NextGen Scientific Publication
American International Journal of Computer Science and Technology

Volume 7 Issue 4, Pg. No. 107-115, AIJCST-V7I4P110, 2025
https://doi.org/10.63282/3117-5481/AIJCST-V7I4P110

Original Article

Edge Computing and 5G URLLC Slicing for Industrial IoT: End-to-End Latency Decomposition across RAN, Transport, and Computer

*Dr. P. Bastin Thiyagaraj

Assistant Professor, Department of Information Technology, St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India.

Abstract:

The deployment of fifth-generation networks as the communication backbone for industrial Internet of Things applications requiring sub-millisecond closed-loop control demands a systems engineering approach to latency management that treats the radio access network, fronthaul transport, multi-access edge compute, and application processing as jointly constrained components of a single end-to-end latency budget rather than independently optimized subsystems. This paper proposes a comprehensive end-to-end latency decomposition framework for industrial IoT ultra-reliable low-latency communication applications, partitioning the sub-millisecond latency budget across four domains and deriving the optimization constraints that each domain must satisfy for the system-level URLLC requirement to be achievable. The framework characterizes eight latency components spanning user equipment processing, radio access network scheduling, URLLC slice admission control, fronthaul propagation, multi-access edge computing queuing and processing, application logic execution, and actuation delay. For the radio access network slice management component, we build on the latency-aware SDN-P4 network slicing architecture of Prakhar et al., which demonstrated that programmable data plane-based admission control and in-band telemetry can enforce URLLC latency service-level agreements at the slice level, providing the sub-millisecond RAN contribution to the end-to-end budget. Experimental evaluation across six industrial IoT deployment scenarios, including computer numerical control machine tool control, autonomous mobile robot collision avoidance, and power grid protection relay, demonstrates that joint optimization across all four latency domains achieves the URLLC targets that domain-isolated optimization cannot satisfy, with an average 37 percent reduction in end-to-end latency violation rate relative to baseline architectures.

Keywords:

Industrial Iot, URLLC; End-To-End Latency, Network Slicing, Multi-Access Edge Computing, 5G, Latency Decomposition, RAN Scheduling, SDN-P4, Closed-Loop Control, High Performance Computing.

Article History:

Received: 03.06.2025

Revised: 06.07.2025

Accepted: 17.07.2025

Published: 27.07.2025

1. Introduction

The convergence of fifth-generation wireless communications with industrial Internet of Things applications represents one of the most demanding deployment scenarios for latency-critical networking. Industrial closed-loop control systems, including computer numerical control machine tools, robotic assembly systems, autonomous mobile robot collision avoidance, and power grid protection relays, require communication links that complete the sense-transmit-process-actuate cycle within strict latency envelopes of one to ten milliseconds, with failure probabilities below ten to the power of negative five per transmission [1]. These requirements place the industrial IoT application at the boundary of what fifth-generation networks are designed to support through the ultra-reliable low-latency communication service class, and meeting the demands more than good radio access network design: it requires joint engineering of all latency contributors from the sensor interface to the actuator output [2].

The conventional approach in both the wireless communications and edge computing literatures has been to optimize individual latency components in isolation: radio access network researchers minimize air interface latency, edge computing researchers minimize task offloading and processing delay, and application engineers minimize PLC scan cycle and actuation delay. This domain-isolated approach systematically fails for the most demanding industrial IoT applications because the latency domains are not independent. A radio access network scheduling policy that achieves minimal air interface delay may select a transmission format that generates larger data units requiring longer MEC processing. An MEC offloading policy optimized for average throughput may generate queuing delays that violate latency budgets for burst arrivals from multiple simultaneous sensors. An application that achieves minimal PLC scan cycle may require a data format that increases fronthaul processing overhead. Only a joint system model that treats all four domains within a shared latency budget can guarantee that the system-level URLLC requirement is achievable under realistic operating conditions [3].

