TSNBench: Benchmarking LLM Proficiency in Time-Sensitive Networking

Rubi Debnath[1*] **Daniel Bujosa**[2] **Luxi Zhao**[3]
Silviu S. Craciunas[2,4] **Paul Pop**[2] **Sebastian Steinhorst**[1]
[1]Technical University of Munich, Germany [2]Technical University of Denmark, Denmark
[3]Beihang University, China [4]NXP Semiconductors, Austria

Abstract

We present TSNBench, the first benchmark for evaluating large language model (LLM) proficiency in Time-Sensitive Networking (TSN), a suite of IEEE 802.1 standards for deterministic communication with bounded latency in safety-critical domains such as autonomous vehicles, aviation, defense, and industrial automation. While LLMs have been extensively evaluated on general knowledge tasks, their capabilities in safety-critical networking domains remain largely unexplored. TSNBench comprises 939 expert-validated multiple-choice questions (MCQs) covering diverse TSN mechanisms, along with 100 open-ended Worst-Case Delay (WCD) computation tasks for each of Credit-Based Shaper (CBS) and Cyclic Queuing and Forwarding (CQF), across varying topologies and traffic flows. The ground-truth (GT) WCD values are computed using a verified Network Calculus (NC) solver for CBS and closed-form upper bounds for CQF. We evaluate 16 LLMs and find that although models achieve 67 to 95% accuracy on MCQs, they fail substantially on open-ended WCD computation. For CBS, the best-performing model, GPT-5, deviates from the GT by 150.2 μs on average (mean absolute error, MAE), and its predicted WCD is within 50 μs of the GT for only 39.4% of flows, while all other models exceed an MAE of 250 μs. Providing the WCD equations in the prompt lowers the error for 9 of 10 models on CQF but only 2 of 9 on CBS. Such errors are large relative to TSN latency budgets and can lead to violations of real-time constraints and unsafe configurations. TSNBench demonstrates that MCQ benchmarks may overestimate LLM capabilities in safety-critical networking.

1 Introduction

Recent advances in large language models (LLMs) across different domains such as engineering [Jackson et al., 2025, Guo et al., 2025], medicine [Xie et al., 2025, Liu et al., 2023, Li et al., 2024], clinical practice [Kweon et al., 2024], computer networking [Sharma and Yegneswaran, 2023], telecommunications [Maatouk et al., 2026, Ferrag et al., 2026, Oluwaseyi et al., 2025, Gajjar et al., 2025], and automation [Shen et al., 2024] have shown groundbreaking performance in assisting engineers, practitioners, researchers [Huang et al., 2023, Sun et al., 2024], and doctors in solving real-world problems. System engineers are increasingly using LLMs to design and configure networks [Wang et al., 2024a], generate code, and analyze network logs. With this, they are entering new territory: safety-critical application domains such as autonomous vehicles, aerospace [Fiori et al., 2024, Sanchez-Garrido et al., 2021], defense [Elliott, 2024], and industrial communication [Zhang et al., 2024]. In these contexts, the accuracy, reliability, and consistency of LLMs become far more than leaderboard metrics, as they become engineering requirements.

*Corresponding author. Email: rubi.debnath@tum.de

40th Conference on Neural Information Processing Systems (NeurIPS 2026). Track on Evaluations and Datasets.

Time-Sensitive Networking (TSN) [IEEE, 2018], standardized by the IEEE 802.1 Working Group (WG), is a layer-2 Ethernet technology that provides deterministic communication guarantees for safety-critical applications. TSN deployments typically separate traffic based on timing criticality. Safety-critical periodic communication with guaranteed latency and bounded jitter is categorized as time-triggered (TT) traffic and is served using the IEEE 802.1Qbv timed-gate mechanism. TT transmissions are controlled by a Gate-Control List (GCL), computed offline using exact methods such as SMT-based synthesis [Craciunas et al., 2016] or heuristic approaches [Pop et al., 2016, Gavriluţ et al., 2018, Bujosa et al., 2022]. In contrast, periodic or sporadic communication requiring bounded end-to-end latency but less stringent jitter control is classified as Audio Video Bridging (AVB) stream traffic [Böhm and Wermser, 2021, Bruckner et al., 2019]. Consequently, Worst-Case Delay (WCD) estimation errors of tens or hundreds of microseconds are significant, as they can consume timing margins, violate deadlines, or lead to infeasible TSN configurations. In mission-critical deployments, such errors can have severe consequences. A misconfigured TSN network can cause, for example, a robotic arm to miss a critical assembly step, a brake system to fail on a highway, an aircraft control system to respond incorrectly, a defense mechanism to collapse, or a spacecraft to miss a vital signal. These failures may result from sub-millisecond timing violations caused by a single misconfiguration. These risks highlight the importance of accurate analysis and configuration in TSN systems, especially as LLMs are increasingly integrated into network management workflows. Therefore, their domain proficiency must be rigorously evaluated. However, to the best of our knowledge, no existing benchmark evaluates LLM proficiency in TSN.

To fill this gap, we introduce TSNBench, the first benchmark for evaluating LLM proficiency in TSN, comprising two complementary evaluation components. The first is a 939-question expert-validated multiple-choice question and answer (MCQA) dataset, generated from 83 peer-reviewed research papers using three LLMs from distinct model families and rigorously reviewed by five domain experts, each with over eight years of TSN research experience. The second is a set of open-ended questions requiring multi-step WCD computation for two widely deployed TSN mechanisms, namely Credit-Based Shaper (CBS) [IEEE, 2010] and Cyclic Queuing and Forwarding (CQF) [IEEE, 2017, Yan et al., 2020], across varying network topologies and traffic flows, with ground truth (GT) computed using a verified Network Calculus (NC) solver [Zhao et al., 2018] for CBS and closed-form mathematical upper bounds for CQF [Wang et al., 2023]. These open-ended WCD questions are intended as a stress test of standalone model capability, not as a deployment workflow for free-text LLM timing outputs. Background on TSN, NC, CBS, and CQF is provided in Appendices B to E.

While general-purpose benchmarks such as MMLU [Hendrycks et al., 2021] and MMLU-Pro [Wang et al., 2024b] evaluate broad subject knowledge spanning elementary mathematics, history, and law, they are fundamentally unsuited for safety-critical domain-specific evaluation. Answering an MCQ about elementary school history is categorically different from answering TSN terminology questions and correctly computing a WCD under NC constraints for a given network topology. Without a benchmark that captures this distinction, there is no principled way to measure LLM progress in deterministic networking domains. TSNBench is designed precisely to expose this gap.

We evaluate 16 LLMs comprising open-source and closed-source models, as well as general-purpose and reasoning-specialized architectures. Our results reveal a striking dissociation, where models achieve 67 to 95% accuracy on MCQA yet fail substantially on open-ended WCD computation. The best-performing model on CBS, GPT-5, still deviates from the ground-truth WCD by 150.2 μs on average (mean absolute error, MAE), while all other models exceed 250 μs. This is concerning in a domain where timing violations of tens of microseconds, even 1% of a 1000 μs deadline, may cause system failures. The dataset (CC-BY 4.0) is available at `https://doi.org/10.34740/kaggle/ds/13019279` and the code (MIT) at `https://github.com/tum-esi/TSNBench`.

Our key contributions are:

1. **First expert-validated TSN benchmark:** TSNBench evaluates LLM knowledge of TSN mechanisms through 939 expert-validated MCQs derived from peer-reviewed TSN literature.
2. **Open-ended timing-analysis tasks:** TSNBench includes open-ended WCD computation tasks for CBS and CQF with GT computed using a verified NC solver for CBS and closed-form mathematical bounds for CQF.
3. **Evaluation across 16 LLMs:** We evaluate both open-source and closed-source models, including general-purpose and reasoning-specialized models, and show that high MCQA accuracy does not reliably predict accurate WCD computation.

4. **Analysis beyond closed-book prompting:** We test whether providing the WCD equations in the prompt (formula-in-context) lowers the error, how sensitive the error is to prompt wording and output format, and whether the findings hold on a real-world topology.

In summary, TSNBench gives the real-time networking and machine learning communities the first rigorous resource for evaluating LLM proficiency in TSN and the limits of LLMs in safety-critical, computationally demanding domains.

2 Related Work

General LLM Benchmarks: Benchmarking and datasets are essential for measuring LLM progress and identifying key gaps and limitations [Hendrycks et al., 2021, Wang et al., 2024b]. General knowledge benchmarks such as MMLU [Hendrycks et al., 2021] and MMLU-Pro [Wang et al., 2024b] evaluate broad subject knowledge including elementary mathematics, history, computer science, and law, using multiple-choice questions. Domain-specific benchmarks have extended this paradigm to medicine [Xie et al., 2025, Liu et al., 2023, Li et al., 2024], clinical practice [Kweon et al., 2024], law [Guha et al., 2023], code generation [Hua et al., 2025, Huang et al., 2024], and scientific research [Sun et al., 2024]. While these benchmarks have driven significant progress, they are not designed to evaluate safety-critical networking tasks. Most rely on multiple-choice evaluation, and none assess whether a model can perform the multi-step computational reasoning required in safety-critical networking domains. TSNBench addresses this gap by introducing MCQA and open-ended WCD computation questions with GT computed by a verified state-of-the-art NC solver and closed-form bounds, providing an evaluation of TSN that no existing general benchmark captures.

Networking and Telecommunications Benchmarks: In the last few years, several benchmarks have evaluated LLM proficiency in networking and telecommunications domains. TeleQnA [Maatouk et al., 2026] presents an MCQ dataset for telecommunications, generated from research documents and 3GPP standards and validated by domain experts. 6G-Bench [Ferrag et al., 2026] presents an MCQ-based dataset for 6G networks containing 3,722 difficult questions validated through automated filtering and expert human review. Beyond question-answering benchmarks, NetConfEval [Wang et al., 2024a] evaluates LLMs on network configuration tasks and demonstrates that LLMs can simplify and automate complex network management tasks.

LLMs for TSN and Real-Time Networks: The application of LLMs to TSN management and orchestration is still at a very early stage, with only limited initial studies available. Windmann et al. [2025] explored the use of LLMs for configuring hybrid 5G/TSN networks by assisting users with manual configuration tasks and suggesting configurations in a 5G-TSN network. However, this work remains preliminary and does not provide experimental results. Overall, prior work does not provide a systematic benchmark or rigorous evaluation of LLM proficiency across TSN mechanisms, nor does it assess computational reasoning capabilities for WCD analysis. TSNBench fills this gap by providing the first structured benchmark covering both declarative TSN knowledge through MCQA and computational reasoning through open-ended WCD evaluation.

3 TSNBench

Unlike established domains such as medicine [Xie et al., 2025], 5G [Oluwaseyi et al., 2025, Maatouk et al., 2026], general human knowledge [Phan et al., 2026, Hendrycks et al., 2021, Wang et al., 2024b], coding [Hua et al., 2025, Huang et al., 2024], and law [Guha et al., 2023], no open-source TSN dataset exists for LLM evaluation [Zhang et al., 2024, Peng et al., 2023, Zanbouri et al., 2025, Adil et al., 2026]. As highlighted in [Liu et al., 2023], the data source determines the reliability of a dataset, and generating a high-quality dataset is a crucial prerequisite for meaningful benchmarking. We describe the TSNBench construction pipeline below, with full details provided in Appendix F.

3.1 Dataset Source Selection

Published research papers and standards are among the most reliable sources for building domain-specific datasets [Liu et al., 2023]. Since TSN knowledge originates primarily from peer-reviewed research and IEEE 802.1 TSN standards, we curate a collection of open-access research documents

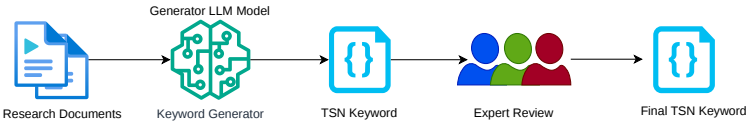


Figure 1: TSNBench keyword-generation pipeline. TSN keywords are extracted from research documents using an LLM, expert-verified, and used for MCQA generation as described in Section 3.3.

as our source corpus. To avoid copyright issues and exclude papers with incorrect results or flawed methodologies, we include only published or accepted papers that are openly available. For papers not available in open-access form, we use arXiv versions of papers that have been published or accepted, excluding unpublished preprints with unverified results. Where possible, we also collect author manuscript versions with proper attribution. To ensure quality, we prioritize highly cited papers from reputable venues while accounting for publication timeline, as recent papers naturally have fewer citations. In total, we collect 83 research papers covering a broad range of TSN mechanisms, including Time-Aware Shaper (TAS), CBS, CQF, NC-based schedulability analysis, performance evaluation, hardware experiments, combined shapers such as TAS+CBS [Zhao et al., 2022], and Multi-CQF [Alexandris et al., 2022].

3.2 Keyword and Acronym Extraction

TSN employs specialized vocabulary, similar to other communication domains [Andrews et al., 2014, Saad et al., 2020, Ma et al., 2019]. A successful LLM that understands TSN should be able to reason correctly about TSN terminology. A model that cannot differentiate between TAS and CBS, or cannot correctly expand TSN-specific acronyms, cannot be considered proficient in TSN. To capture this dimension, we extract keywords and acronyms widely used in TSN literature and use them to guide MCQA generation. All terms are extracted from the 83 research documents using Claude Sonnet 4, as shown in Table 6, and stored in JSON format. Each document is preprocessed to remove non-relevant content, including author names, affiliations, figures, tables, URLs, and pseudocode. The model is instructed to extract only terms defined within the document, without relying on pretrained knowledge, and to provide each term's acronym, full form, and one-to-two-sentence definition from the source. The extracted set is then reviewed by domain experts to resolve duplicates, retaining the longer definition in cases of conflict. Figure 1 illustrates this pipeline.

3.3 MCQA Generation, Post-Processing, and Expert Review

Raw MCQA Generation: To optimize time and reduce manual effort, we use an LLM-based approach to generate MCQAs from research documents. The keyword file is provided alongside the research documents as additional input, serving as an independent source to complement research paper content during generation. We use three models from distinct families, namely Claude Sonnet 4, GPT-4o mini, and Llama 3.1 70B, as shown in Table 6 (in Appendix F). These models are deliberately selected to ensure diverse styles and reasoning capabilities, thereby reducing generative bias. The same system prompt is used for all models, and each research paper is assigned to exactly one model in a round-robin manner.

Post-Processing: LLM-generated MCQAs cannot be used directly for benchmarking, as they may contain incorrectly formulated questions, incomplete options, or vague and incorrect answer choices. To address positional bias introduced by the generating model, answer options are shuffled randomly prior to human expert review, with the correct answer label updated to reflect the new ordering.

Human-Based Domain Expert Review: Given the safety-critical nature of TSN, rigorous human validation is essential. We engage five TSN domain experts: three senior professors with more than 15 years of research experience and two postdoctoral researchers with more than 8 years of expertise. Each question is independently evaluated with four outcomes: (i) *accept* - correct and clear; (ii) *revise* - requires modification for clarity or correctness; (iii) *reject* - the question is incorrect, misleading, or irrelevant; or (iv) *doubtful* - the expert is uncertain and passes it to remaining reviewers for consensus. Questions without consensus are discarded. Full review criteria are provided in Appendix F.3.

Mechanism Coverage: To check for under-representation, we categorize all 939 MCQs by TSN mechanism (Table 7 in the Appendix). TAS accounts for the largest share of the questions (36.1%), reflecting its larger research base. Each TSN mechanism accounts for at least 5% of the questions. Table 1 summarizes the dataset statistics and Figure 2 illustrates the full pipeline.

