plane Development Kit (DPDK) [12]. Measurement Agents may be deployed on any Linux-based physical or virtual machine, given that they are looped into the data plane of the monitored network segment so that they have access to the packets of the monitored data flows. A single MAs may be deployed as standalone agent, or multiple (two or more) MAs may be deployed along different locations on an end-to-end path, in which case they provide cooperative measurements. Deployments are illustrated in Figure 18.

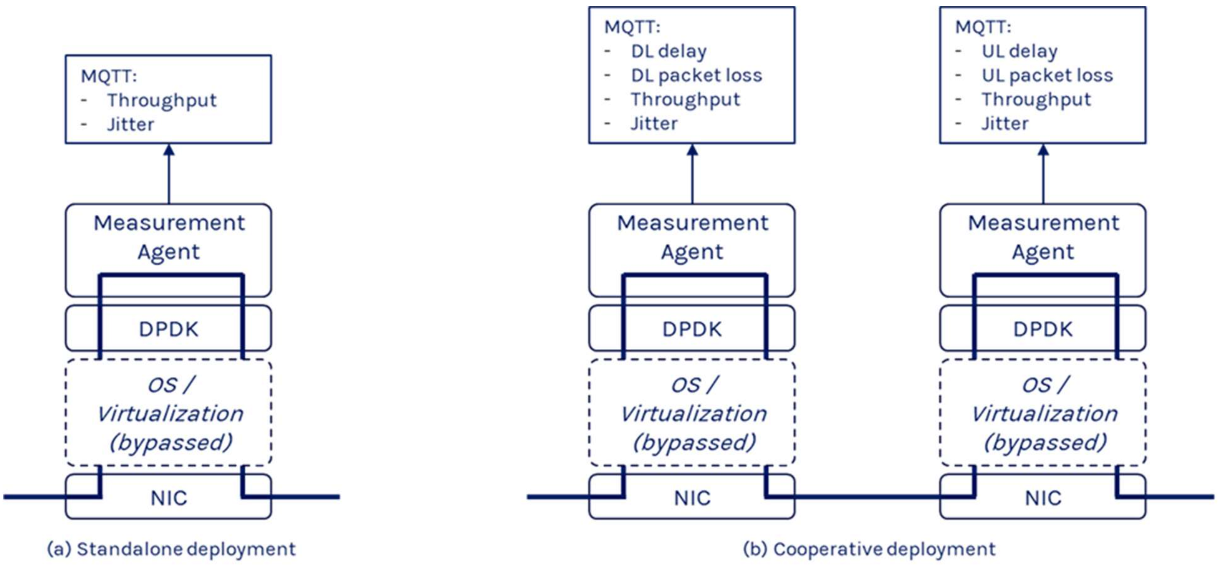


Figure 18: Deployment of MAs: (a) standalone deployment of a single agent; (b) cooperative deployment of multiple agents.

An individual MA supports the following measurements:

- throughput measurement separately in UL/DL direction, with a configurable aggregation (e.g., individual flow-based measurements, or flow aggregates) and on the time granularity (e.g., 10-100-1000 ms measurement windows)
- jitter measurement (based on packet inter-arrival times), with configurable aggregation and granularity (similarly as above)

Multiple MAs deployed at different locations on an end-to-end path enables the following additional measurements:

- directional delay measurements (DL/UL) between any two adjacent MAs, with configurable aggregation and granularity (see above)
- directional packet loss measurements (DL/UL) between any two adjacent MAs, with configurable aggregation and granularity (see above)

The MAs take advantage of the available protocol headers adaptively detected on the packets to insert timestamps (for directional delay measurements) and sequence numbers (for directional packet loss measurements). In decreasing order of preference, the following protocol headers are used:

- GTP-U extension headers (when the MA is deployed on a 3GPP N3 interface)
- TCP option headers (when the packet contains a TCP header)
- IP option headers

The MAs report the measurement results to a message bus implemented with the Message Queuing Telemetry Transport (MQTT) protocol [13]. Different measurements are posted to different topics, allowing consumers of the monitoring to selectively subscribe to their data of interest. The topics are configurable to enable integration to different data collection systems.