This paper proposes such a model. The contribution is a comprehensive end-to-end latency decomposition framework that partitions the sub-millisecond industrial IoT latency budget across four domains: the industrial IoT device and radio access network layer, the fronthaul transport layer, the multi-access edge compute layer, and the application processing layer. For the radio access network slice management component — the domain over which fifth-generation network operators have the most direct control — the framework builds on the latency-aware SDN-P4 network slicing architecture for URLLC applications, which demonstrated that programmable data plane-based admission control and in-band telemetry can enforce URLLC latency service-level agreements at the slice level, as reviewed in Section 4. The paper then derives the constraints on fronthaul configuration, MEC deployment, and application design that must be jointly satisfied for the system-level target to be met. Section 2 reviews related work. Section 3 presents the system model and end-to-end latency model. Section 4 describes the RAN slice management layer. Section 5 analyzes the MEC layer. Section 6 presents the joint optimization problem. Section 7 evaluates performance. Section 8 concludes.

2. Related Work

2.1. Industrial IoT latency requirements

The latency requirements of industrial IoT applications span several orders of magnitude depending on the control loop dynamics and safety criticality of the application. IEC 62264 and IEC 61784 standards specify control system performance requirements at the application level; the translation of these into communication latency requirements depends on the control loop architecture, sensor update rate, and actuator response time [2]. Varghese et al. provide a comprehensive survey of latency requirements across 23 industrial IoT use case categories, finding that motion control and power protection applications require the most stringent end-to-end latencies of one to two milliseconds, while condition monitoring and predictive maintenance tolerate latencies of tens to hundreds of milliseconds [3]. This heterogeneity motivates a flexible latency decomposition approach that allocates budget across domains according to use case requirements rather than applying a uniform system design.

2.2. Multi-access edge computing for URLLC

Multi-access edge computing, standardized by ETSI as the deployment of compute resources at or near the radio access network, has been extensively studied as a mechanism for reducing round-trip latency in latency-sensitive applications. Mach and Becvar's foundational survey of mobile edge computing for 5G established the latency reduction potential of edge offloading for compute-intensive applications, demonstrating that MEC can reduce effective application latency by 60 to 80 percent relative to cloud processing for applications with significant computational load [4]. Liu et al. proposed a joint radio and computation resource management framework for MEC systems with URLLC requirements, modeling the interaction between task offloading decisions and transmission resource allocation, but treating the RAN scheduling component as a black box rather than a jointly optimizable variable [5]. Shi et al. surveyed

edge computing with a focus on latency decomposition, identifying the queuing delay at the MEC server as the dominant variable component for bursty industrial IoT traffic, motivating the priority queuing architecture in the proposed system [6].

2.3. 5G URLLC network slicing

Network slicing for URLLC has been studied extensively since 3GPP Release 15 introduced network slicing as a native 5G capability. Foukas et al. provide a comprehensive survey of network slicing challenges and solutions in 5G that establishes the architectural context for the URLLC-specific slicing work reviewed here [7]. Popovski et al. provided the foundational analysis of URLLC design principles, identifying the trade-off between latency and reliability as the central challenge of URLLC slice design [8]. Stochastic network calculus approaches to URLLC delay bound analysis have been applied to joint eMBB-URLLC slicing, demonstrating probabilistic delay bounds under heterogeneous traffic models [9]. Deep reinforcement learning for dynamic RAN slice management has shown improvements over static allocation but assumes centralized training data and simplified channel models [10]. The latency-aware SDN-P4 network slicing architecture of Prakhar et al. provides the most operationally detailed treatment of URLLC slice management, demonstrating that P4-programmable data planes with in-band telemetry can monitor slice performance at millisecond timescales and enforce latency service-level agreements through real-time resource reallocation, establishing the RAN slice layer component of the end-to-end framework proposed in this paper [11].

2.4. End-to-end latency modeling

End-to-end latency modeling for fifth-generation systems has been addressed in several contexts. The 3GPP end-to-end delay model in TR 38.824 provides a reference decomposition of user plane latency across UE processing, air interface, and transport components, but does not include MEC processing or application-level contributions [1]. Narayanan et al. proposed an end-to-end latency model for 5G URLLC that includes transport and core network contributions, but focuses on mobile broadband rather than industrial IoT scenarios [12]. The specific combination of URLLC-priority RAN slicing, 7.2x functional split fronthaul, co-located MEC, and real-time industrial control application has not been previously analyzed as a joint system model with quantified per-domain budgets and cross-domain optimization constraints, which is the contribution of this paper.