Table 1: TSNBench dataset construction statistics. Full generation details are in Appendix F.

Type	Category	Count
MCQA	Total raw questions generated by models	1326
	Questions removed after expert review	387
	Questions revised by domain experts	185
	Questions in the final dataset (used for benchmarking)	**939**
Open-ended questions	Credit-Based Shaper (CBS)	100
	Cyclic Queuing and Forwarding (CQF)	100
Orion Crew Exploration Vehicle (CEV)	CBS	10
	CQF	10

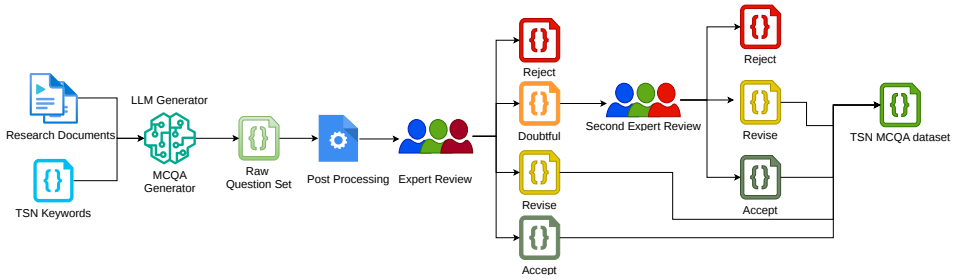


Figure 2: Pipeline of our TSNBench MCQA dataset generator, showing all steps from raw generation to the final validated dataset.

3.4 Open-Ended Question Formulation

While MCQA evaluates declarative TSN knowledge, open-ended questions assess whether LLMs can perform the multi-step mathematical reasoning required in real TSN deployment. We evaluate WCD computation, as WCD is a central key performance indicator (KPI) in TSN network design and directly determines whether a network meets its stringent timing requirements. We select two TSN mechanisms for this evaluation: CBS and CQF. CBS is widely deployed for audio-video traffic and requires NC-based analysis, making it mathematically demanding. CQF is a more recently standardized TSN mechanism whose WCD can be computed from a closed-form equation given routing and cycle duration (T), providing a complementary evaluation that isolates formula application from NC complexity. Together, these two mechanisms span a meaningful range of WCD computation difficulty. GT WCD values are computed using a verified state-of-the-art NC tool [Zhao et al., 2018] for CBS and closed-form mathematical upper bounds for CQF. We release all GT WCD values alongside the questions to support future open-source community evaluations. Each open-ended question is formulated by domain experts, as shown in Figure 3, and comprises three components: network topology, flow information, and flow routing. In TSNBench, three topologies are used to cover a broad range of scenarios: (i) one-switch topology (Figure 12), (ii) medium-mesh topology (Figure 13), and (iii) ring topology, representing industrial networks (Figure 14). We also evaluate a real-world topology (Figure 15). Each topology consists of end nodes and switches connected via Ethernet links, with unicast traffic flows transmitted from a sender to a single receiver. Flows consist of Ethernet frames whose maximum payload is bounded by the Maximum Transmission Unit (MTU). Further topology, flow, and routing details are provided in Appendix F.6.

3.5 Prompt Design

For both MCQA and open-ended evaluations, each prompt defines the model's role as a TSN expert. For MCQA, we use zero-shot prompting with no in-context examples, representing a conservative

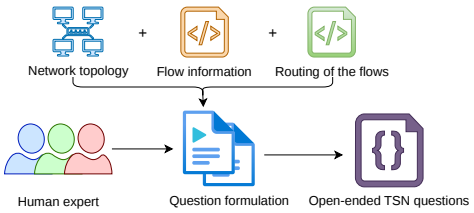


Figure 3: Pipeline for TSNBench open-ended question formulation by domain experts. Each question comprises three components: network topology, flow information, and flow routing.

approach that measures inherent TSN proficiency, ensuring that the output performance reflects the model's domain knowledge rather than in-context pattern matching. For both question types, the model is asked to provide a confidence score alongside its answer.

For open-ended questions, we use two settings: (i) closed-book (CB), and (ii) formula-in-context (FC). In the CB setting, we use a zero-shot prompt, providing no example WCD calculations or NC or CQF equations, ensuring the model independently recalls and applies the correct computational methodology. In the FC setting, the prompt additionally provides the NC equations for CBS WCD analysis and the upper-bound equations of CQF WCD calculation, to test whether the errors persist when the model does not have to recall the methodology. The open-ended prompt comprises three variable components: network topology, flow parameters, and pre-computed shortest-path routes. The same prompt template is used across all 100 open-ended evaluation instances per mechanism and setting, with only these three components varying. Fixed network constants are maintained throughout to ensure comparability across models and instances. To measure the sensitivity to the prompt itself, the CB prompt is additionally evaluated in two variants: Variant A paraphrases the instructions, and Variant B requests the answer as plain text instead of JSON. A detailed discussion of the open-ended prompt design is provided in Appendix F.5.

3.6 Model Scoring and Ground Truth

For the MCQA dataset, performance is measured as the percentage of questions answered correctly, reported as accuracy. For the open-ended questions, we evaluate the computational reasoning capability of each model by comparing its predicted WCD values against GT values. For CBS, GT WCD values are derived using NC-based Total Flow Analysis (TFA). Specifically, the worst-case delay upper bound D_f^h for flow $f \in \mathcal{F}_{M_i}^h$ at h equals the worst-case delay upper bound $D_{M_i}^h$ for all flows with the same priority M_i aggregating at the output port h,

$$D_f^h = D_{M_i}^h = hDev(\alpha_{M_i}^h, \beta_{M_i}^h) = \sup_{t \geq 0} \left\{ \inf \left\{ \tau \geq 0 \mid \alpha_{M_i}^h(t) \leq \beta_{M_i}^h(t+\tau) \right\} \right\}, \tag{1}$$

where $\alpha_{M_i}^h(t)$ represents the arrival curve of aggregate flows of priority M_i passing through h, and $\beta_{M_i}^h(t)$ represents the service curve for these corresponding flows. The end-to-end WCD for a flow is obtained by summing per-port delay bounds along its route. Full NC methodology and proofs are provided in Appendix C.

For CQF, the worst-case end-to-end delay is given by the closed-form expression

$$\mathrm{WCD} = f_i.\phi + (\mathrm{SW}_{\mathrm{num}} + 1) \cdot \mathrm{T} + \xi, \tag{2}$$

where $f_i.\phi$ is the flow offset at the source node in μs, $\mathrm{SW}_{\mathrm{num}}$ is the number of switches along the flow route, T is the cycle duration in μs, and ξ denotes the network-specific delays including propagation delay, switching delay (per switch), and time synchronization error. Details and a description of this bound are provided in Appendix E.

4 Experiments

We evaluate 16 state-of-the-art LLMs spanning open-source and closed-source models across general-purpose and reasoning-specialized architectures. Table 9 in Appendix G provides the full list of models with their model IDs and organizations. All models are accessed via their vendor APIs with

Table 2: TSNBench MCQA results across 16 models. Accuracy is the percentage of correct answers out of 939 questions, and consistency measures whether the model gives the same response across three runs. All models are evaluated at temperature 0.0 for deterministic performance. Models without temperature support use their default setting and are marked with †. Full model details are given in Table 9, and extended results with temperature comparisons are provided in Table 10 in Appendix G.

Model	Accuracy (%)	Avg. Consistency	Avg. Latency (ms)	Avg. Conf.	ECE↓	Brier↓	CW Rate↓
Grok 4.1 Fast†	93.2	0.9858	6673	0.9509	0.0198	0.0599	99.0
Grok 4.1 Fast (Non-Reasoning)	91.7	0.9986	515	0.9760	0.0589	0.0764	100.0
DeepSeek-V3.2 (Non-Thinking)	94.0	0.9993	804	0.9312	**0.0104**	0.0526	96.4
GPT-4o	91.8	0.9957	729	0.8782	0.0384	0.0765	99.2
GPT-4o mini	88.3	0.9950	799	0.9004	0.0171	0.0974	77.8
Llama 3.3 70B	88.9	0.9950	365	0.9082	0.0346	0.0918	100.0
Mistral Medium 3.1	92.1	0.9965	653	**0.9779**	0.0574	0.0750	100.0
Mistral Large 3	92.8	0.9975	5498	0.9476	0.0194	0.0646	100.0
Claude Sonnet 4.5	**95.3**	0.9993	1842	0.9374	0.0178	**0.0429**	86.6
o3†	94.7	0.9840	3845	**0.7524**	**0.1951**	0.0852	**3.4**
GPT-5†	95.0	0.9886	5630	0.8773	0.0720	0.0475	51.7
DeepSeek-V3.2 (Thinking)†	94.7	0.9819	4400	0.9202	0.0240	0.0487	78.1
Gemini 2.5 Flash	90.1	0.9847	6744	0.9674	0.0793	0.0942	95.4
Llama 3.2 1B	**67.4**	1.0	669	0.8529	0.1840	**0.2544**	99.0
Qwen3 8B	83.7	0.9897	**15103**	0.8616	0.0360	0.1322	100.0
Ministral 3 8B	86.9	0.9954	**345**	0.9649	0.0971	0.1230	100.0

† Temperature parameter not supported. Evaluated with default settings. ↓ lower is better.

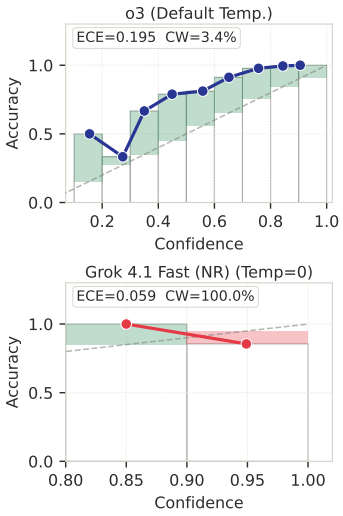


Figure 4: Reliability plot for o3 and Grok 4.1 Fast (NR). Full reliability analysis is in Figure 16.

no fine-tuning applied: GPT (OpenAI API), DeepSeek (DeepSeek API), Mistral (Mistral AI API), Claude (Anthropic API), Gemini (Google AI API), Grok (xAI API), and Llama and Qwen (Hugging Face inference router). All client-side operations, including prompt construction, API handling, response parsing, and metric computation, are performed on a standard workstation. To assess repeatability, each MCQ is evaluated three times at temperature 0.0 and three times at 0.7, and each open-ended question in up to three runs at 0.0. Models that do not expose a temperature parameter use the vendor default configuration, as noted in Table 2. Since TSN is widely used in safety-critical domains, deterministic responses are essential, as non-determinism would undermine the reliability of LLM-based TSN reasoning. Full cost and latency details are provided in Appendix G.2, Table 11.

4.1 MCQA Evaluation

Contamination Analysis: Since the MCQs were generated using models from families included in the evaluation, as shown in Table 6, contamination is a potential concern. We therefore separate the evaluated models into generator families (Claude, GPT, Llama) and non-generator families (all remaining models) and compare their average MCQA accuracy. Generator-family models achieve an average accuracy of 88.8%, whereas non-generator-family models achieve 91.0%. The generator-family models do not perform better than the non-generator-family models, so we do not observe evidence of a systematic advantage. This analysis does not rule out all possible contamination pathways, but it addresses this specific concern. The open-ended timing tasks are less likely to be affected because their topology, flow, and routing inputs were constructed specifically for TSNBench.

Evaluation Metrics: Model performance on the MCQA dataset is measured using accuracy, defined as the percentage of correctly answered questions out of 939, averaged across three runs. We additionally report Expected Calibration Error (ECE) [Pavlovic, 2025] and Brier score [Hoessly, 2026] to evaluate the alignment between the model's expressed confidence and its actual correctness. Calibration is particularly critical in safety-critical domains such as TSN, where high-confidence incorrect answers may lead to misleading configuration decisions, deadline violations, or network instability in industrial and automotive systems. We therefore also evaluate the Confidently Wrong (CW) rate to determine the fraction of incorrect answers where the model expresses high confidence ($\geq$0.8). All calibration metrics are computed on all 939 MCQs across three runs per model.

Results and Discussion: Table 2 reports accuracy, average (avg.) consistency, calibration, and average latency for all 16 models. The top performers are Claude Sonnet 4.5 (95.3%) and GPT-5 (95.0%), with Claude Sonnet 4.5 also achieving the lowest Brier score (0.0429), indicating strong accuracy and calibration. Llama 3.2 1B achieves the lowest accuracy (67.4%), consistent with its substantially smaller parameter count compared with the other models.

A notable finding emerges from the reasoning models. Despite their stronger general reasoning capabilities, o3, GPT-5, and DeepSeek-V3.2 (Thinking) do not outperform the best non-reasoning models on MCQA, all scoring below Claude Sonnet 4.5. This suggests that TSN MCQA performance is primarily driven by domain knowledge rather than general reasoning, and that reasoning-specialized architectures offer limited advantage on declarative knowledge retrieval tasks.

The calibration results reveal key differences across models. While most models are well-calibrated (ECE < 0.06), o3 has the highest ECE (0.1951) despite 94.7% accuracy, yet achieves the lowest CW rate (3.4%), rarely assigning high confidence to incorrect answers (refer to Figure 4). In contrast, many non-reasoning models have CW rates of 100%, assigning high confidence to incorrect answers. Mistral Medium 3.1 has the highest average confidence (0.9779) while maintaining 92.1% accuracy. All models have zero refusal rate, indicating that the MCQA dataset does not trigger response refusals.

4.2 Open-Ended Question Evaluation

Evaluation Metrics: For the open-ended questions, we report the MAE, computed per test case (TC). Each TC_x consists of n_x flows, where n_x is the number of flows in TC_x. Each flow is denoted as f_i, $i = 1, \ldots, n_x$. For each flow f_i, $\hat{y}_{TC_x,f_i}$ denotes the WCD predicted by the model for TC_x and y_{TC_x,f_i} denotes the GT WCD of flow f_i for TC number x, computed using a verified NC solver for CBS and using Eq. 2 for CQF. Each TC is evaluated in up to three runs at one temperature per model (0.0, or the vendor default for models without temperature support), and $\hat{y}_{TC_x,f_i}$ is the average of the predictions of flow f_i over the valid runs. A run is discarded if it returns no WCD values, only zeros, or a flow identifier that does not exist in the TC (e.g., f_0 to f_2 instead of f_1 to f_3). Non-numeric answers such as "unbounded" or "inf" and values above 10^9 μs are treated as missing. With P_x denoting the flows of TC_x that receive a predicted WCD, the MAE of a TC is defined as:

$$\text{MAE}_{TC_x} = \frac{1}{|P_x|} \sum_{f_i \in P_x} \left| \hat{y}_{TC_x,f_i} - y_{TC_x,f_i} \right|, \tag{3}$$

where $x \in \{1, \ldots, 100\}$ is the TC index. TCs in which fewer than 80% of the flows are predicted are excluded. The overall MAE of a model is the average over its set $\mathcal{V}$ of valid TCs ($|\mathcal{V}| \leq 100$):

$$\text{MAE} = \frac{1}{|\mathcal{V}|} \sum_{x \in \mathcal{V}} \text{MAE}_{TC_x}, \tag{4}$$

MAE is reported with its standard deviation across the valid TCs. A model is reported only if at least 50 valid TCs remain. We additionally report the median MAE across TCs as a robust measure against outlier TCs. We also report the percentage of flows whose absolute error falls below 50, 100, 500, and 1000 μs. We evaluate each model in the CB and FC settings and we report the relative change $\Delta = \left(\frac{\text{MAE}_{\text{CB}} - \text{MAE}_{\text{FC}}}{\text{MAE}_{\text{CB}}} \right) \cdot 100\,\%$, so a positive Δ means that providing the formula lowers the error. A worked example is given in Appendix G.4 (Table 12).