The code of the implementation is not public but available to reviewers on a request (NOK contact: Péter Szilágyi, peter.1.szilagyi@nokia-bell-labs.com).

5.2 Industrial cross-domain TSN

5.2.1 Development of extensions for operation of 5GS as TSN bridge. Time-based schedulers on 3GPP

During the first year of the project, we focused on the characterization of Time Sensitive Networking (TSN) features using the 5G System. According to TS 23.501 [10], 5G System can be integrated as a TSN bridge by using TSN-Translators at Device and Network side (DS-TT and NW-TT respectively). In Deliverable 2.1

[4], we described the current setup of the Smart Factory use case [11] which is included in the *Industrial Cross-Domain TSN* activity. For doing the characterization we provided a software switch with TSN capabilities that can behave as TSN translator in UE and Network side (DS-TT and NW-TT respectively). Some of the current features are:

- Provide TSN ports at TSN Translators points.
- Support of 802.1Qb [26] time-based scheduling with time slices and guard period.
- Support of Frame replication 802.1CB [27].
- Compatible 802.1Qcp [28] YANG model

Characterization is ongoing and we foresee to deliver public results during 2023/early 2024.e

Repository available **here**.

5.2.2 802.1Qbv TSN bridge based on XDP

In accordance with the objectives outlined in Deliverable D2.1, this section provides an overview of the initial implementation of the XDP 802.1Qbv [15] bridge within the framework of the Prefict6G project. The implementation leverages the cutting-edge technologies of eBPF and XDP to achieve efficient, high-performance and TSN switching.

The implementation process followed the guidelines and specifications laid out in Deliverable D2.1. We have realized the first version of the switch, integrating some of the proposed technologies. The codebase has been made publicly available on our project's GitLab repository for reference and collaboration.

The main achievements of this first version are:

1. Switch Implementation

The XDP bridge has been implemented successfully regarding its switching capabilities, as stated in D2.1. Using eBPF, we have achieved a flexible and programmable switching infrastructure and a high-performance switch.

2. Queuing System

The queuing system is being developed, laying the foundation for future integration with the Qbv implementation. While the queuing system has been incorporated into the codebase, its currently untested and on hold. All the measurements have been made by using single queuing.

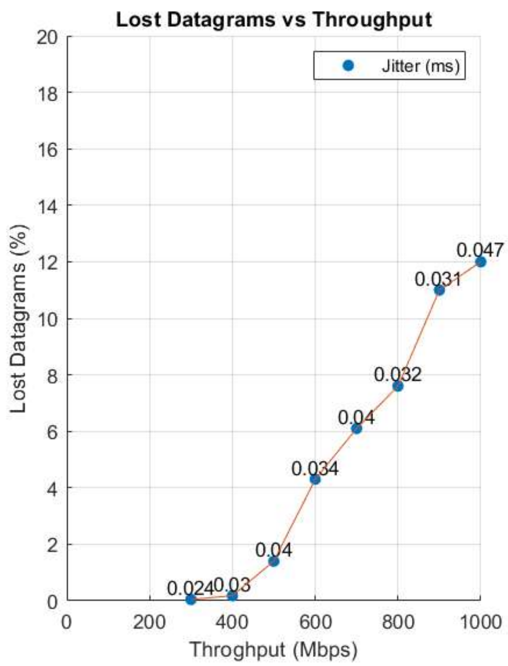
3. Characterization and Performance Results

To assess the efficacy of our XDP 802.1Qbv bridge implementation, testing has been conducted on real machines. The results are summarized in the accompanying Figure 19. These characterizations provide insights into the switch's performance metrics, as well as different traffic characterization. In the preliminary phase of our study, we employed both TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) to generate network traffic across the switch. This approach enabled us to assess performance across a spectrum of bandwidth capacities. We strategically varied the window sizes, utilized different ports, and initiated parallel traffic streams to simulate a realistic network environment. Additionally, we established multiple connections simultaneously to further stress-test the network infrastructure.