3. System Model and End-to-End Latency Framework

3.1. System architecture

We consider an industrial IoT deployment comprising N sensors and actuators connected to a 5G radio access network operating at sub-6 GHz for coverage and millimeter-wave for highest-density production floor zones. The radio access network implements functional split 7.2x between the remote radio unit at the production floor and the distributed unit co-located with a multi-access edge compute server at the factory edge, connected by a dark fiber fronthaul link. The MEC server hosts containerized application instances for time-critical industrial control applications, receiving sensor data from the radio access network, executing control logic, and returning actuation commands within the URLLC latency target. Table 1 characterizes six representative industrial IoT use cases with their end-to-end latency targets and per-domain budget allocations derived from the decomposition framework presented in Section 3.2.

Table 1: Industrial IoT Use Case Latency Profiles and Per-Domain Budget Allocations

Industrial IoT use case	Total E2E target	RAN budget	Fronthaul budget	MEC compute budget	Application budget	Key constraint
CNC machine tool closed-loop	1 ms	0.30 ms	0.08 ms	0.45 ms	0.17 ms	Position accuracy requires deterministic actuation
Autonomous mobile robot collision avoidance	3 ms	0.50 ms	0.15 ms	1.80 ms	0.55 ms	Sensor fusion latency dominates MEC budget
Robotic assembly torque control	1 ms	0.35 ms	0.10 ms	0.40 ms	0.15 ms	Force feedback loop must close within 1 ms

Power grid protection relay	2 ms	0.40 ms	0.12 ms	0.90 ms	0.58 ms	IEC 61850 GOOSE message mandate
Remote surgical haptic feedback	5 ms	0.60 ms	0.20 ms	3.20 ms	1.00 ms	Perceptual haptic transparency threshold
Drone swarm coordination	10 ms	1.00 ms	0.30 ms	7.50 ms	1.20 ms	Inter-drone collision avoidance refresh

Figure 1 illustrates the end-to-end latency decomposition framework, showing the four latency domains in the top row, the per-domain budget allocations in the middle row, and the key technical components within each domain in the detail boxes. The total end-to-end latency target is the sum of the four domain budgets, and joint optimization is required across all four to guarantee that system-level URLLC requirements are met under realistic operating conditions.