Results and Discussion: Table 3 presents the WCD computation results for both CBS and CQF across all 100 TCs under both CB and FC settings. The central finding is a striking dissociation between MCQA accuracy and computational reasoning performance. Models that achieve above 90% accuracy on MCQA still fail substantially on open-ended WCD computation, with the best-performing model, GPT-5, achieving an MAE of 150.2 μs on CBS, which is concerning because industrial TSN traffic can have strict timing requirements [Ekrad et al., 2025]. Medians are given in Tables 13 and 14, and per-TC results in Tables 20 to 23 in Appendix G.4.

For CBS, most models produce large errors, with all models except GPT-5 exceeding 250 μs MAE. Several models exhibit distinct failure modes. On CBS, Llama 3.2 1B returns WCD values for only 25 of its 61 evaluated TCs, all of them zero, and no parsable value for the others, so none of its TCs is valid. Grok 4.1 Fast returns truncated JSON, providing flow profile metadata but no WCD values, suggesting that the model hit an output length limit. DeepSeek-V3.2 (Thinking) returns empty responses for all CBS TCs and all but one CQF TC. The NC-based computation required for CBS is mathematically demanding and complex, and the zero-shot setting reveals that most models cannot independently recall or correctly apply the full NC methodology. Notably, the OpenAI reasoning models GPT-5 and o3 achieve the lowest CBS MAE of all models (150.2 and 255.8 μs), suggesting that multi-step mathematical reasoning capability provides an advantage for NC-based WCD computation even when it does not improve MCQA accuracy.

Table 3: Open-ended WCD estimation results for CBS and CQF across 100 test cases (TCs), reported as MAE ± standard deviation across valid TCs (↓ lower is better), under the closed-book (CB) and formula-in-context (FC) settings. Δ is the relative change of the MAE from CB to FC, with each MAE taken over the valid TCs of its setting. Positive values mean the formula lowers the error. Medians and full statistics are reported in Tables 13 and 14.

Model	CBS WCD MAE			CQF WCD MAE		
	CB (µs) ↓	FC (µs) ↓	Δ (%)	CB (µs) ↓	FC (µs) ↓	Δ (%)
Grok 4.1 Fast†,¶	–	–	–	130.1 ± 85.0	–	–
Grok 4.1 Fast (Non-Reasoning)¶	**3637.3 ± 3934.5**	–	–	163.9 ± 37.8	–	–
DeepSeek-V3.2 (Non-Thinking)¶	–	–	–	168.1 ± 66.3	–	–
GPT-4o⋆	–	–	–	82.2 ± 264.7	11.3 ± 15.8	+86.3
GPT-4o mini	378.5 ± 189.7	461.6 ± 999.5	−21.9	180.5 ± 82.0	**1377.1 ± 2818.7**	**−662.9**
Llama 3.3 70B	301.2 ± 171.0	333.5 ± 171.3	−10.7	173.3 ± 73.5	5.1 ± 15.0	+97.1
Mistral Medium 3.1	339.6 ± 224.9	374.2 ± 232.7	−10.2	150.3 ± 42.2	33.0 ± 31.3	+78.0
Mistral Large 3	270.6 ± 180.2	273.5 ± 179.9	−1.1	**54.2 ± 28.2**	19.3 ± 17.8	+64.3
Claude Sonnet 4.5	293.0 ± 173.9	307.7 ± 178.3	−5.0	213.2 ± 1062.5	30.3 ± 37.2	+85.8
o3†	255.8 ± 315.4	501.9 ± 2341.3	**−96.2**	101.6 ± 75.9	**0.0 ± 0.0**	**+100.0**
GPT-5†	**150.2 ± 198.2**	**161.4 ± 188.0**	−7.5	107.0 ± 69.0	**0.0 ± 0.0**	**+100.0**
Gemini 2.5 Flash	554.6 ± 1831.6	258.0 ± 231.2	**+53.5**	112.0 ± 90.1	0.1 ± 0.5	+99.9
Qwen3 8B	–	**51855.3 ± 460034.8**	–	–	2.0 ± 6.2	–
Ministral 3 8B	1643.3 ± 2618.4	1551.7 ± 2735.6	+5.6	**3057.4 ± 4112.4**	1354.6 ± 2166.2	+55.7

"–" Not reported: fewer than 50 valid TCs, or not evaluated (¶).
† Temperature parameter not supported, evaluated with default settings.
⋆ GPT-4o returned all-zero WCD values (trivial failure) for 51 of the 100 CBS TCs, leaving 49 valid CBS TCs in the CB setting and 17 in the FC setting. It produced valid WCD estimates for all 100 CQF TCs.
¶ Retired by the provider before the FC and prompt-sensitivity experiments, so only CB results are reported.
Not shown (fewer than 50 valid TCs in every setting): Llama 3.2 1B, DeepSeek-V3.2 (Thinking). Detailed failure modes are described in Appendix H.

Table 4: Percentage of flows with an absolute WCD error below 50, 100, 500, and 1000 μs in the CB setting for CBS and CQF. For each model, the percentage is computed per TC over the flows with a finite prediction and averaged over the valid TCs. FC results are given in Table 15.

Model	CBS (%)				CQF (%)			
	<50µs	<100µs	<500µs	<1000µs	<50µs	<100µs	<500µs	<1000µs
Grok 4.1 Fast†,¶	–	–	–	–	28.2	50.0	99.5	100.0
Grok 4.1 Fast (Non-Reasoning)¶	2.6	5.5	23.2	31.1	0.0	16.7	100.0	100.0
DeepSeek-V3.2 (Non-Thinking)¶	–	–	–	–	3.8	23.4	99.4	99.8
GPT-4o⋆	–	–	–	–	84.4	87.7	95.0	97.0
GPT-4o mini	1.0	6.8	74.3	96.6	1.7	18.9	98.1	99.6
Llama 3.3 70B	8.8	18.6	84.7	98.5	3.3	16.9	98.9	100.0
Mistral Medium 3.1	10.7	21.6	77.1	97.0	1.5	23.2	99.8	100.0
Mistral Large 3	14.8	25.3	86.8	98.0	48.6	92.6	100.0	100.0
Claude Sonnet 4.5	7.2	19.3	85.6	98.1	49.1	71.3	98.0	98.0
o3†	22.0	41.8	85.9	95.9	28.1	63.2	99.8	100.0
GPT-5†	39.4	60.4	94.1	98.0	41.9	58.4	99.9	100.0
Gemini 2.5 Flash	16.6	29.3	80.5	94.2	42.0	60.4	98.5	100.0
Ministral 3 8B	1.9	4.1	21.4	67.8	2.6	6.4	14.2	46.3

"–", †, ⋆, ¶ and the models not shown are explained in Table 3. All thresholds are in μs (↑ higher is better).

For CQF, performance is more varied, with median MAE ranging from 1.2 μs (GPT-4o) to 1,046 μs (Ministral 3 8B). GPT-4o achieves this lowest median despite returning all-zero WCD values for 51 of the 100 CBS TCs, suggesting it can correctly apply the CQF closed-form equation, while Mistral Large 3 achieves the lowest mean MAE (54.2 μs). Llama 3.2 1B exhibits the most severe hallucination failure, listing up to 1,013 invented flows (F0 to F1012) instead of the flows of the TC and returning WCD = 0 for all of them. Qwen3 8B has fewer than 50 valid TCs in the CB setting due to repeated API timeouts or responses with flow IDs that do not exist in the TC, and is reported only in the FC setting (Table 3). Ministral 3 8B, despite its small size, produces valid responses for both mechanisms but with large errors (MAE 1,643.3 μs on CBS and 3,057.4 μs on CQF), showing that a valid answer format alone does not lead to correct WCD values.

Effect of formula-in-context: Providing the NC equations in the prompt does not help most models on CBS. CBS WCD computation is not a single formula but a multi-step derivation of arrival curves, the service curve, and the delay bound. Of the nine models with CBS results in both settings, FC lowers the mean MAE only for Gemini 2.5 Flash (+53.5%) and Ministral 3 8B (+5.6%) and raises

it for the other seven, most strongly for o3 (-96.2%, median from 138.8 to 187.4 μs). This suggests that formulas help models that lack a working strategy but can hurt models that already have one. For CQF, where the WCD reduces to a single closed-form upper bound, FC lowers the MAE for 9 of 10 models, by 56 to 100%. o3 and GPT-5 reach an MAE of 0.0 μs, and Gemini 2.5 Flash and Llama 3.3 70B come close (0.1 and 5.1 μs). GPT-4o mini is the exception. Its CQF error rises from 180.5 to 1,377.1 μs and its median from 175.3 to 323.4 μs, so the increase is not caused only by a few outlier TCs. These results suggest that formulas help reliably when the task is substitution, but their effect depends on the model's own reasoning when the task requires a multi-step derivation.

Error thresholds: Error thresholds show how rarely the predictions are precise. In the CBS CB setting, even GPT-5 places only 39.4% of flows within 50 μs of the GT, and all other models stay at or below 22% (Table 4, with FC results in Table 15).

Prompt sensitivity: We repeat the CB evaluation with a paraphrased prompt (Variant A) and a prompt that requests plain text instead of JSON output (Variant B). On CBS, results are largely robust to paraphrasing for most models but sensitive to the output format (Table 16 in the Appendix). Plain-text output raises the mean CBS MAE by more than two orders of magnitude for GPT-4o mini (316 times higher, Table 16). Claude Sonnet 4.5, GPT-5, and o3 remain stable and reach a lower MAE under both variants than with the original CB prompt. The most stable models are o3 (CV 1.2%) and Claude Sonnet 4.5 (CV 2.0%). Further results are given in Tables 17 to 19 in the Appendix.

Real-world topology case study: To test whether the findings carry over to a real network topology, we evaluate CBS and CQF on the Orion Crew Exploration Vehicle (CEV) topology (Figure 15). For CBS, as on the synthetic topologies, FC lowers the mean MAE for only a minority of models (4 of 10), namely GPT-4o mini, o3, GPT-5, and Qwen3 8B. In the CB setting, o3 reaches the lowest MAE of 256.1 μs (Appendix Tables 24, 26, and 27). Per-TC results are given in Tables 28 and 29. For CQF, FC lowers the mean MAE for 7 of 11 models (Appendix Tables 25 and 30 to 33).

5 Conclusion

We present TSNBench, the first benchmark for evaluating LLM proficiency in Time-Sensitive Networking (TSN), comprising 939 expert-validated multiple-choice questions (MCQs) and 100 open-ended questions per mechanism for Credit-Based Shaper (CBS) and Cyclic Queuing and Forwarding (CQF). The ground truth (GT) WCD values are computed using a verified Network Calculus (NC) solver for CBS and closed-form mathematical upper bounds for CQF. We evaluate 16 LLMs and find that models achieve 67-95% MCQA accuracy yet fail substantially on open-ended WCD computation, with the best model (GPT-5) still deviating from the GT WCD by 150.2 μs on average on CBS. Despite CBS being extensively researched and an older mechanism, models cannot correctly apply NC, whereas CQF, with its simpler closed-form equation, is handled more successfully, suggesting that WCD computation performance is governed by mathematical complexity rather than mechanism maturity. Moreover, providing the WCD equations in the prompt lowers the CQF error for 9 of 10 models but the CBS error for only 2 of 9. TSNBench demonstrates that MCQ benchmarks substantially overestimate LLM capability in safety-critical domains.

Limitations and Future Directions: TSNBench has five main limitations. First, the MCQs are generated from open-access research papers, which limits coverage of recently standardized mechanisms such as ATS (Appendix F.2, Table 7). Second, the MCQs were generated with models from the same families that we evaluate, so some contamination cannot be ruled out. Our contamination analysis finds no advantage for the generator families, and the open-ended tasks are constructed specifically for TSNBench and are therefore unlikely to be affected. Third, MCQA accuracy is high for the strongest models, which limits its ability to distinguish between them. We view this saturation as further evidence for our main claim that MCQA overestimates practical proficiency, and the open-ended tasks are designed to expose this gap. Fourth, the open-ended evaluation covers only CBS and CQF. Extending to TAS is a natural next step, though its NP-hard gate control list (GCL) synthesis problem poses additional challenges beyond CBS and CQF. Fifth, the open-ended tasks use three synthetic topologies and one real-world topology and evaluate standalone model behavior with and without formulas in the prompt. These results should not be interpreted as a recommended deployment workflow for safety-critical TSN systems. Evaluating LLMs in settings where they produce checkable artifacts verified by deterministic analysis tools is an important direction for future versions of TSNBench.

Acknowledgments and Disclosure of Funding

Funding in direct support of this work: This work received no external funding or financial support.

Competing interests: The authors declare no competing interests.

Disclaimer: Silviu S. Craciunas is employed by NXP Semiconductors and Technical University of Denmark (DTU). The views expressed in this paper are those of the authors and do not necessarily reflect those of NXP Semiconductors.

References

Astrit Ademaj and Thomas Enzinger. Industrial Automation Traffic Types and their Mapping to QoS/TSN Mechanisms. IEEE 802.1 contribution, IEC/IEEE 60802, 2018. URL https://www.ieee802.org/1/files/public/docs2018/60802-ademaj-traffic-type-characterization-1118-v01.pdf.

Muhammad Adil, Tie Qiu, Xiaobo Zhou, Danish Javeed, Zhenrui Cao, and Dapeng Oliver Wu. Integrated 5G and Time Sensitive Networking for Emerging Applications: A Survey of Advancements, Challenges, and Future Directions. *IEEE Communications Surveys & Tutorials*, 28:4016–4050, 2026. doi: 10.1109/COMST.2025.3632286.

Konstantinos Alexandris, Paul Pop, and Tongtong Wang. Configuration and Evaluation of Multi-CQF Shapers in IEEE 802.1 Time-Sensitive Networking (TSN). *IEEE Access*, 10:109068–109081, 2022. doi: 10.1109/ACCESS.2022.3214007.

Jeffrey G. Andrews, Stefano Buzzi, Wan Choi, Stephen V. Hanly, Angel Lozano, Anthony C. K. Soong, and Jianzhong Charlie Zhang. What Will 5G Be? *IEEE Journal on Selected Areas in Communications*, 32(6):1065–1082, 2014. doi: 10.1109/JSAC.2014.2328098.

Dietmar Bruckner, Marius-Petru Stănică, Richard Blair, Sebastian Schriegel, Stephan Kehrer, Maik Seewald, and Thilo Sauter. An Introduction to OPC UA TSN for Industrial Communication Systems. *Proceedings of the IEEE*, 107(6):1121–1131, 2019. doi: 10.1109/JPROC.2018.2888703.

Daniel Bujosa, Mohammad Ashjaei, Alessandro V Papadopoulos, Thomas Nolte, and Julián Proenza. HERMES: Heuristic multi-queue scheduler for TSN time-triggered traffic with zero reception jitter capabilities. In *Proceedings of the 30th International Conference on Real-Time Networks and Systems*, pages 70–80. Association for Computing Machinery, 2022. doi: 10.1145/3534879.3534906.