Beyond these parameters, our experiment also included prolonged sessions of traffic exchange to observe the network's stability and endurance over extended periods. Basic ping tests were conducted as well, serving as a foundational metric for network responsiveness and latency.

The results are based on a work in progress status, comparing our switch capabilities versus a high end not TSN Switch.

Repository available **here**.



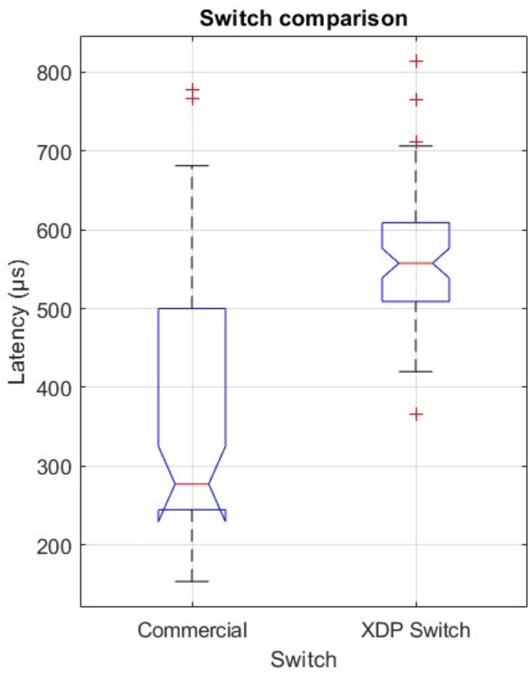


Figure 19: Performance comparison of XDP Switch

5.2.3 DetNet implementation up to L3

This task involves the practical implementation of two documents published by the Deterministic Networking Working Group of the IETF, previously explained in D2.1:

- RFC 8964: Deterministic Networking (DetNet) Data Plane: MPLS
- RFC 9025: Deterministic Networking (DetNet) Data Plane: MPLS over UDP/IP

Both implementations incorporate two of the most intriguing techniques provided by the concept of DetNet:

- Explicit Routes: Assignment of predetermined, unchanging routes so that data flows always follow the same paths. In this implementation, these routes are established based on static routes defined in the "rules/" folder.
- PREOF (Packet Replication Elimination and Ordering Functions) Services: These services are responsible for replicating packets across different network paths, eliminating duplicate packets, and ordering packets before they reach their destination.

The presented network topology in Figure 20 represents the tested configuration of DetNet. Through examination and validation, this topology has been tested successfully and it is verified for traffic processing up to and including layer 3.

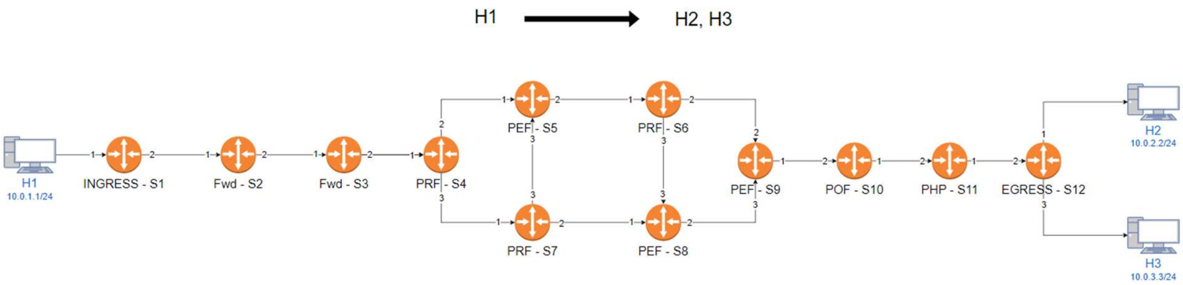


Figure 20: DetNet topology tested.

The following functionalities are considered:

- Ingress: Access point to the DetNet network. According to source and destiny, IP addresses introduce a sequence number (d-CW) a Service-Label and one or various Forwarding –Labels.
    - Service Label: An MPLS label is added upon entry into the DetNet network and can be exchanged within the network. It instructs the receiving node on which PREOF function to perform on the received packet.
    - Forwarding Label: Indicates to the receiving node whether to send the packet to the service sub-layer or to forward it back into the network.
- Egress: Exit of the DetNet network. It decapsulates the original message (ICMP in this case) and forwards it to the destiny host.