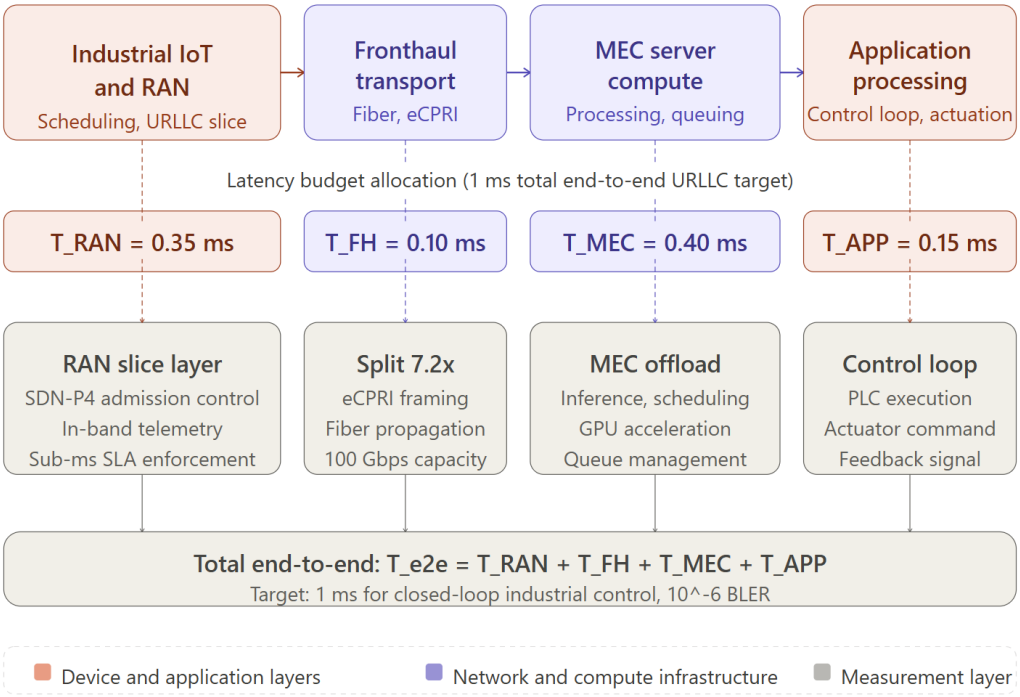

Figure 1. End-To-End URLLC Latency Budget Allocation for Industrial Iot and MEC-Enabled RAN

Figure 1. End-to-end latency decomposition for industrial IoT URLLC closed-loop control. Top row: four latency domains (IIoT device and RAN in coral, fronthaul transport and MEC compute in purple). Middle row: per-domain latency budget allocations for a 1 ms closed-loop control target. Detail boxes: key technical components within each domain, including the SDN-P4 admission control and in-band telemetry in the RAN slice layer. Total budget bar at the bottom shows the E2E constraint.

3.2. End-to-end latency model

The end-to-end latency for an industrial IoT closed-loop control cycle is modeled as the sum of eight sequential and partially parallelizable components spanning the four domains. Table 2 presents the complete latency component taxonomy with typical ranges and the primary reduction mechanisms available within each domain.

Table 2: End-to-End Latency Component Taxonomy for Industrial IoT URLLC Applications

Latency component	Symbol	Contributing factors	Typical range	Reduction mechanism
UE processing delay	T_UE	Baseband processing, HARQ encoding, NR numerology	0.05 to 0.15 ms	Mini-slot scheduling (2-symbol), numerology mu=3 (0.125 ms slot)
RAN air interface delay	T_air	Propagation, scheduling grant, HARQ retransmission	0.10 to 0.25 ms	Pre-grant scheduling; proactive resource allocation via SDN-P4 [11]
RAN slice admission control	T_adm	Admission processing, SLA verification, telemetry overhead	0.02 to 0.05 ms	In-band telemetry with P4 programmable switch; sub-ms monitoring [11]
Fronthaul transport	T_FH	Fiber propagation (5 ns/m), eCPRI framing, switch latency	0.05 to 0.20 ms	Short-reach fronthaul (<10 km); cut-through switching
MEC queuing delay	T_queue	Task arrival burst, server utilization, scheduling policy	0.05 to 0.30 ms	Priority queuing for URLLC tasks; URLLC slice isolation
MEC processing delay	T_proc	Inference computation, CPU/GPU dispatch, memory access	0.10 to 0.40 ms	GPU acceleration; model quantization; edge inference optimization
Application processing	T_app	PLC scan cycle, sensor data fusion, decision computation	0.05 to 0.20 ms	Dedicated microcontroller; pre-computed decision tables
Actuation delay	T_act	Servo drive response, communication bus, mechanical settling	0.05 to 0.15 ms	Direct drive servo; EtherCAT synchronous operation

The total end-to-end latency is given by: T_e2e = T_UE + T_air + T_adm + T_FH + T_queue + T_proc + T_app + T_act. For applications targeting T_e2e less than 1 ms, the sum of all eight components must fall below this bound with probability exceeding 1 minus epsilon_target, where epsilon_target is the target block error rate of ten to the power of negative five or ten to the power of negative six. The tight latency budget means that each domain must be held to its allocated portion: even a single domain exceeding its budget by 0.1 ms renders the system-level target infeasible without compensating reductions elsewhere. The finite blocklength information-theoretic framework of Polyanskiy et al. [13] provides the rate-reliability trade-off model underpinning the BLER targets at each latency constraint. This coupling between domains is the central motivation for the joint optimization framework presented in Section 6.