Daniel Bujosa Mateu. *Improved Configuration and Analysis Solutions for Time-Sensitive Networks with Support for Legacy Systems*. PhD thesis, Mälardalen University, Västerås, Sweden, 2024.

Martin Böhm and Diederich Wermser. Multi-Domain Time-Sensitive Networks—Control Plane Mechanisms for Dynamic Inter-Domain Stream Configuration. *Electronics*, 10(20), 2021. ISSN 2079-9292. doi: 10.3390/electronics10202477.

Silviu S. Craciunas, Ramon Serna Oliver, Martin Chmelík, and Wilfried Steiner. Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, RTNS '16, pages 183–192, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450347877. doi: 10.1145/2997465.2997470. URL https://doi.org/10.1145/2997465.2997470.

Rubi Debnath, Mustafa Selman Akinci, Devika Ajith, and Sebastian Steinhorst. 5GTQ: QoS-Aware 5G-TSN Simulation Framework. In *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, pages 1–7, 2023a. doi: 10.1109/VTC2023-Fall60731.2023.10333533.

Rubi Debnath, Philipp Hortig, Luxi Zhao, and Sebastian Steinhorst. Advanced Modeling and Analysis of Individual and Combined TSN Shapers in OMNeT++. In *2023 IEEE 29th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 176–185, 2023b. doi: 10.1109/RTCSA58653.2023.00029.

Rubi Debnath, Philipp Hortig, Luxi Zhao, and Sebastian Steinhorst. Quantifying the Impact of Frame Preemption on Combined TSN Shapers. In *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, pages 1–9, 2024. doi: 10.1109/NOMS59830.2024.10575564.

Rubi Debnath, Mohammadreza Barzegaran, and Sebastian Steinhorst. Toward an Optimized Multi-Cyclic Queuing and Forwarding in Time-Sensitive Networking With Time Injection. *IEEE Internet of Things Journal*, 12(20):43034–43051, 2025a. doi: 10.1109/JIOT.2025.3597560.

Rubi Debnath, Luxi Zhao, Mohammadreza Barzegaran, and Sebastian Steinhorst. CyclicSim: Comprehensive Evaluation of Cyclic Shapers in Time-Sensitive Networking. In *2025 IEEE 22nd Consumer Communications & Networking Conference (CCNC)*, pages 01–09, 2025b. doi: 10.1109/CCNC54725.2025.10975975.

Rubi Debnath, Luxi Zhao, and Sebastian Steinhorst. Learning-Based Traffic Classification for Mixed-Critical Flows in Time-Sensitive Networking. In *ICC 2025 - IEEE International Conference on Communications*, pages 5926–5932, 2025c. doi: 10.1109/ICC52391.2025.11161468.

Kasra Ekrad, Inés Alvarez Vadillo, Bjarne Johansson, Saad Mubeen, and Mohammad Ashjaei. A Methodology to Map Industrial Automation Traffic to TSN Traffic Classes. In *2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, 2025. doi: 10.1109/ETFA65518.2025.11205571.

Leonard Elliott. Time-Sensitive Networking (TSN) in Military Ground Vehicle Architectures. In *2024 NDIA Michigan Chapter Ground Vehicle Systems Engineering and Technology Symposium*, Novi, Michigan, United States, August 2024. National Defense Industrial Association Michigan Chapter. doi: 10.4271/2024-01-4122.

Mohamed Amine Ferrag, Abderrahmane Lakas, and Mérouane Debbah. 6G-Bench: An Open Benchmark for Semantic Communication and Network-Level Reasoning With Foundation Models in AI-Native 6G Networks. *IEEE Open Journal of the Communications Society*, 7:3305–3330, 2026. doi: 10.1109/OJCOMS.2026.3680457.

Norman Finn. Introduction to Time-Sensitive Networking. *IEEE Communications Standards Magazine*, 2(2):22–28, 2018. doi: 10.1109/MCOMSTD.2018.1700076.

Tiziana Fiori, Francesco Giacinto Lavacca, Francesco Valente, and Vincenzo Eramo. Proposal and Investigation of a Lite Time Sensitive Networking Solution for the Support of Real Time Services in Space Launcher Networks. *IEEE Access*, 12:10664–10680, 2024. doi: 10.1109/ACCESS.2024.3353466.

Pranshav Gajjar, Cong Shen, and Vijay K Shah. Tele-LLM-Hub: Building Context-Aware Multi-Agent LLM Systems for Telecom Networks. In *NeurIPS 2025 Workshop: AI and ML for Next-Generation Wireless Communications and Networking*, 2025. URL https://openreview.net/forum?id=AencYkmJtl.

Voica Gavriluţ and Paul Pop. Traffic-type Assignment for TSN-based Mixed-criticality Cyber-physical Systems. *ACM Transactions on Cyber-Physical Systems*, 4(2), January 2020. ISSN 2378-962X. doi: 10.1145/3371708.

Voica Gavriluţ, Luxi Zhao, Michael L. Raagaard, and Paul Pop. AVB-Aware Routing and Scheduling of Time-Triggered Traffic for TSN. *IEEE Access*, 6:75229–75243, 2018. doi: 10.1109/ACCESS.2018.2883644.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Aditya K, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 44123–44279. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/89e44582fd28ddfea1ea4dcb0ebbf4b0-Paper-Datasets_and_Benchmarks.pdf.

Xingang Guo, Yaxin Li, XiangYi Kong, Yilan Jiang, Xiayu Zhao, Zhihua Gong, Yufan Zhang, Daixuan Li, Tianle Sang, Beixiao Zhu, Gregory Jun, Yingbing Huang, Yiqi Liu, Yuqi Xue, Rahul Dev Kundu, Qi Jian Lim, Yizhou Zhao, Luke Alexander Granger, Mohamed Badr Younis, Darioush Keivan, Nippun Sabharwal, Shreyanka Sinha, Prakhar Agarwal, Kojo Vandyck, Hanlin Mai, Zichen Wang, Aditya Venkatesh, Ayush Barik, Jiankun Yang, Chongying Yue, Jingjie He, Libin Wang, Licheng Xu, Hao Chen, Jinwen Wang, Liujun Xu, Rushabh Shetty, Ziheng Guo, Dahui Song, Manvi Jha, Weijie Liang, Weiman Yan, Bryan Zhang, Sahil Bhandary Karnoor, Jialiang Zhang, Rutva Pandya, Xinyi Gong, Mithesh Ballae Ganesh, Feize Shi, Ruiling Xu, Yifan Zhang, Yanfeng Ouyang, Lianhui Qin, Elyse Rosenbaum, Corey Snyder, Peter Seiler, Geir Dullerud, Xiaojia Shelly Zhang, Zuofu Cheng, Pavan Kumar Hanumolu, Jian Huang, Mayank Kulkarni, Mahdi Namazifar, Huan Zhang, and Bin Hu. Toward Engineering AGI: Benchmarking the Engineering Design Capabilities of LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=Wmsnx7EPel.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Linard Hoessly. On misconceptions about the Brier score in binary prediction models. *Glob. Epidemiol.*, 11: 100242, June 2026.

Tianyu Hua, Harper Hua, Violet Xiang, Benjamin Klieger, Sang T. Truong, Weixin Liang, Fan-Yun Sun, and Nick Haber. ResearchCodeBench: Benchmarking LLMs on Implementing Novel Machine Learning Research Code. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=3k70Vt0YFS.

Dong Huang, Yuhao Qing, Weiyi Shang, Heming Cui, and Jie M. Zhang. EffiBench: Benchmarking the Efficiency of Automatically Generated Code. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 11506–11544. Curran Associates, Inc., 2024. doi: 10.52202/079017-0367. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/15807b6e09d691fe5e96cdecde6d7b80-Paper-Datasets_and_Benchmarks_Track.pdf.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Benchmarking Large Language Models as AI Research Agents. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023. URL https://openreview.net/forum?id=kXlTY0BmK3.

IEEE. IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams. *IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005)*, pages C1–72, 2010. doi: 10.1109/IEEESTD.2009.5375704.

IEEE. IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks–Amendment 29: Cyclic Queuing and Forwarding. *IEEE 802.1Qch-2017 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd(TM)-2015, IEEE Std 802.1Q-2014/Cor 1-2015, IEEE Std 802.1Qbv-2015, IEEE Std 802.1Qbu-2016, IEEE Std 802.1Qbz-2016, and IEEE Std 802.1Qci-2017)*, pages 1–30, 2017. doi: 10.1109/IEEESTD.2017.7961303.

IEEE. IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks. *IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014)*, pages 1–1993, 2018. doi: 10.1109/IEEESTD.2018.8403927.

Jason J Jackson, Terry Huang, Henry Velasquez, Kevin Zhu, and Sunishchal Dev. Predicting Emergent Software Engineering Capabilities by Fine-tuning. In *NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling*, 2025. URL https://openreview.net/forum?id=EwchHtwavV.

Sunjun Kweon, Jiyoun Kim, Heeyoung Kwak, Dongchul Cha, Hangyul Yoon, Kwanghyun Kim, Jeewon Yang, Seunghyun Won, and Edward Choi. EHRNoteQA: An LLM Benchmark for Real-World Clinical Practice Using Discharge Summaries. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 124575–124611. Curran Associates, Inc., 2024. doi: 10.52202/079017-3958. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e15c4afff22f12c4986c1fcb4e941e03-Paper-Datasets_and_Benchmarks_Track.pdf.

Shuyue Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan S. Ilgen, Emma Pierson, Pang Wei Koh, and Yulia Tsvetkov. MediQ: Question-Asking LLMs and a Benchmark for Reliable Interactive Clinical Reasoning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 28858–28888. Curran Associates, Inc., 2024. doi: 10.52202/079017-0908. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/32b80425554e081204e5988ab1c97e9a-Paper-Conference.pdf.

Junling Liu, Peilin Zhou, Yining Hua, Dading Chong, Zhongyu Tian, Andrew Liu, Helin Wang, Chenyu You, Zhenhua Guo, Lei Zhu, and Michael Lingzhi Li. Benchmarking Large Language Models on CMExam - A comprehensive Chinese Medical Exam Dataset. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 52430–52452. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a48ad12d588c597f4725a8b84af647b5-Paper-Datasets_and_Benchmarks.pdf.

Yongsen Ma, Gang Zhou, and Shuangquan Wang. WiFi Sensing with Channel State Information: A Survey. *ACM Comput. Surv.*, 52(3), June 2019. ISSN 0360-0300. doi: 10.1145/3310194. URL https://doi.org/10.1145/3310194.

Ali Maatouk, Fadhel Ayed, Nicola Piovesan, Antonio De Domenico, Merouane Debbah, and Zhi-Quan Luo. TeleQnA: A Benchmark Dataset to Assess Large Language Models Telecommunications Knowledge. *IEEE Network*, 40(2):253–260, 2026. doi: 10.1109/MNET.2025.3576035.

Ahmed Nasrallah, Akhilesh S. Thyagaturu, Ziyad Alharbi, Cuixiang Wang, Xing Shao, Martin Reisslein, and Hesham Elbakoury. Performance Comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS). *IEEE Access*, 7:44165–44181, 2019. doi: 10.1109/ACCESS.2019.2908613.

Giwa Oluwaseyi, Michael Adewole, Tobi Awodumila, and Pelumi Aderinto. The LLM as a Network Operator: A Vision for Generative AI in the 6G Radio Access Network. In *NeurIPS 2025 Workshop: AI and ML for Next-Generation Wireless Communications and Networking*, 2025. URL https://openreview.net/forum?id=81mgAfsFJv.

Maja Pavlovic. Understanding Model Calibration - A gentle introduction and visual exploration of calibration and the expected calibration error (ECE). In *The Fourth Blogpost Track at ICLR 2025*, 2025. URL https://openreview.net/forum?id=BxBeCjQd2y.

Yifei Peng, Boxin Shi, Tigang Jiang, Xiaodong Tu, Du Xu, and Kun Hua. A Survey on In-Vehicle Time-Sensitive Networking. *IEEE Internet of Things Journal*, 10(16):14375–14396, 2023. doi: 10.1109/JIOT.2023.3264909.

Long Phan, Alice Gatti, Nathaniel Li, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Dan Hendrycks, et al. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649(8099):1139–1146, January 2026. ISSN 1476-4687. doi: 10.1038/s41586-025-09962-4. URL http://dx.doi.org/10.1038/s41586-025-09962-4.

Paul Pop, Michael Lander Raagaard, Silviu S. Craciunas, and Wilfried Steiner. Design Optimization of Cyber-Physical Distributed Systems using IEEE Time-sensitive Networks (TSN). *IET-CPS*, 1(1):86–94, 2016. doi: 10.1049/iet-cps.2016.0021.

Laria Reynolds and Kyle McDonell. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380959. doi: 10.1145/3411763.3451760. URL https://doi.org/10.1145/3411763.3451760.

Walid Saad, Mehdi Bennis, and Mingzhe Chen. A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems. *IEEE Network*, 34(3):134–142, 2020. doi: 10.1109/MNET.001.1900287.

Jorge Sanchez-Garrido, Beatriz Aparicio, José Gabriel Ramírez, Rafael Rodriguez, Mariasole Melara, Lorenzo Cercós, Eduardo Ros, and Javier Diaz. Implementation of a Time-Sensitive Networking (TSN) Ethernet Bus for Microlaunchers. *IEEE Transactions on Aerospace and Electronic Systems*, 57(5):2743–2758, 2021. doi: 10.1109/TAES.2021.3061806.

Ramon Serna Oliver, Silviu S. Craciunas, and Wilfried Steiner. IEEE 802.1Qbv Gate Control List Synthesis Using Array Theory Encoding. In *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 13–24, 2018. doi: 10.1109/RTAS.2018.00008.

Prakhar Sharma and Vinod Yegneswaran. PROSPER: Extracting Protocol Specifications Using Large Language Models. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, HotNets '23, pages 41–47, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400704154. doi: 10.1145/3626111.3628205. URL https://doi.org/10.1145/3626111.3628205.

Yongliang Shen, Kaitao Song, Xu Tan, Wenqi Zhang, Kan Ren, Siyu Yuan, Weiming Lu, Dongsheng Li, and Yueting Zhuang. TaskBench: Benchmarking Large Language Models for Task Automation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 4540–4574. Curran Associates, Inc., 2024. doi: 10.52202/079017-0148. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/085185ea97db31ae6dcac7497616fd3e-Paper-Datasets_and_Benchmarks_Track.pdf.

Johannes Specht and Soheil Samii. Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks. In *2016 28th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 75–85, 2016. doi: 10.1109/ECRTS.2016.27.

Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. SciEval: a multi-level large language model evaluation benchmark for scientific research. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i17.29872. URL https://doi.org/10.1609/aaai.v38i17.29872.

Changjie Wang, Mariano Scazzariello, Alireza Farshin, Simone Ferlin, Dejan Kostić, and Marco Chiesa. NetConfEval: Can LLMs Facilitate Network Configuration? *Proc. ACM Netw.*, 2(CoNEXT2), June 2024a. doi: 10.1145/3656296. URL https://doi.org/10.1145/3656296.

Xiaolong Wang, Haipeng Yao, Tianle Mai, Zehui Xiong, Fu Wang, and Yunjie Liu. Joint Routing and Scheduling With Cyclic Queuing and Forwarding for Time-Sensitive Networks. *IEEE Transactions on Vehicular Technology*, 72(3):3793–3804, 2023. doi: 10.1109/TVT.2022.3216958.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark. In *The Thirty-Eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b. URL https://openreview.net/forum?id=y10DM6R2r3.