Among the capabilities contemplated on the RFCs previously mentioned, we have implemented:

- PHP: Penultimate Hop Popping, inherited from MPLS removes Forwarding Labels before reaching the Egress node.
- PRF: Replication function node
- PEF: Elimination function node
- FWD: Packets are not uploaded to service layer but simply forwarded according to the F-Labels

The next step involves the testing of these DetNet implementations in conjunction with an Extended Berkeley Packet Filter (eBPF) backend. Incorporating eBPF into our implementation enables us to add our P4 DetNet implementation as an additional layer to a specified network topology.

In this context, we are particularly interested in applying our implementation to the experiments, as they offer a suitable environment for assessing the real-world performance and reliability of DetNet within different network scenarios.

Repository available **here**.

5.2.4 Deterministic end-to-end multidomain use case.

To create a comprehensive scenario that involves different time-sensitive domains, we can envision a network setup where various components are interconnected, demonstrating the flow of information and control across distinct but integrated technologies. This setup will involve four key stages:

- **TSN with Wi-Fi 6 and TSN Switch:** This stage includes a TSN Wi-Fi 6 environment and a TSN switch. These components are connected to an end-user device, such as a robotic dog, enabling real-time communication and control. The robotic dog represents a practical application of TSN capabilities, requiring precise timing and low latency for its operations.

- **OAI 5G SA Network:** Here, a 5G User Equipment (UE) is connected to an OpenAirInterface (OAI) gNodeB. This connection extends to the OAI core network, where the traffic is routed through a User Plane Function (UPF) to the next stage.
- **Ericsson 5G SA Network**: The traffic from the previous stage reaches an Ericsson UPF, part of the Ericsson 5G Core (5GC) network. This UPF is connected to an Ericsson gNodeB and another 5G UE. This stage demonstrates the interoperability between different vendors and technologies within the 5G ecosystem, ensuring seamless communication and data flow.
- **Telefonica TSN Switches:** Finally, the network setup includes Telefonica TSN switches forming a bridge, culminating in a TSN end application. This end application could be a critical industrial control system or a high precision manufacturing tool, showcasing the application of TSN technology in a real-world scenario.

This scenario, shown in Figure 22, effectively demonstrates how Wi-Fi 6, and 5G technologies can be interconnected to provide a seamless, deterministic, and high-performance network solution, catering to applications that require precise timing, high reliability, and low latency. The setup highlights the synergy between different technological domains, from local area networking with TSN to wide-area connectivity with 5G, all integrated to serve complex, time-sensitive applications.

This is the evolution of the architecture presented on D2.1, detailing the involved domains Implementation of the different domains are being developed by the different partners, and updates will be presented in future deliverables.