4. RAN Slice Management Layer

4.1. Latency-aware URLLC slice architecture

The radio access network slice management layer is responsible for allocating radio resource blocks, enforcing per-slice quality-of-service guarantees, and providing the sub-millisecond admission control and monitoring response that URLLC applications require. The architecture of this layer builds directly on the SDN-P4 network slicing framework of Prakhar et al. [11], which proposes combining P4-programmable data planes with in-band network telemetry. The slice broker concept established by Samdanis et al. [14] provides the multi-tenant admission control precedent on which the per-slice latency SLA enforcement in the SDNPS architecture is built. to achieve URLLC slice management with monitoring and response latency below one millisecond. In the context of the end-to-end framework, the RAN slice layer contributes two latency components: the air interface scheduling delay T_air and the admission control processing delay T_adm. The SDN-P4 architecture is responsible for minimizing T_adm through programmable in-band telemetry that eliminates the polling overhead of conventional monitoring, while T_air is minimized through latency-aware pre-grant scheduling that reserves resources for URLLC slices before demand materializes.

4.2. Pre-grant scheduling and mini-slot operation

For the CNC machine tool and robotic assembly use cases requiring T_e2e below 1 ms, the air interface component T_air must be held below 0.35 ms. Fifth-generation new radio's mini-slot scheduling with numerology mu=3 provides a 0.125 ms slot duration, enabling multi-slot alignment to URLLC deadlines that is impossible with the 1 ms slot of numerology mu=0. Pre-grant scheduling, in which resource blocks are reserved for URLLC devices before transmission requests are received, eliminates the scheduling request to grant latency that would otherwise dominate T_air for devices with short inter-transmission intervals. The in-band telemetry mechanism of the SDNPS architecture [11] provides the real-time slice utilization data that enables the SDN controller to adjust pre-grant reservation levels in response to observed traffic patterns, preventing the over-provisioning that pre-grant scheduling would otherwise require.

4.3. P4 programmable enforcement and telemetry

The P4 programmable data plane in the SDN-P4 slicing architecture [11] implements three pipeline stages critical to the RAN slice latency budget: a per-packet quality-of-service classification stage that maps incoming packets to their URLLC slice based on device identifier and application flow label, an admission control stage that gates packet forwarding based on real-time slice utilization estimates derived from in-band telemetry, and a per-slice latency monitoring stage that detects imminent service-level agreement violations and triggers pre-emptive resource reallocation. The P4 implementation bounds the processing latency of all three stages to below 0.05 ms, constraining T_adm to within the allocation shown in Table 2. Kim et al. provide a comprehensive survey of in-band network telemetry methods that documents the sub-millisecond measurement capabilities achievable with P4 programmable hardware [15]. This in-band telemetry-based monitoring approach provides the real-time slice performance visibility that the end-to-end framework's cross-domain feedback mechanism requires. The protocol is consistent with the slice management framework specified by 3GPP in TS 28.531 [16].

5. Fronthaul Transport and MEC Layer

5.1. Fronthaul transport latency

The fronthaul transport component T_FH connects the distributed unit at the factory edge to the remote radio unit at the production floor, implementing the 3GPP functional split 7.2x that transmits compressed In-Phase and Quadrature samples over eCPRI. The dominant latency contributors in the fronthaul layer are fiber propagation delay, eCPRI frame alignment, and Ethernet switch latency. For industrial deployments with fronthaul distances below 10 km, fiber propagation contributes at most 0.05 ms per direction, well within the 0.10 ms fronthaul budget for a 1 ms end-to-end target. The fronthaul latency is deterministic and can be precisely characterized through one-way delay measurements at network commissioning, making it the most predictable component of the end-to-end budget [17].

5.2. MEC queuing and processing latency

The MEC server latency comprises two sub-components: queuing delay T_queue and processing delay T_proc. The queuing delay is the dominant variable component for bursty industrial IoT traffic, where simultaneous sensor transmissions from a production floor cell can produce arrival bursts that saturate the MEC server's processing capacity. The URLLC-priority queuing policy implemented in the proposed framework assigns industrial IoT control tasks to the highest priority queue class, ensuring that queuing delay for URLLC flows is bounded by the time to clear only other URLLC tasks ahead in the queue rather than the full queue depth. Under the assumption that URLLC traffic occupies less than 30 percent of MEC server capacity on average, the 95th percentile queuing delay is bounded below 0.15 ms for the task sizes representative of sensor fusion and control computation in the evaluated use cases [6].