Stefan Windmann, Janis Albrecht, Maxim Friesen, and Jürgen Jasperneite. NetPilot - Towards LLM-Assisted Configuration of Hybrid TSN/5G Networks. In *2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4, 2025. doi: 10.1109/ETFA65518.2025.11205555.

Jiacheng Xie, Yang Yu, Ziyang Zhang, Shuai Zeng, Jiaxuan He, Ayush Vasireddy, Xiaoting Tang, Congyu Guo, Lening Zhao, Congcong Jing, Guanghui An, and Dong Xu. TCM-Ladder: A Benchmark for Multimodal Question Answering on Traditional Chinese Medicine. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=ZDrT1eG54T.

Jinli Yan, Wei Quan, Xuyan Jiang, and Zhigang Sun. Injection Time Planning: Making CQF Practical in Time-Sensitive Networking. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 616–625, 2020. doi: 10.1109/INFOCOM41043.2020.9155434.

Jialin Yang, Dongfu Jiang, Tony He, Sherman Siu, Yuxuan Zhang, Disen Liao, Zhuofeng Li, Huaye Zeng, Yiming Jia, Haozhe Wang, Benjamin Schneider, Chi Ruan, Wentao Ma, Zhiheng Lyu, Yifei Wang, Yi Lu, Quy Duc Do, Ziyan Jiang, Ping Nie, and Wenhu Chen. StructEval: Benchmarking LLMs' Capabilities to Generate Structural Outputs. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=buDwV7LUA7. J2C Certification.

Kouros Zanbouri, Md. Noor-A-Rahim, Jobish John, Cormac J. Sreenan, H. Vincent Poor, and Dirk Pesch. A Comprehensive Survey of Wireless Time-Sensitive Networking (TSN): Architecture, Technologies, Applications, and Open Issues. *IEEE Communications Surveys & Tutorials*, 27(4):2129–2155, 2025. doi: 10.1109/COMST.2024.3486618.

Tianyu Zhang, Gang Wang, Chuanyu Xue, Jiachen Wang, Mark Nixon, and Song Han. Time-Sensitive Networking (TSN) for Industrial Automation: Current Advances and Future Directions. *ACM Comput. Surv.*, 57(2), October 2024. ISSN 0360-0300. doi: 10.1145/3695248. URL https://doi.org/10.1145/3695248.

Luxi Zhao, Paul Pop, Zhong Zheng, and Qiao Li. Timing Analysis of AVB Traffic in TSN Networks Using Network Calculus. In *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 25–36, 2018. doi: 10.1109/RTAS.2018.00009.

Luxi Zhao, Paul Pop, Zhong Zheng, Hugo Daigmorte, and Marc Boyer. Latency Analysis of Multiple Classes of AVB Traffic in TSN With Standard Credit Behavior Using Network Calculus. *IEEE Transactions on Industrial Electronics*, 68(10):10291–10302, 2021. doi: 10.1109/TIE.2020.3021638.

Luxi Zhao, Paul Pop, and Sebastian Steinhorst. Quantitative Performance Comparison of Various Traffic Shapers in Time-Sensitive Networking. *IEEE Transactions on Network and Service Management*, 19(3):2899–2928, 2022. doi: 10.1109/TNSM.2022.3180160.

Luxi Zhao, Yida Yan, and Xuan Zhou. Minimum Bandwidth Reservation for CBS in TSN With Real-Time QoS Guarantees. *IEEE Transactions on Industrial Informatics*, 20(4):6187–6198, 2024. doi: 10.1109/TII.2023.3342466.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We present TSNBench, the first benchmark for evaluating LLM proficiency in safety-critical Time-Sensitive Networking. TSNBench covers TSN knowledge through expert-validated MCQA and computational reasoning through open-ended WCD computation for CBS and CQF. Ground-truth values are computed using a verified state-of-the-art Network Calculus solver for CBS and closed-form mathematical upper bounds for CQF. Dataset construction is detailed in Section 3, and evaluation results are presented in Section 4. A detailed description of TSNBench is also given in Appendix F.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of TSNBench in detail in Section 5. A more detailed explanation is provided in Appendix A.1.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The Network Calculus background and methodology underlying the CBS ground truth are provided in Appendix C. The closed-form WCD equation and its proof for CQF are given in Section 3.6 and Appendix E, respectively. All network constants, parameters, and assumptions used in the open-ended evaluation are detailed in Appendix F.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Full details of dataset generation, evaluation framework, and results are provided in Section 3, Section 4, and Appendix F. The TSNBench dataset, comprising MCQs and open-ended WCD test cases with GT values, is publicly released on Kaggle under the CC-BY 4.0 license at `https://doi.org/10.34740/kaggle/ds/13019279`. The code, prompts, and evaluation scripts are open-sourced on GitHub under the MIT license at `https://github.com/tum-esi/TSNBench`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The TSNBench dataset is publicly available on Kaggle at https://doi.org/10.34740/kaggle/ds/13019279 under the CC-BY 4.0 license, with Croissant metadata. It contains the complete MCQA dataset with answer keys and all open-ended test cases with network topologies, flow parameters, and routing configurations. Ground-truth WCD values are provided for all test cases, computed via Network Calculus for CBS and via closed-form mathematical upper bounds for CQF, enabling direct result verification. The full TSNBench codebase is available at https://github.com/tum-esi/TSNBench under the MIT license, including a README with step-by-step environment setup and reproduction instructions.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental details including research document selection, MCQA generation models, expert review criteria, evaluation models, temperature configurations, and prompt design are provided in Section 3 and Section 4. Dataset construction is summarized in Table 1 and illustrated in Figure 2 for MCQA and Figure 3 for open-ended questions. A more detailed explanation is given in Appendices F and F.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For the open-ended questions, Tables 3, 13, and 14 report the MAE of each model as mean $\pm$ standard deviation, and Tables 13 and 14 also give the median MAE. The standard deviation is the population standard deviation of the per-TC MAE across the valid TCs (n) of a model. The prediction of each flow is averaged over up to three valid runs at one temperature per model before the MAE is computed, so the standard deviation captures the variation across TCs, not across runs. For MCQA, accuracy is averaged over three runs, and the consistency of answers across runs is reported in Table 10. We do not run formal significance tests. More results are given in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments are conducted via vendor APIs on a standard workstation with no specialized compute required (details are given in Section 4). Per-model API latency and cost for both MCQA and the closed-book open-ended evaluations are reported in Table 11 in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide the broader positive and negative impact of TSNBench in Appendix A.3 and A.4 respectively.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: TSNBench does not pose any such risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The TSNBench code and dataset are owned and released by the authors under the MIT and CC-BY 4.0 licenses, respectively. All research documents used for dataset construction are open-access publications, arXiv versions of published or accepted papers, or author manuscript versions. The authors of all directly used manuscript versions are properly credited, and full acknowledgments are disclosed in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: TSNBench introduces two new assets, the 939-question expert-validated MCQA dataset and 100 open-ended WCD computation questions per mechanism on synthetic topologies plus 10 open-ended questions per mechanism on the real-world Orion CEV topology, all with verified ground-truth WCD values. Both are publicly released on Kaggle at `https://doi.org/10.34740/kaggle/ds/13019279` under the CC-BY 4.0 license. The code and evaluation scripts are released on GitHub at `https://github.com/tum-esi/TSNBench` under the MIT license, with a README that provides complete instructions for evaluation setup and result reproduction.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: In this paper, we do not have any involvement with crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: NA.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are an integral part of the TSNBench framework, used for keyword extraction and MCQA generation as described in Section 3. All extracted keywords and generated questions are reviewed by five domain experts with extensive TSN research experience, ensuring that LLM-generated content does not compromise the scientific rigor or correctness of the benchmark.

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

A Limitations and Broader Impact

A.1 Limitations

While TSNBench fills a significant research gap and proposes a step forward towards evaluating TSN capabilities in LLMs, it has several limitations:

Dataset scope: TSNBench currently only covers CBS and CQF in open-ended questions. Evaluating other TSN mechanisms is necessary to fully cover the entire TSN mechanisms.

MCQA Scope: MCQs are solely developed using published research papers and the IEEE standards are not used to generate the MCQs. Solving the license issue and utilizing standards to include MCQs using IEEE 802.1 standard will enhance the entire MCQA dataset.

Topology coverage: TSNBench open-ended questions currently cover four different topologies: one-switch, medium-mesh, ring topology, and Orion CEV topology. Covering diverse topologies and flow parameters will present a comprehensive evaluation.

A.2 Improvement Strategies

To address the limitations of TSNBench, we propose the following additions and improvements in future versions of TSNBench.

1. **Larger and more diverse dataset:** Our current TSNBench dataset covers 100 TCs per mechanism across three synthetic topology types, and 10 TCs on the real-world Orion CEV topology. In future versions, we will include larger and more complex topologies with higher flow counts. As model performance improves, more complex open-ended evaluations should be integrated with complex topologies and combined TSN mechanisms.
2. **Additional scheduling mechanisms:** TSNBench currently evaluates CBS and CQF. Future versions should extend to TAS and ATS to cover a broader range of the TSN standard suite.
3. **Updated MCQA:** Our MCQA dataset was developed using published or accepted research documents that are openly available. In future work, we will update the dataset with MCQAs formulated directly from TSN standards.
4. **Fine-tuned and domain-adapted models.** TSNBench currently evaluates general-purpose LLMs without any TSN-specific fine-tuning. Future versions should benchmark domain-adapted models trained on TSN standards and network calculus literature.

A.3 Broader Impact

TSNBench enables the real-time systems community and the machine learning community to objectively measure LLM performance and readiness for management and deployment assistance in safety-critical deterministic networks. By highlighting the critical aspects of TSN and the performance gap of the models between MCQA and computational reasoning, TSNBench exposes weaknesses of current models which may lead to misconfigurations and safety-critical issues. This benchmark provides a concrete direction to improve LLMs for deterministic networking. TSNBench further highlights the potential benefits of using LLMs to automate the management and deployment of TSN networks. Moreover, open-sourced GT WCD values computed by NC solvers for CBS provide a reliable resource for the entire community to further evaluate different benchmarking datasets.

A.4 Negative Impacts

While TSNBench is intended to advance research on LLM proficiency in TSN, we acknowledge the following potential negative impacts.

Overreliance on model outputs: Models trained on the open-access dataset provided by TSNBench may achieve high accuracy on WCD analysis tasks, which could lead practitioners to deploy such models directly in real-world deployments without independent verification. Any WCD values or network configuration decisions produced by an LLM should be verified using formally verified solvers and NC tools before real-world deployment.

False confidence from MCQA performance: Our results demonstrate that strong MCQA performance does not transfer to open-ended WCD estimation. A practitioner or system engineer who evaluates an LLM solely on MCQA benchmarks may incorrectly conclude that the model is suitable for TSN configuration tasks, leading to unsafe deployments in systems where timing guarantees are required.

Data contamination and benchmark overfitting: As TSNBench is released as an open-access dataset, future models may be trained directly on the benchmark questions, leading to inflated performance that does not reflect genuine TSN reasoning capability. We recommend that researchers introduce randomization in the test cases to prevent bias in results. Researchers should be cautious when interpreting results from models whose training data may overlap with the TSNBench dataset.

Misuse of the dataset: The dataset can be used to train models to configure TSN networks. Owing to the safety-critical nature of TSN applications, such models could potentially be exploited by attackers to manipulate network configurations, introduce timing violations, or deliberately cause deadline misses in industrial and automotive systems.

B Time-Sensitive Networking

Time-Sensitive Networking (TSN) [Finn, 2018] is a set of amendments and additions to the IEEE 802.1 standards that, since its inception in 2012, has become one of the most relevant technologies for enabling deterministic and real-time communications over Ethernet networks. TSN extends standard Ethernet by introducing mechanisms for bounded latency, low jitter, and high reliability, making it suitable for applications such as industrial automation, automotive systems, and professional audio-video networks. Figure 5 showcases a simple TSN network with flows.

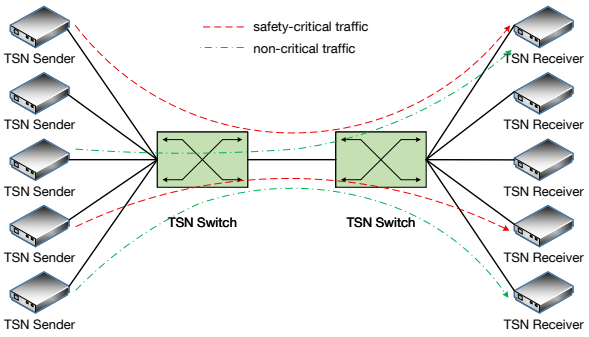


Figure 5: A sample TSN network with TSN senders, receivers, and TSN switches in the network. TSN senders are sending mixed-criticality traffic, including safety-critical and non-critical traffic to the TSN receivers.

In TSN, communication between end-stations is based on the transmission of Ethernet frames across a network of interconnected Ethernet links and TSN switches. These switches, as well as the output ports of end-stations, implement a queuing architecture with up to eight First-In-First-Out (FIFO) queues, each associated with one of the eight traffic priorities defined in IEEE 802.1Q [IEEE, 2018]. TSN is not just limited to the wired domain. The growing necessity of deterministic communication has extended to wireless domain gaining a significant interest in wireless-TSN networks. Although TSN is fundamentally an IEEE 802.1 bridged Ethernet technology, wireless and 5G-TSN [Debnath et al., 2023a] integration requires additional adaptation or translation functions, together with time-synchronization mechanisms that preserve deterministic latency guarantees across heterogeneous network segments. We showcase a 5G-TSN system in Figure 6, where TSN senders are sending mixed criticality traffic types to wireless receiver nodes over a TSN switch and 5G system in the network. Some of the most commonly used abbreviations in TSN are given in Table 5.

Frames are classified into traffic classes and assigned to egress queues based on their priority, with transmission selection typically governed by strict priority. Industrial TSN traffic is commonly categorized into traffic types such as isochronous traffic, cyclic-synchronous traffic, cyclic-asynchronous traffic, network-control traffic, alarms and events, configuration and diagnostics, and best-effort

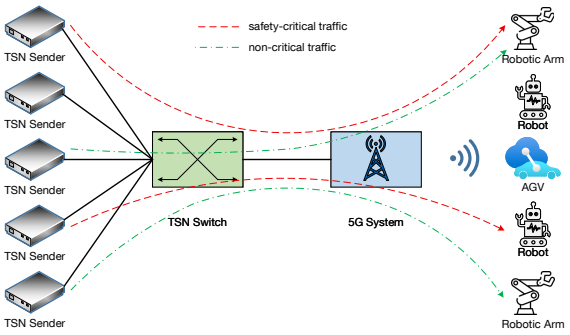


Figure 6: A sample wireless-TSN network with TSN senders, wireless receivers (such as robotic arm and automated guided vehicles (AGVs)), TSN switches, and 5G system in the network. TSN senders are sending mixed-critical including safety-critical and non-critical traffic to the wireless receivers.

Table 5: Abbreviations and mechanisms used in TSNBench.