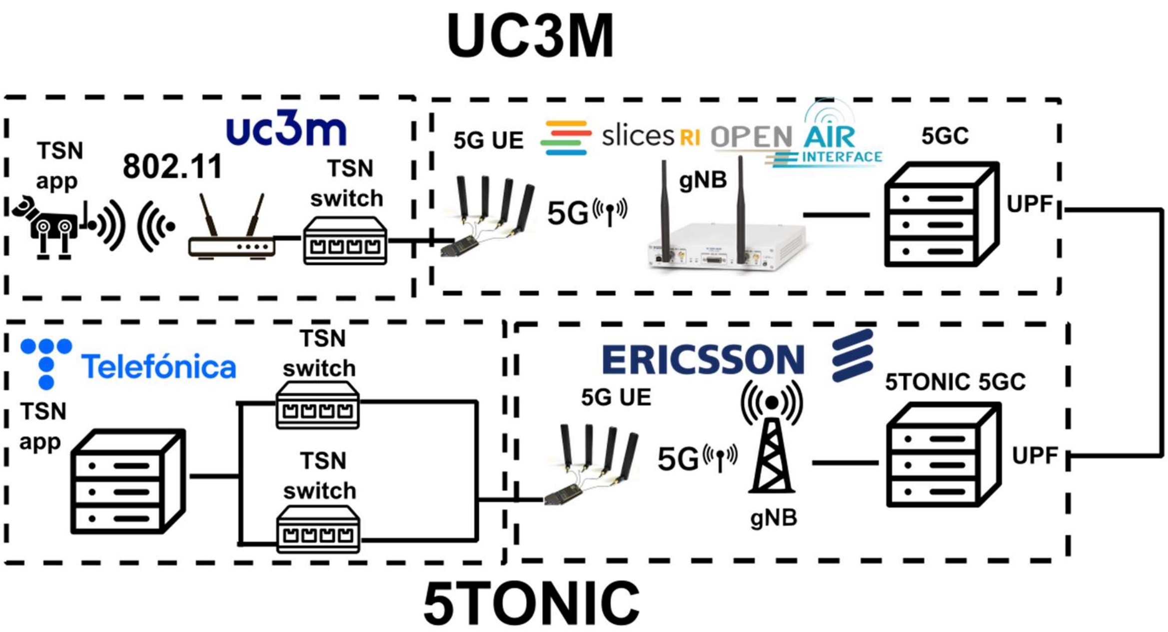


Figure 21 - Multidomain architecture with partners

Figure 22: Testbed in UC3M

6 References

[1] R. Croonenbroeck, L. Underberg, A. Wulf, and R. Kays, "Measurements for the development of an enhanced model for wireless channels in industrial environments," in Proc. 2017 IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob), Rome, Italy, 2017, pp. 1-8.

[2] A. A. Esswie, K. I. Pedersen, and P. E. Mogensen, "Online radio pattern optimization based on dual reinforcement-learning approach for 5G URLLC networks," IEEE Access, vol. 8, pp. 132922-132936, 2020.

[3] PREDICT-6G, "Analysis of use cases and system requirements," Jun. 2023.

[4] PREDICT-6G, "Release 1 of PREDICT-6G MDP innovations," Jun. 2023.

[5] "GoPiGo," GoPiGo.io, 2023. [Online]. Available: https://gopigo.io/gopigo/. [Accessed: Nov. 7, 2023].

[6] "IWR1843BOOST," Texas Instruments, 2023. [Online]. Available: https://www.ti.com/tool/IWR1843BOOST. [Accessed: Nov. 7, 2023].

[7] "RFC InterDigital GitHub," GitHub, 2023. [Online]. Available: https://github.interdigital.com/rcf.

[8] S. Sudhakaran et al., "Zero-delay roaming for mobile robots enabled by wireless TSN redundancy," in Proc. 2023 IEEE 19th Int. Conf. Factory Commun. Syst. (WFCS), 2023, pp. 1–8.

[9] A. Morato, S. Vitturi, F. Tramarin, C. Zunino, and M. Cheminod, "Time-sensitive networking to improve the performance of distributed functional safety systems implemented over Wi-Fi," Sensors, vol. 23, no. 7825, 2023. [Online]. Available: https://doi.org/10.3390/s23187825.

[10] ETSI, "TS 23 501 - v15.2.0 - 5G; system architecture for the 5G system," ETSI, 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/15.02.00_60/ts_123501v150200p.pdf. [Accessed: Nov. 7, 2023].

[11] PREDICT-6G, "Release 1 of PREDICT-6G MDP innovations, Figure 29," Jun. 2023.

[12] "Data Plane Development Kit (DPDK)," Linux Foundation, 2023. [Online]. Available: https://www.dpdk.org/.

[13] "Message Queuing Telemetry Transport (MQTT) protocol," MQTT.org, 2023. [Online]. Available: https://mqtt.org/.

[14] 3GPP, "TS 38.300," 3GPP Portal TS 38.300, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3191 (accessed Nov. 21, 2023).