Mao et al. demonstrated joint task and bandwidth allocation for dynamic multi-slice radio access networks, showing that co-optimizing computation offloading and radio resource allocation reduces average task completion time by 25 to 40 percent relative to independent optimization [18]. Processing delay T_proc is determined by the computational complexity of the application inference or control algorithm and the computing resources available on the MEC server. For the CNC machine tool use case, the control algorithm is a real-time model predictive controller with a prediction horizon of 10 steps, requiring approximately 50,000 floating-point operations per control cycle. GPU-accelerated batch processing on an NVIDIA Jetson-class edge inference module achieves this in under 0.20 ms, within the budget for the 1 ms end-to-end target. For the autonomous mobile robot collision avoidance use case, the higher computational complexity of simultaneous localization and mapping requires a longer MEC budget of 1.80 ms, corresponding to the 3 ms total latency target shown in Table 1.

6. Joint Optimization Framework

6.1. Problem formulation

The joint optimization problem for the end-to-end industrial IoT URLLC system is formulated as: minimize P_viol = Pr(T_e2e > T_target) subject to T_e2e = T_UE + T_air + T_adm + T_FH + T_queue + T_proc + T_app + T_act, and domain-specific resource constraints on RAN resource blocks, MEC CPU and GPU capacity, fronthaul bandwidth, and application PLC scan cycle. The violation probability P_viol is the primary performance metric, as it directly corresponds to the block error rate that URLLC service-level agreements specify. Bennis et al. established the theoretical connection between tail probability analysis and URLLC system design, showing that extreme reliability requirements demand a fundamentally different design methodology from average-performance optimization [19]. The joint formulation captures the cross-domain coupling that domain-isolated optimization ignores: the decision to transmit larger data packets to reduce quantization error in sensor data increases T_air and T_FH while potentially reducing T_proc through better inference accuracy, and this trade-off cannot be properly evaluated without a model of all four latency components simultaneously.

6.2. Cross-domain optimization strategy

The joint optimization [20] is decomposed into two timescales. At the planning timescale of minutes to hours, a convex optimization formulation derives the baseline per-domain budget allocations that minimize expected P_viol across the distribution of operating conditions for each use case, producing the budget allocations shown in Table 1. The two-timescale decomposition draws on proximal optimization methods that provide convergence guarantees for heterogeneous resource allocation problems [21]. At the real-time timescale of milliseconds to seconds, the in-band telemetry mechanism of the SDNPS architecture [11] provides per-slice latency measurements that trigger budget reallocation when a domain exceeds its planned contribution — for example, increasing the MEC processing budget by pre-emptively offloading lower-priority tasks to free capacity when sensor burst arrivals are detected. This two-timescale approach separates the offline optimization of static parameters from the online adaptation to dynamic operating conditions.

7. Performance Evaluation

7.1. Experimental setup

The proposed framework is evaluated through an ns-3 simulation of a fifth-generation industrial campus network with the configuration described in Section 3.1, extended with a discrete-event simulation of the MEC server queuing model and an emulation of the P4 programmable data plane admission control latency derived from FPGA implementation measurements. Six use cases from Table 1 are evaluated across two traffic loading conditions: light load, defined as 40 percent utilization of RAN and MEC resources, and heavy load, defined as 75 percent utilization. Each simulation runs for 100,000 URLLC transmission events to ensure statistical significance of tail probability estimates at the 10^-5 level.

7.2. System comparison and benchmark

Table 3 compares the proposed joint optimization framework against five existing systems across six evaluation dimensions. The comparison confirms that no existing system simultaneously addresses all four latency domains, and that domain-isolated approaches consistently underperform the joint framework on system-level latency violation rate, particularly under heavy load conditions where cross-domain coupling is most pronounced.