Abbreviation	**Full form**
TSN	Time-Sensitive Networking
TAS	Time-Aware Shaper
CBS	Credit-Based Shaper
ATS	Asynchronous Traffic Shaper
CQF	Cyclic Queuing and Forwarding
NC	Network Calculus
WCD	Worst-Case Delay
AVB	Audio Video Bridging
TT	Time-Triggered
GCL	Gate Control List
BE	Best Effort

traffic [Ademaj and Enzinger, 2018]. These traffic types require different timing guarantees: safety-critical isochronous traffic is typically mapped to time-triggered (TT) traffic, requiring guaranteed latency and bounded jitter, and is commonly handled by time-triggered mechanisms such as the Time-Aware Shaper (TAS) [Craciunas et al., 2016, Serna Oliver et al., 2018]. In contrast, cyclic-synchronous or cyclic-asynchronous traffic that requires bounded end-to-end latency but less stringent jitter control is commonly mapped to AVB stream traffic and is often supported by the Credit-Based Shaper (CBS) [Zhao et al., 2018]. TSN also defines mechanisms such as Asynchronous Traffic Shaper (ATS) [Specht and Samii, 2016, Debnath et al., 2023b, Nasrallah et al., 2019], Frame preemption (FP) [Debnath et al., 2024], and Cyclic Queuing and Forwarding (CQF) [Wang et al., 2023, Debnath et al., 2025a, Yan et al., 2020] to provide deterministic communication under different traffic and deployment assumptions.

These mechanisms regulate when and how frames are transmitted, allowing the network to provide guarantees such as bounded delay, jitter, and controlled bandwidth allocation. In the MCQA dataset of TSNBench, we covered the basics of different TSN mechanisms, including TAS, CBS, ATS, and CQF. The MCQAs are theoretical in nature and cover the basic understanding of the mechanisms without going into their mathematical or analytical details. In contrast, for the open-ended questions, we evaluate the capability of the models to perform numerical analysis, formulate mathematical equations, and find the WCD values for the flows in the network. For this, we selected two TSN mechanisms: CBS and CQF. The WCD values of the flows using the CBS mechanism are calculated using NC analysis, which is mathematically complex. Therefore, we also evaluate the CQF mechanism as a simpler mechanism. The WCD values of the flows using the CQF mechanism can be directly calculated using the routing of the flow and the cycle duration. The working mechanism and architecture of CBS and CQF are described in detail in Appendices D and E. The theory of NC is further explained along with the mathematical equations in Appendix C.

C Network Calculus Theory

Network Calculus (NC) is a theory for calculating worst-case bounds in communication networks based on min-plus algebra. Its basic paradigm involves two operators: convolution $\otimes$

$$(f \otimes g)(t) = \inf_{0 \le s \le t} \{f(t-s) + g(s)\}, \tag{5}$$

and deconvolution $\oslash$,

$$(f \oslash g)(t) = \sup_{s \ge 0} \{f(t+s) - g(s)\}. \tag{6}$$

Based on this algebra, the arrival curve and the service curve are constructed to describe the maximum arrival traffic data and the minimum service capability over any time interval, respectively. In the hybrid TSN/TAS+CBS architecture, the service for ET traffic is constrained not only by the bandwidth reservation, but also by high-priority TT traffic. We adopt the state-of-the-art network calculus model [Zhao et al., 2021, 2024] to ensure deadline guarantees for ET flows with an arbitrary number of SR classes in the TSN/TAS+CBS architecture. Since our open-ended CBS questions do not have any TAS mechanism, we use the TSN/TAS+CBS architecture without the TAS mechanism in it with only CBS mechanism for the AVB flows in the network.

As described in [Zhao et al., 2024], the service curve $\beta(t)$ is for constraining the minimum service capabilities, satisfying

$$\mathcal{R}^*(t) \ge (\mathcal{R} \otimes \beta)(t). \tag{7}$$

The function $\mathcal{R}(t)$ (resp. $\mathcal{R}^*(t)$) is the input (resp. output) cumulative function counting the total data bits of the flow that arrive at (resp. depart from) the server up to time t. A typical example of a service curve is the rate-latency form,

$$\beta_{R,T}(t) = R[t-T]^+ \tag{8}$$

with the service rate R and latency T. The notation $[x]^+$ equals x if $x \ge 0$, and 0 otherwise.

In the hybrid TSN/TAS+CBS architecture, the CBS service curve [Zhao et al., 2021] for the arbitrary SR Class M_i ($i \in [1, N_{SR}]$) with the impact of TT traffic at the output port h is,

$$\beta^h_{M_i}(t) = idSl^h_{M_i} \left[t - \frac{\alpha^h_{TAS}(t)}{C} - \frac{c^{h,\max}_{M_i}}{idSl^h_{M_i}} \right]^+_{\uparrow}, \tag{9}$$

where $c^{h,\max}_{M_i}$ is the credit upper bound for SR Class M_i, $idSl$ denotes the idleSlope,

$$c^{h,\max}_{M_i} = idSl^h_{M_i} \cdot \frac{\sum_{j=1}^{i-1} c^{h,\min}_{M_j} - l^{h,\max}_{>i}}{\sum_{j=1}^{i-1} idSl^h_{M_j} - C}, \tag{10}$$

where $l^{h,\max}_{>i} = \max_{j>i}\{l^{h,\max}_{M_j}, l^{h,\max}_{BE}\}$ is the maximum frame size with priority lower than Class M_i at h, $l^{h,\max}_{M_j}$ is the maximum frame size of Class M_i at h, and $c^{h,\min}_{M_i}$ is the lower credit bound of SR Class M_i,

$$c^{h,\min}_{M_i} = sdSl^h_{M_i} \cdot \frac{l^{h,\max}_{M_i}}{C}, \tag{11}$$

where $sdSl$ denotes the sendSlope.

$\alpha^h_{TAS}(t)$ in Eq. (9) is the arrival curve of TT traffic scheduled by GCL.

The arrival curve $\alpha(t)$ is for constraining the arrival process of the flow, satisfying

$$\mathcal{R}(t) \le (\mathcal{R} \otimes \alpha)(t). \tag{12}$$

A typical example of an arrival curve is the burst-rate form,

$$\alpha(t) = b + \rho \cdot t, \tag{13}$$

for $t > 0$ and 0 otherwise, with the parameters b as the maximum burst tolerance and ρ as the long-term rate of the flow.

For each ET flow f at its source ES h_0, the arrival curve can be modeled as,

$$\alpha_f^{h_0}(t) = b_f^{h_0} + \rho_f^{h_0} t, \tag{14}$$

where $b_f^{h_0} = l_f$, and $\rho_f^{h_0} = l_f / P_f$. The arrival curve of flow f at intermediate node h is the output arrival curve of f departing from the server h^-,

$$\alpha_f^h(t) = \alpha_f^{h^-} \oslash \delta_{D_f^{h^-}}(t), \tag{15}$$

where $D_f^{h^-}$ is the latency upper bound of flow f queuing at server h^-, and $\delta_D(t)$ is the pure-delay function.

The aggregate arrival curve for ET flows of SR Class M_i at h is obtained by summing the arrival curves of individual flows. It also incorporates the link shaping curve and the CBS shaping curve to improve the tightness of the analysis results.

$$\alpha_{M_i}^h(t) = \sum_{h^- \in \mathcal{H}} \sum_{f \in \mathcal{F}_{M_i}^{h^-,h}} \alpha_f^h(t) \wedge \sigma_{link}^{h^-,h}(t) \wedge \sigma_{M_i}^{h^-,h}(t), \tag{16}$$

where $x \wedge y = \min\{x, y\}$, $\sigma_{link}^{h^-,h}(t)$ is the link shaping curve from the preceding output h^- to the current output port h:

$$\sigma_{link}^{h^-,h}(t) = Ct + l_{M_i}^{h^-,h,\max}, \tag{17}$$

considering the packetization impact of the maximum frame size $l_{M_i}^{h^-,h,\max}$ of flows with Class M_i from h^- to h. $\sigma_{M_i}^{h^-,h}(t)$ is the CBS shaping curve of Class M_i from h^- to h:

$$\sigma_{M_i}^{h^-,h}(t) = idSl_{M_i}^{h^-} \left[t - \frac{\beta_{TAS}^{h^-}(t)}{C} + \frac{c_{M_i}^{h^-,\max} - c_{M_i}^{h^-,\min}}{idSl_{M_i}^{h^-}} \right] + l_{M_i}^{h^-,h,\max}, \tag{18}$$

$\beta_{TAS}^h(t)$ represents the minimum service supplied to TT traffic on the output port h.

With NC-based Total Flow Analysis (TFA), the worst-case delay upper bound D_f^h for flow $f \in \mathcal{F}_{M_i}^h$ at h equals the worst-case delay upper bound $D_{M_i}^h$ for all flows with the same priority M_i aggregating at h,

$$D_f^h = D_{M_i}^h = hDev(\alpha_{M_i}^h, \beta_{M_i}^h) = \sup_{t \geq 0} \left\{ \inf \left\{ \tau \geq 0 \mid \alpha_{M_i}^h(t) \leq \beta_{M_i}^h(t+\tau) \right\} \right\} \tag{19}$$

where $\alpha_{M_i}^h(t)$ is the arrival curve of aggregate flows of Class M_i from Eq. (16), and $\beta_{M_i}^h(t)$ is the service curve for Class M_i from Eq. (9). The upper bound of the worst-case end-to-end delay for the flow f is then obtained by summing the per-port latency bounds along its route.

D Credit-Based Shaper

Credit-Based Shaper (CBS) is a TSN mechanism designed to prevent starvation of lower-priority traffic while guaranteeing a reserved portion of bandwidth for higher-priority queues, thereby providing reliability through bounded end-to-end delays. Traffic assigned to queues using CBS is typically referred to as Audio Video Bridging (AVB) traffic.

Here, we build on the description from [Bujosa Mateu, 2024]. In CBS, each AVB queue is associated with a credit value. This credit increases over time when a frame is waiting to be transmitted or when the credit is negative, and decreases while a frame is being transmitted. Moreover, if the credit is positive and there are no AVB frames waiting to be transmitted, the credit is immediately reset to 0. The rates at which credit is increased and decreased are defined by the parameters *idleSlope* and *sendSlope*, respectively. Each queue implementing CBS is configured with its own *idleSlope*

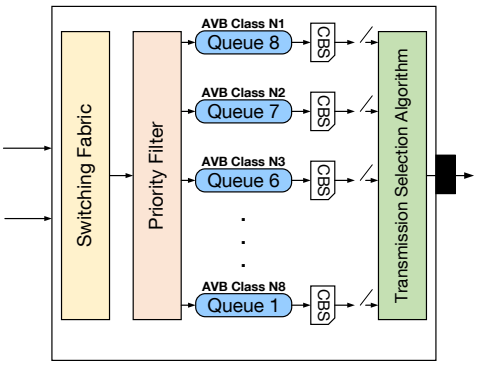


Figure 7: A simple CBS mechanism with eight queues in the egress port of the switch with different AVB classes mapped to different queues.

and *sendSlope* values, which determine its allocated bandwidth share. In particular, the bandwidth reserved for a queue is expressed as Eq. (20). A queue is eligible for transmission only when its credit is zero or positive.

$$\text{Reserved BW} = \frac{idleSlope}{idleSlope + sendSlope} \cdot BW \tag{20}$$

Consider the example illustrated in Figure 8, which includes two AVB queues and one Best Effort (BE) queue. Frames 1 and 4 are assigned to the higher-priority AVB queue, while frames 2 and 3 belong to the lower-priority AVB queue and the BE queue, respectively.

At time T0, both AVB queues are eligible for transmission. Due to strict priority scheduling, the higher-priority AVB queue (priority 2) is selected, and frame 1 is transmitted. During this transmission, its credit decreases, while the credit of the lower-priority AVB queue increases because it is waiting.

At time T1, the higher-priority AVB queue has accumulated negative credit and is therefore no longer eligible for transmission. As a result, the lower-priority AVB queue is selected, and frame 2 is transmitted, even though a higher-priority frame (frame 4) is waiting. During this time, the lower-priority queue's credit decreases, while the higher-priority queue's credit recovers.

By time T2, both AVB queues have negative credit, making them ineligible for transmission. Consequently, the BE queue is selected, and frame 3 is transmitted, despite the presence of a higher-priority AVB frame waiting.

Finally, at time T3, the credit of the higher-priority AVB queue has recovered to zero, making it eligible again. Therefore, frame 4 is transmitted.

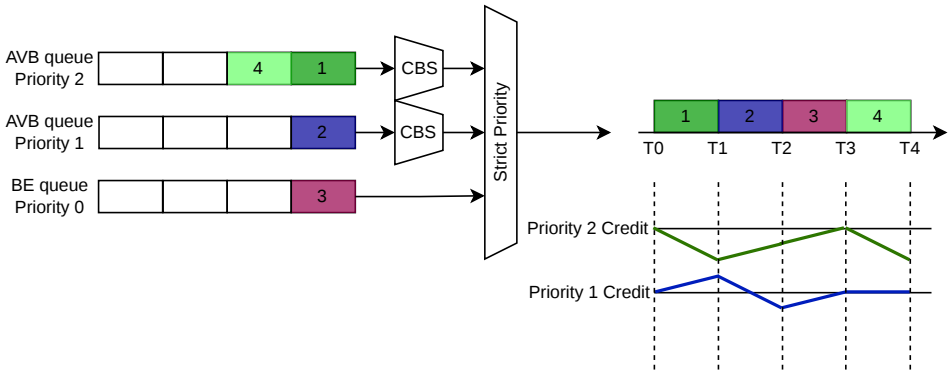


Figure 8: TSN output port with two AVB queues employing CBS and one BE queue.

E Cyclic Queuing and Forwarding (CQF)

Cyclic Queuing and Forwarding (CQF) [IEEE, 2017, Debnath et al., 2025b] is a TSN shaping mechanism which uses a single cycle duration, denoted as T, across the entire network. T is the minimum scheduling unit where we put the TSN flows. Furthermore, T defines the granularity of the end-to-end delay of the flows in the network. The unit of T is in μs in TSNBench. In a TSN switch, every egress port in the network has eight queues. TSN flows are stored in the queues depending on their priority. In CQF, for each egress port, two queues are used: an `even queue` and an `odd queue`. Figure 11 shows the basic working diagram of CQF with two queues (`even` and `odd`). As shown in Figure 11, CQF works by employing two queues, let's say, Q_8 and Q_7 for TT flows by operating them in a ping-pong manner where Q_7 receives and Q_8 transmits at the first cycle slot (T_1). During the second cycle slot (T_2), Q_8 receives and Q_7 transmits. Selecting or allotting a cycle slot for a flow means selecting the cycle slot number (within the hypercycle H) and the queue for the flow.

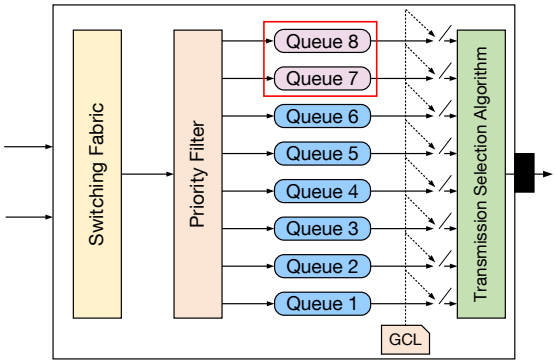


Figure 9: A simple CQF mechanism with eight queues in the egress port of the switch with two queues (Queue 8 and 7) operating as even and odd queue as shown in red.

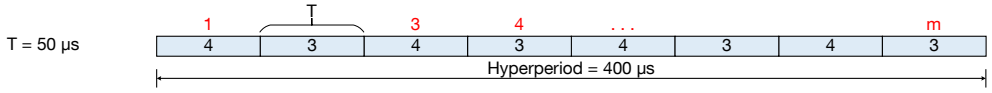


Figure 10: The Hypercycle also known as the scheduling cycle of the CQF (400 μs) with cycle duration (T) of 50μs. The different cycle slots are numbered as 1,2 $\cdots$ m in red.