[15] "IEEE Standard for Local and metropolitan area networks--Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks," in IEEE Std 802.1Q-2011 (Revision of IEEE Std 802.1Q-2005) , vol., no., pp.1-1365, 31 Aug. 2011, doi: 10.1109/IEEESTD.2011.6009146.

[16] Chen, Q., and Zhu, Y.-H. Scheduling channel access based on target wake time mechanism in 802.11ax wlans. IEEE Transactions on Wireless Communications 20, 3 (2021), 1529–1543.

[17] Le Boudec, J.-Y., and Thiran, P. Network calculus: a theory of deterministic queuing systems for the internet. Springer, 2001.

[18] Schneider, B., Sofia, R. C., and Kovatsch, M. A proposal for time-aware scheduling in wireless industrial iot environments. In NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium (2022), pp. 1–6.

[19] Wang, H., Schmitt, J., and Ciucu, F. Performance modelling and analysis of unreliable links with retransmissions using network calculus. In Proceedings of the 2013 25th International Teletraffic Congress (ITC) (2013), pp. 1–9.

[20] S. Krishnan, J. Woodyatt, E. Kline, J. Hoagland and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, 2012. DOI: 10.17487/RFC6564

[21] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 8200, 2017. DOI: 10.17487/RFC8200

[22] RFC Editors, "INTERNET PROTOCOL", 1981. DOI: 10.17487/RFC0791

[23] IANA, "Internet Protocol Version 4 Parameters", 2018. Online: https://www.iana.org/assignments/ip-parameters/ip-parameters.xhtml

[24] IEEE, "Standard for Local and metropolitan area networks – Frame Replication and Elimination for Reliability", 802.1CB, 2017.

[25] ETSI, "Integrated Sensing And Communications". 2023. Online: https://portal.etsi.org/tb.aspx?tbid=912&SubTB=912

[26] "IEEE Standard for Local and metropolitan area networks -- Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic," in IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015) , vol., no., pp.1-57, 18 March 2016, doi: 10.1109/IEEESTD.2016.8613095.

[27] "IEEE Standard for Local and metropolitan area networks--Frame Replication and Elimination for Reliability," in IEEE Std 802.1CB-2017 , vol., no., pp.1-102, 27 Oct. 2017, doi: 10.1109/IEEESTD.2017.8091139.

[28] "IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks--Amendment 30: YANG Data Model," in IEEE Std 802.1Qcp-2018 (Amendment to IEEE Std 802.1Q-2018) , vol., no., pp.1-93, 14 Sept. 2018, doi: 10.1109/IEEESTD.2018.8467507.

[29] J. Kaippallimalil, S. Gundavelli, and S. Dawkins, "Media Header Extensions for Wireless Networks", IETF Internet Draft, TSVWG, 2023. Online: https://datatracker.ietf.org/doc/html/draft-kaippallimalil-tsvwg-media-hdr-wireless-03

[30] B. Hinden and G. Fairhurst, "IPv6 Hop-by-Hop Options Processing Procedures", Online: draft-ietf-6man-hbh-processing-16 - IPv6 Hop-by-Hop Options Processing Procedures

7 Annex

The implementations provided in this document can be found in the GIT repositories listed in the following table. Please note that these repositories are subject to continuous upgrades and improvements, so we recommend checking them regularly for the latest updates and enhancements.

Repository name	GIT URL
Dynamic Scheduling	https://gitlab.netcom.it.uc3m.es/predict-6g/dynamic-scheduling-upc
P6 TSN XDP Switch	https://gitlab.netcom.it.uc3m.es/predict-6g/p6-tsn-xdp-switch
P6 DetNet	https://gitlab.netcom.it.uc3m.es/predict-6g/P6-DetNet
Sensing Framework	https://gitlab.netcom.it.uc3m.es/predict-6g/ide-sensing-framework
TSN redundancy to interference/attacks	https://gitlab.netcom.it.uc3m.es/predict-6g/intel-tsn-redundancy-to-interference-attacks
Time-based schedulers on 3GPP	https://github.com/5Tonic/gopsa
Aligning rTWT and IEEE 802.1Qbv	https://gitlab.netcom.it.uc3m.es/predict-6g/twtscheduler