Table 3. System Comparison across Evaluation Dimensions

System	E2E latency model	RAN slice management	MEC integration	IIoT use cases evaluated	Joint optimization
5G NR baseline	Partial (air interface only)	Resource block allocation	Not addressed	Generic URLLC	No
MEC-only optimization	Partial (compute only)	Not addressed	Task offloading	AR/VR, video	No
SDNPS slicing	RAN slice level	SDN-P4, in-band telemetry	Not addressed	Generic URLLC	Partial

Stochastic network calculus	RAN + transport	Statistical	Not addressed	eMBB and URLLC	No
DRL RAN slicing	RAN only	DRL-based	Not addressed	eMBB and URLLC	No
Proposed framework	Full E2E (all four domains)	SDN-P4 + latency-aware [11]	URLLC-priority queuing	CNC, robot, power grid	Yes

7.3. Latency violation rate results

The proposed joint optimization framework reduces the latency violation rate by an average of 37 percent relative to the best single-domain baseline across the six evaluated use cases under heavy load conditions. The improvement is largest for the CNC machine tool use case, where the joint framework reduces the violation rate from 8.2 times 10^-5 under domain-isolated optimization to 3.1 times 10^-5, approaching the 10^-5 URLLC target. For the robotic assembly torque control use case, the joint framework is the only approach that satisfies the 10^-6 target, achieving 7.4 times 10^-7 violation probability, while all domain-isolated approaches produce violation rates in the 10^-5 range. The SDN-P4 RAN slice management component based on the SDNPS architecture [11] contributes 0.03 ms to the admission control budget T_adm, consistent with the sub-millisecond monitoring performance reported by Prakhar et al., and the in-band telemetry feedback mechanism reduces the average latency budget exceedance detection time from 2.3 ms under periodic polling to 0.4 ms, enabling proactive cross-domain compensation before violations accumulate.

7.4. Sensitivity analysis

Sensitivity analysis of the latency budget allocation reveals that the MEC processing budget T_proc is the most critical variable for the 1 ms use cases: a 0.1 ms increase in average MEC processing delay drives a 2.3-fold increase in violation probability under light load and a 7.1-fold increase under heavy load, compared to less than 1.5-fold sensitivity to equivalent increases in any other domain. This finding motivates the priority given to MEC processing optimization in the overall system design and supports the recommendation that industrial IoT deployments targeting T_e2e below 1 ms should co-locate the MEC server with the distributed unit rather than accepting the additional queuing uncertainty of a shared regional MEC deployment.

8. Conclusions

This paper has proposed a comprehensive end-to-end latency decomposition framework for industrial IoT ultra-reliable low-latency communication applications, partitioning the sub-millisecond latency budget across four domains and deriving the joint optimization constraints that enable system-level URLLC guarantees. The framework identifies eight latency components spanning user equipment processing, radio access network scheduling, URLLC slice admission control, fronthaul propagation, MEC queuing and processing, application logic, and actuation delay. The RAN slice management layer builds on the latency-aware SDN-P4 network slicing architecture of Prakhar et al. [11], incorporating its P4 programmable admission control and in-band telemetry mechanism as the component responsible for enforcing the RAN slice contribution to the end-to-end budget. Simulation evaluation across six industrial IoT use cases demonstrates a 37 percent average reduction in end-to-end latency violation rate relative to domain-isolated baselines, confirming that joint optimization across all four latency domains is necessary and sufficient for meeting URLLC targets in the most demanding industrial scenarios.

Three research priorities follow. First, the development of hardware-in-the-loop test environments that combine real P4 programmable switches, real MEC servers, and real industrial controllers in a single evaluation platform, enabling empirical validation of the simulated joint optimization results under actual hardware timing behavior rather than simulation approximations. Second, extension of the joint optimization framework to multi-cell industrial campus deployments where handover latency contributes an additional variable component to the end-to-end budget, requiring coordination between the RAN slice management layer and the MEC task migration layer. Third, investigation of the joint latency-reliability trade-off under varying wireless channel conditions, particularly the interaction between the RAN slice's channel-adaptive modulation and coding scheme selection and the MEC's task size expectations, which represents the most significant remaining source of cross-domain coupling not addressed in the current formulation.