In the CQF evaluation of TSNBench, we provide the cycle duration (T) and the network-specific delays to the model as input through the prompt in all prompt variants.

WCD CQF: The worst-case end-to-end delay of the TT flows in the CQF network is quantified as follows:

$$\mathrm{WCD} = f_i.\phi + (\mathrm{SW}_{\mathrm{num}} + 1) \cdot \mathrm{T} + \xi, \tag{21}$$

where $f_i.\phi$ is the offset of the flow f_i in μs, $\mathrm{SW}_{\mathrm{num}}$ is the total number of switches in the route of the TT flow, T is the cycle duration in μs, and ξ denotes the network-specific delays: propagation delay (d_{prop}), switching delay (d_{sw}), and time synchronization error ($\mathrm{sync}_{\mathrm{error}}$). In TSNBench, the switching delay d_{sw} is added once per switch, while the propagation delay and the time synchronization error are added once per flow:

$$\xi = (\mathrm{SW}_{\mathrm{num}} \cdot d_{\mathrm{sw}}) + d_{\mathrm{prop}} + \mathrm{sync}_{\mathrm{error}}. \tag{22}$$

F More on TSNBench

F.1 MCQA keyword extraction and question generation

Table 6 lists the models used in the dataset pipeline. Claude Sonnet 4 extracts the TSN keywords, and Claude Sonnet 4, GPT-4o mini, and Llama 3.1 70B generate the MCQs, with each source paper

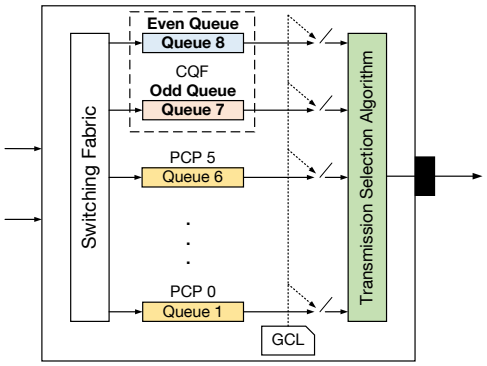


Figure 11: In this figure, we showcase the even and the odd queue in CQF architecture and during one cycle duration one queue receives the flows and another queue transmits the flows received in the previous cycle duration.

assigned to exactly one generator in a round-robin manner. All models are used with default settings and were last accessed in April 2026.

Table 6: Models used in the TSNBench keyword extraction and question generation pipeline. All models are used with default settings and last accessed in April 2026. Claude Sonnet 4 serves two distinct roles: keyword extraction and question generation. These roles use identical model configurations but operate on different inputs and prompts. Full dataset generation details are provided in Appendix F.

Model	API	Model ID	Organization	Usage
Claude Sonnet 4	Anthropic API	claude-sonnet-4-20250514	Anthropic	Keyword extractor
Claude Sonnet 4	Anthropic API	claude-sonnet-4-20250514	Anthropic	Generator
GPT-4o mini	OpenAI API	gpt-4o-mini	OpenAI	Generator
Llama 3.1 70B	HF Router	Llama-3.1-70B-Instruct	Meta	Generator

F.2 MCQ topic coverage by TSN mechanism

Our 83 source papers were selected using explicit inclusion criteria, described in Sec. 3.1. A key part of this selection process was restricting our sources to published or accepted papers that are openly available. We chose to avoid paywalled versions of publications from IEEE and other publishers out of respect for their copyright terms. While we recognize this narrows the pool of available source material, we believe it reflects a responsible and ethical approach to constructing the dataset.

On the risk of under-representation, we provide a topic-coverage breakdown of the MCQs against different TSN mechanisms. By default, most of the research papers are published for TAS and CBS as these are the oldest mechanisms, and this is reflected in how much open-access material exists for each.

We used a 5% representation threshold to check coverage, and all major TSN mechanisms are at or above this threshold. ATS sits almost exactly at the boundary (5.0%), which we attribute to its comparatively recent standardization and the correspondingly smaller published literature base available for source-paper selection (Sec. 3.1). We do not consider this a critical gap. Instead, we highlight ATS as the mechanism most in need of attention in future dataset expansion.

"General background" category (29.2%) consists of foundational TSN concepts, lexicon, keywords, and context spanning multiple mechanisms (TAS, CBS, ATS, CQF, GCL, reliability). We view this breadth as consistent with the benchmark's goal of assessing overall TSN domain knowledge, not solely mechanism-specific recall.

Table 7: Topic coverage of the MCQ dataset, categorized by mechanism.

Category	TSN Mechanism	Count	% of total	Comments
Latency, schedule table, and configuration	Time-Aware Shaper (TAS)	339	36.1%	Most researched mechanism
	Credit-Based Shaper (CBS)	94	10.0%	Oldest mechanism
	Asynchronous Traffic Shaper (ATS)	47	5.0%	Very recently standardized
	Cyclic Queuing and Forwarding (CQF)	54	5.8%	Recently started gaining attention from researchers for its added benefits.
WCD analysis	TAS CBS ATS CQF	81	8.6%	
Reliability & Resource Management	Frame replication and elimination Path control and reservation Per stream filtering and policing Stream reservation protocol	50	5.3%	
General background	TAS CBS ATS CQF Frame replication and elimination Path control and reservation Stream reservation protocol Time synchronization Legacy system	274	29.2%	General questions covering basic questions, keywords expansion covering all topics of TSN and legacy networks related to TSN.
Total		**939**	**100.0%**	

F.3 Human evaluation decision mechanism

To maintain the same standards across all human reviewers, we use the following rules to evaluate the MCQA dataset. There are four possible options for every question in the MCQA dataset.

1. **Accept:**
   i. Technically correct.
   ii. Clearly worded and self-contained.
   iii. Unambiguous options.
   iv. Accurate and sufficient explanation.
   v. The correct answer is actually the correct answer.
2. **Reject:**
   i. Incorrect or misleading.
   ii. Poorly constructed beyond revision.
   iii. Irrelevant to TSN.
   iv. Incomplete information.
   v. Too paper-dependent.
   vi. Duplicate questions.
3. **Revise:**
   i. Minor issues in grammar, clarity, or wording.
   ii. Options need improvement.
   iii. Explanation needs refinement.
4. **Doubtful:**
   i. Paper-specific or uncertain about the correctness of the question.

ii. Explanation seems questionable.

iii. Needs further clarification.

For a doubtful multiple-choice question, we read the research paper and re-evaluate the question. Afterward, the decision can be accept, reject, or revise; if it is still doubtful, we send it to another expert reviewer for a consensus-based group decision.

Key principles followed while reviewing the dataset: We ensured that the MCQAs were technically accurate and aligned with TSN fundamentals. We avoided tricky questions and preferred clarity over complexity. The same set of rules was given to all expert reviewers who worked on this dataset and served as human judges. After the review, 185 questions were revised by the domain experts, as shown below in Table 8.

Table 8: Statistical data of the MCQA dataset after domain expert human review.

Category	**Count**
Total questions revised by domain experts	185

F.4 Sample Questions

We present three representative sample questions from our MCQA dataset below.

Q1 *TSN Keyword*

What does TAS stand for in TSN traffic management?

A. Transmission Access Scheduler
B. Traffic Analysis System
C. Time-Aware Shaper
D. Traffic Admission Service

Correct Answer: **C**

Q2 *Research Paper*

In a Cyclic Queuing and Forwarding (CQF) network what fundamental limitation would prevent effective fault tolerance using Frame Replication and Elimination for Reliability (FRER) in a linear topology where each switch has maximum transmission unit (MTU) sized frames frequently queued?

A. CQF's ping-pong queue switching would create timing conflicts with FRER's frame elimination mechanism.
B. EMI interference would corrupt both original and replicated frames equally, making spatial redundancy ineffective.
C. FRER cannot detect bit errors caused by EMI since it lacks Cyclic Redundancy Check (CRC) verification capabilities.
D. Linear topologies cannot provide the disjoint paths required for FRER's spatial redundancy approach, forcing expensive hardware additions.

Correct Answer: **D**

Q3 *Research Paper*

What fundamental challenge makes the Time Aware Shaper (TAS) implementation complex despite its ability to provide guaranteed end-to-end delays?

A. The requirement to synchronize all network devices to a common time reference.

B. The need to maintain separate queues for each traffic class simultaneously.

C. The difficulty in estimating worst-case transmission times for variable-length frames.

D. The synthesis of the gate control list, which is an NP-hard problem.

Correct Answer: **D**

F.5 Prompt design of open-ended questions

CB prompt: For the CBS and CQF mechanisms, two different approaches are used for WCD calculation. NC is used to calculate the CBS WCD, whereas an analytical mathematical calculation is used to find the WCD for the CQF mechanism. Since these two mechanisms work differently, we design prompts tailored to each mechanism.

Role: We start by defining the role of the model: "You are an expert Time-Sensitive Networking (TSN) orchestrator." We inject three network inputs: (i) network topology, (ii) TSN flow information, and (iii) the routes of the flows. We use the prompt-as-program [Reynolds and McDonell, 2021] approach to separate the network topology, flow information, and flow routes. All of these are provided in text format. However, to evaluate different topologies, flows, and routes, we separate them from the prompt logic. This ensures that the prompt remains the same across different network topologies and parameters.

Constants: To correctly calculate the WCD, information about the network parameters is required. To prevent the model from assuming these values and to keep the constant values consistent across all models, we provide this information in the prompt.

Constants for CBS open-ended questions:

$Bandwidth = 100$ Mbps,

$Propagation\ delay = 1\,\mu$s,

$Switching\ delay = 1\,\mu$s,

$Time\ synchronization\ error = 1\,\mu$s,

The switches of the network are cut-through switches,

$IdleSlope = 75\%$

By controlling these network parameters, we directly mitigate hallucinations and assumptions about numerical values.

Architecture Restriction: TSN supports multiple architectures that affect the Quality of Service (QoS) and the WCD of the flows. The prompt restricts the model to using only one TSN mechanism through the following directive.

For the CBS mechanism, we use:

TSN Mechanism:

Only Credit-Based Shaper (CBS, IEEE 802.1Qav) is allowed;

All flows are AVB Class A, PCP = 6, using queue 6 only.

For the CQF mechanism, we use:

TSN Mechanism:

Only Cyclic Queuing and Forwarding (CQF, IEEE 802.1Qch) is allowed;

All flows are TT, PCP = 7, using queue 7 (odd) and 6 (even) only.

Our reasoning is that letting the model select the TSN architecture or mechanism is a separate benchmarking problem, where the model is evaluated on architecture design performance. In TSNBench, our goal is to benchmark LLMs in TSN. Without an explicit restriction, the model may select an incorrect or inappropriate mechanism, producing a hallucinated architecture that does not satisfy the QoS requirements of the flows. This restriction forces the model to use a single solution space. It further ensures that the WCDs provided by different models are not caused by architectural faults or mechanism selection ambiguity, but rather by calculation and implementation errors within the specified mechanism.

Structured Output: We instruct the model through the prompt to provide the output strictly in JSON format [Yang et al., 2026].

Formula-in-context (FC) prompt: The FC prompt is identical to the CB prompt above, except that the following description is provided in the prompt.

Network Calculus Equations (use these directly; do not derive alternative formulations):

Arrival curve, for flow `f` with frame size `l_f` and period `P_f` at its source node `h0`:
`alpha_f^h0(t) = b_f^h0 + rho_f^h0 * t`, for `t > 0`, else 0 where `b_f^h0 = l_f`, `rho_f^h0 = l_f / P_f`

Arrival curve propagation to an intermediate node h (output arrival curve of `f` departing server `h-`):
`alpha_f^h(t) = alpha_f^(h-)` (deconvolved with) `delta_{D_f^(h-)}(t)`
where `D_f^(h-)` is the latency upper bound of flow `f` at server `h-`, and `delta_D(t)` is the pure-delay function.

Aggregate arrival curve for all flows of priority class `M_i` at port `h`:
`alpha_{M_i}^h(t)` = sum over incoming ports `h-` and flows `f` in class `M_i` of `[ alpha_{M_i}^{h-,h}(t) AND link-shaping-curve AND CBS-shaping-curve ]`
where `AND` denotes `min(x, y)`.

Link shaping curve from `h-` to `h`:
`sigma_link^{h-,h}(t) = C*t + l_{M_i}^{h-,h,max}`
where `C` is the link rate and `l_{M_i}^{h-,h,max}` is the max frame size of class `M_i` flows from `h-` to `h`.

CBS shaping curve of class `M_i` from `h-` to `h`:
`sigma_{M_i}^{h-,h}(t) = idleSlope_{M_i} * t + (c_{M_i}^{h-,max} - c_{M_i}^{h-,min}) + l_{M_i}^{h-,h,max}`

Credit upper bound for class `M_i` at port `h`:
`c_{M_i}^{h,max} = idleSlope_{M_i}^h * ( sum_{j<i} c_{M_j}^{h,min} - l_{>i}^{h,max} ) / ( sum_{j<i} idleSlope_{M_j}^h - C )`
where `l_{>i}^{h,max}` is the max frame size of any class with priority lower than `M_i` (or best-effort) at `h`, `j<i` means the AVB class `j` with priority higher than `i`.

Credit lower bound for class `M_i` at port `h`:
`c_{M_i}^{h,min} = sendSlope_{M_i}^h * l_{M_i}^{h,max} / C`
where `sendSlope_{M_i}^h=idleSlope_{M_i}^h-C`

CBS service curve (rate-latency form) for class `M_i` at port `h`:
`beta_{M_i}^h(t) = idleSlope_{M_i}^h * [ t - c_{M_i}^{h,max} / idleSlope_{M_i}^h ]^+_{uparrow}`
where `[f(t)]^+_{uparrow}=max_{0<=s<=t}\{f(s), 0\}`.

Worst-case per-hop delay at port `h` (horizontal deviation, NC Total Flow Analysis): `D_f^h = D_{M_i}^h = hDev(alpha_{M_i}^h, beta_{M_i}^h) = sup_{t>=0} { inf { tau >= 0 : alpha_{M_i}^h(t) <= beta_{M_i}^h(t + tau)}}`

End-to-end WCD for flow `f`:
`WCD_f = sum of D_f^h` over every port `h` along `f`'s route (as given in Path input),
plus propagation delay, and switching delay per hop as specified in Constants.

For CQF, we provide the following description:

CQF worst case delay (WCD) equation (use these directly; do not derive alternative formulations):

`Hypercycle = LCM(all flow periods)`

WCD of flow number `i` is:

`WCD_f_i = offset_i + (n + 1) * cycle duration + n * switching delay + propagation delay + time synchronization error`

`offset_i` is the offset (start time) of flow `i` at its source node.

`n` is the number of switches in the route of flow `i` (`n = hop count - 1`).

Propagation delay and time synchronization error are added once per flow.

Prompt variants: For the prompt-sensitivity study (Section 4.2), Variant A paraphrases the CB prompt, and Variant B replaces the JSON output instruction and schema with a fixed plain-text format. All inputs, constants, and the task itself are unchanged. The full text of the CB, FC, Variant A, and Variant B prompts for both mechanisms is available in the repository at https://github.com/tum-esi/TSNBench/tree/main/open_end_evaluation and in the dataset on Kaggle at https://doi.org/10.34740/kaggle/ds/13019279.