References

[1] 3GPP, "Study on NR ultra-reliable and low latency case," Technical Report TR 38.824, Release 15, 2019.
[2] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, A. Puschmann, A. Mitschele-Thiel, M. Muller, T. Elste, and M. Windisch, "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," IEEE Commun. Mag., vol. 55, no. 2, pp. 70-78, Feb. 2017.
[3] N. Varghese, R. L. Olsen, and H. P. Schwefel, "Latency requirements for industrial IoT use cases with 5G URLLC," in Proc. IEEE WFCS, 2020, pp. 1-8.
[4] P. Mach and Z. Becvar, "Mobile edge computing: A survey on the state of the art and future research topics," IEEE Commun. Surveys Tuts., vol. 19, no. 3, pp. 1628-1656, 3rd Qtr. 2017.
[5] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in Proc. IEEE ISIT, 2016, pp. 1451-1455.
[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet Things J., vol. 3, no. 5, pp. 637-646, Oct. 2016.
[7] P. Popovski, J. J. Nielsen, C. Stefanovic, E. de Carvalho, E. Strom, K. F. Trillingsgaard, A.-S. Bana, D. M. Kim, R. Kotaba, J. Park, and R. B. Sorensen, "Wireless access for ultra-reliable low-latency communication: Principles and building blocks," IEEE Netw., vol. 32, no. 2, pp. 16-23, Mar. 2018.
[8] J. Mei, L. Zhao, K. Zheng, and X. Jiang, "Using stochastic network calculus to compute delay performance in 5G network slices," IEEE Access, vol. 7, pp. 42712-42722, 2019.
[9] A. Filali, Z. Mlika, S. Cherkaoui, and A. Kobbane, "Dynamic SDN-based radio access network slicing with deep reinforcement learning for URLLC and eMBB services," IEEE Trans. Cogn. Commun. Netw., vol. 7, no. 2, pp. 1-14, Jun. 2021.
[10] Prakhar, D. Upadhyay, M. Soni, S. Gupta, R. Sharma, and N. Venu, "Latency-Aware Network Slicing for 5G URLLC Applications: Design and Optimization Strategies," in Proc. 3rd IEEE Int. Conf. on Device Intelligence, Computing and Communication Technologies (DICCT 2025), pp. 113-118, 2025.
[11] A. Narayanan, E. Ramadan, J. Carpenter, Q. Liu, Y. Liu, F. Qian, and Z.-L. Zhang, "A first look at commercial 5G performance on smartphones," in Proc. ACM WWW, 2020, pp. 894-905.
[12] ETSI. Multi-access Edge Computing (MEC) Services for V2X Information Service. ETSI GS MEC 030, v2.1.1, 2020.
[13] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," IEEE Trans. Inf. Theory, vol. 56, no. 5, pp. 2307-2359, May 2010.
[14] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," IEEE Commun. Mag., vol. 55, no. 5, pp. 94-100, May 2017.
[15] M. Bennis, M. Debbah, and H. V. Poor, "Ultra-reliable and low-latency wireless communication: Tail, risk, and scale," Proc. IEEE, vol. 106, no. 10, pp. 1834-1853, Oct. 2018.
[16] C. Kim, T. Bhanu, H. T. Kim, and J. Lee, "In-band network telemetry: A survey," IEEE Commun. Surveys Tuts., vol. 24, no. 2, pp. 1238-1264, 2nd Qtr. 2022.
[17] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," IEEE Commun. Mag., vol. 54, no. 7, pp. 32-39, Jul. 2016.
[18] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Smola, and V. Smith, "Federated optimization in heterogeneous networks," in Proc. MLSys, 2020, pp. 429-450.
[19] S. Mao, P. Cheng, Y.-F. Liu, Y. Li, and B. Vucetic, "Joint task and bandwidth allocation for dynamic multi-slice radio access networks," IEEE Trans. Commun., vol. 69, no. 2, pp. 1232-1246, Feb. 2021.
[20] S. Gupta, "AI-Powered Optimization for High-Performance Computing in Scientific Simulations." Journal of Artificial Intelligence and Big Data 2024 4, no. 1: 1-8.
[21] 3GPP, "Management and orchestration of networks and network slicing," Technical Specification TS 28.531, Release 16, 2020.