F.6 TSNBench Open-Ended Question Details

For the open-ended questions, there are three variable entries: network topology, flow information, and flow routing. We use the K-shortest path algorithm to determine the routes of the flows. The routes are then directly provided to the models as input for further evaluation.

Network Topologies Used: For the open-ended questions, we selected three different topologies to evaluate the models: a one-switch topology, a medium-mesh topology, and an industrial ring topology. Figures 12, 13, and 14 represent the one-switch, medium-mesh, and ring topologies used in TSNBench, respectively.

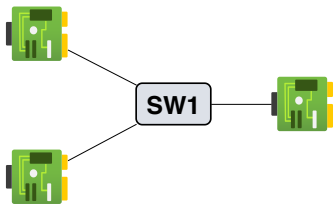


Figure 12: One-switch topology used to evaluate open-ended questions in TSNBench.

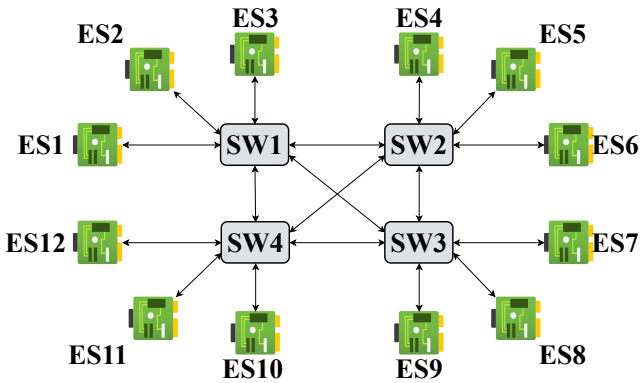


Figure 13: Medium-mesh topology used to evaluate open-ended questions in TSNBench.

Flow parameters: We show the flow information used in TSNBench as follows.

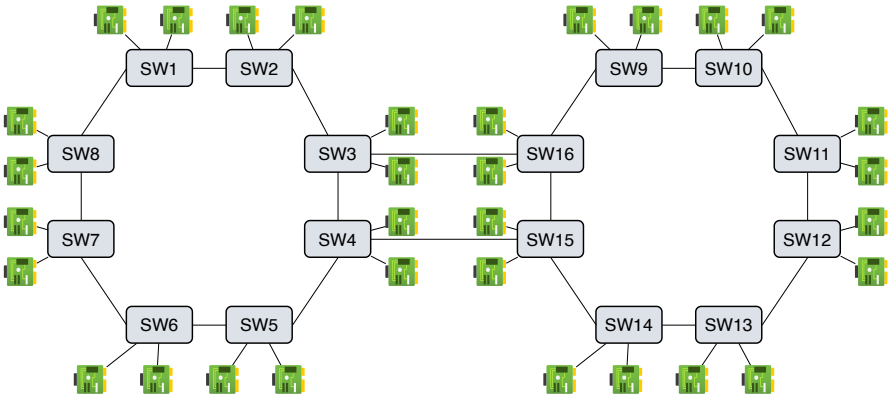


Figure 14: Ring topology representing the industrial ring network used to evaluate open-ended questions in TSNBench.

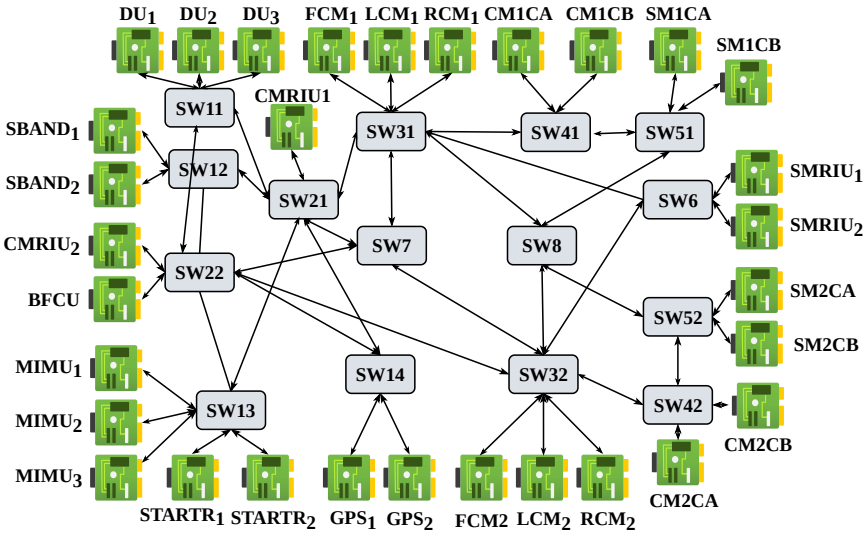


Figure 15: Orion Crew Exploration Vehicle (CEV) topology representing the realistic network topology used to evaluate open-ended questions in TSNBench.

Flow Information *TC1_flows.txt*

```
0,node2_1,node5_2,2500,709,965
1,node5_4,node3_2,2500,610,825
2,node0_4,node0_1,1000,786,887
3,node2_3,node4_3,2500,1088,1233
4,node0_4,node3_3,1000,1015,488
5,node0_4,node0_1,2500,926,501
...
```

Ground Truth WCD Values The ground-truth WCD values of the flows for all open-ended test cases for the CBS mechanism are calculated using a verified NC tool [Zhao et al., 2018, Debnath et al., 2025c, Gavriluţ and Pop, 2020]. For the WCD of the CQF mechanism, we use the mathematical equation given in Eq. 21.

G More on TSNBench Evaluation

We evaluate both open-source and closed-source state-of-the-art LLMs on TSNBench. A detailed list of the models, along with their model IDs and snapshots, is given in Table 9. This ensures that the results are reproducible by the community.

Table 9: Details of the models used for the benchmarking on TSN. Both MCQA and open-ended questions are evaluated on these models. We provide the specific model ID and snapshot for reproducibility.

Chat Models				
Model	Family	Model ID	Organization	Country
Grok 4.1 Fast (Non-Reasoning)	Grok	grok-4-1-fast-non-reasoning	xAI	USA
DeepSeek-V3.2 (Non-Thinking Mode)	DeepSeek	deepseek-chat	DeepSeek AI	China
GPT-4o	GPT	gpt-4o-2024-08-06	OpenAI	USA
GPT-4o mini	GPT	gpt-4o-mini-2024-07-18	OpenAI	USA
Llama 3.3 70B	Llama	Llama-3.3-70B-Instruct	Meta (via HF)	USA
Mistral Medium 3.1	Mistral	mistral-medium-2508	Mistral AI	France
Mistral Large 3	Mistral	mistral-large-2512	Mistral AI	France
Claude Sonnet 4.5	Claude	claude-sonnet-4-5-20250929	Anthropic	USA
Reasoning/Thinking Models				
Grok 4.1 Fast	Grok	grok-4-1-fast-reasoning	xAI	USA
o3	GPT	o3-2025-04-16	OpenAI	USA
GPT-5	GPT	gpt-5-2025-08-07	OpenAI	USA
DeepSeek-V3.2 (Thinking Mode)	DeepSeek	deepseek-reasoner	DeepSeek AI	China
Gemini 2.5 Flash	Gemini	gemini-2.5-flash	Google	USA
Small Models				
Llama 3.2 1B	Llama	llama-3.2-1B	Meta (via HF)	USA
Qwen3 8B	Qwen	Qwen3-8B	Alibaba Cloud	China
Ministral 3 8B	Ministral	ministral-8b-2512	Mistral AI	France

Table 10: Extended evaluation results of TSNBench MCQA dataset across different state-of-the-art models across different families. We provide the accuracy in percentage under different temperature settings (default, set to 0.0, and set to 0.7). The consistency shows the performance of the model in providing the same response across three runs. For those models which do not support temperature = 0.0, we use their default temperature and this is marked next to the model in the table.

Model	**MCQA Accuracy (%)**			**Average Consistency**		
	Default Temp.	**Temp=0.0**	**Temp=0.7**	**Default Temp.**	**Temp=0.0**	**Temp=0.7**
Grok 4.1 Fast†	**93.2**	–	–	0.99	–	–
Grok 4.1 Fast (Non-Reasoning)	–	91.7	91.6	–	1.00	1.00
DeepSeek-V3.2 (Non-Thinking)	–	94.0	93.4	–	1.00	0.98
GPT-4o	–	91.8	92.1	–	1.00	0.98
GPT-4o mini	–	88.3	88.2	–	0.99	0.98
Llama 3.3 70B	–	88.9	89.1	–	0.99	0.99
Mistral Medium 3.1	–	92.1	92.3	–	1.00	0.99
Mistral Large 3	–	92.8	92.9	–	1.00	1.00
Claude Sonnet 4.5	–	**95.3**	**95.3**	–	1.00	1.00
o3†	94.7	–	–	**0.98**	–	–
GPT-5†	**95.0**	–	–	0.99	–	–
DeepSeek-V3.2 (Thinking)†	94.7	–	–	**0.98**	–	–
Gemini 2.5 Flash	–	90.1	90.8	–	**0.98**	0.97
Llama 3.2 1B	–	**67.4**	**67.0**	–	1.00	**0.93**
Qwen3 8B	–	83.7	82.8	–	0.99	0.97
Ministral 3 8B	–	86.9	86.5	–	1.00	0.97

† Temperature parameter not supported. Evaluated with default settings.

G.1 Extended MCQA Experimental Evaluation

We evaluate the models on the MCQA dataset under two different configurations: (i) temperature set to 0.7 and (ii) temperature set to 0.0 (or default temperature settings). The open-ended questions

are evaluated at temperature 0.0 (or default temperature settings). As TSN is used in safety-critical networks, we want to ensure deterministic results, so we report results at temperature 0.0 wherever the model supports it. For models that do not support the temperature parameter, we use the default temperature for evaluation.

Table 10 provides the accuracy and average consistency of the models for the MCQA dataset under the default temperature, temperature set to 0.0, and temperature set to 0.7. Average consistency represents the ability of the model to provide the same results across three runs.

G.2 MCQA and Open-Ended Cost and Latency

The cost and latency of a model are important evaluation parameters for the research community. Spending a large amount of money on benchmark evaluation is a real bottleneck for research groups. Moreover, not all models can be evaluated locally. Table 11 presents the cost and latency of the TSNBench MCQA and open-ended questions. Evaluating MCQA is relatively much cheaper than evaluating open-ended questions.

Table 11: Extended results of cost and latency comparison for MCQA and CB open-ended evaluation in TSNBench. "–" indicates that cost and latency are not reported for this model, as it successfully evaluated fewer than 50 out of 100 TCs, where a TC is considered successfully evaluated only if the model provided WCD estimates for at least 80% of the flows within that TC.

Model	MCQA		CBS Open-ended questions		CQF Open-ended questions	
	Cost (USD)	Latency (ms)	Cost (USD)	Latency (ms)	Cost (USD)	Latency (ms)
Grok 4.1 Fast†	0.2490	18,769,322	–	–	0.3047	49,367,058
Grok 4.1 Fast (Non-Reasoning)	0.2612	1,450,175	0.3256	3,200,483	0.3251	3,383,209
DeepSeek-V3.2 (Non-Thinking)	**0.0420**	2,264,129	–	–	0.4816	13,211,941
GPT-4o	2.3661	2,053,601	–	–	4.3866	**2,122,167**
GPT-4o mini	0.1417	2,251,786	0.3438	7,083,613	0.3642	7,092,033
Llama 3.3 70B	0.5224	1,028,334	0.7399	**1,264,847**	0.7137	1,136,920
Mistral Medium 3.1	0.4432	1,839,587	1.9918	6,335,803	2.2442	6,658,056
Mistral Large 3	0.4868	15,487,495	1.5989	11,149,853	1.2298	8,226,564
Claude Sonnet 4.5	3.5967	5,190,222	21.1719	14,342,470	18.3093	12,583,398
o3†	7.6293	10,831,712	10.9954	10,127,407	10.0134	8,591,382
GPT-5†	**12.8766**	15,860,682	**57.5232**	**74,473,434**	**42.4470**	**58,404,966**
DeepSeek-V3.2 (Thinking)†	0.4069	12,385,365	–	–	–	–
Gemini 2.5 Flash	0.4164	18,732,689	2.6471	24,160,941	2.7690	17,688,716
Llama 3.2 1B	0.0864	1,883,306	–	–	–	–
Qwen3 8B	0.3736	**42,272,121**	–	–	–	–
Ministral 3 8B	0.1237	**972,292**	**0.2425**	10,122,749	**0.2434**	9,603,062

† Temperature parameter not supported. Evaluated with default settings.

G.3 MCQA Reliability Analysis

Figure 16 presents the reliability plot for all 16 evaluated models on the MCQA dataset.

Each diagram shows the observed accuracy against the model's expressed confidence, binned across the confidence range. A perfectly calibrated model would fall on the gray dashed diagonal line. This means the model's confidence would perfectly align with its actual accuracy. The red shaded region indicates overconfidence, meaning the model's confidence exceeds its actual accuracy. The green shaded region indicates underconfidence, meaning the model is more accurate than its expressed confidence suggests.

In safety-critical TSN deployments, overconfidence is significantly more dangerous than underconfidence. A model that is incorrect but expresses high confidence may mislead a network engineer with an erroneous WCD estimate or misconfigured scheduling parameters. By contrast, an underconfident model that expresses uncertainty on correct answers prompts additional verification.

The majority of the evaluated models sit in the high-confidence region (0.8 to 1.0) regardless of their actual accuracy. This indicates that the models tend to exhibit overconfidence.

Grok 4.1 Fast (NR), Llama 3.3 70B, Mistral Medium 3.1, Mistral Large 3, Qwen3 8B, and Ministral 3 8B achieve CW rates of 100%, meaning all incorrect answers fall in the high-confidence range. This represents the most critical calibration behavior for TSN deployment. Grok 4.1 Fast, GPT-4o,



Figure 16: Reliability diagram representing the performance of all 16 state-of-the-art models evaluated on the MCQA dataset in TSNBench. The gray dashed line represents the perfect calibration where confidence is equal to the accuracy. A model which is 100% confident and has 100% accuracy will fall on this gray dashed line. The red shaded region represents the over-confidence of the model (model confidence exceeds the actual accuracy of the model), and the green shaded region represents the under-confidence of the model (actual accuracy of the model exceeds its given confidence).

Gemini 2.5 Flash, and Llama 3.2 1B similarly exhibit CW rates exceeding 95%. A notable exception is o3, which is the only model that falls predominantly in the green underconfident zone, with a CW rate of just 3.4%. Despite having the highest ECE (0.1951) among all evaluated models, o3 is the safest model from a calibration perspective, as it rarely expresses high confidence on incorrect MCQA answers. This highlights an important distinction between aggregate calibration metrics and safety-relevant calibration behavior. DeepSeek-V3.2 (NT) achieves the lowest ECE (0.0104), suggesting strong overall calibration, yet maintains a CW rate of 96.4%, demonstrating that a low ECE does not guarantee safe and realistic confidence behavior.

G.4 More on TSNBench Open-Ended Questions Evaluation

We provide MAE evaluations for the open-ended questions. A sample calculation is given as follows:

MAE calculation example: Consider a model evaluated on three test cases (TCs). These three TCs may have different topologies, different flows and flow parameters, and different routes. Table 12 shows the GT WCD, the predicted WCD and the absolute error of each flow. As in the main evaluation, the predicted WCD of a flow is the average of its predictions over the valid runs of the TC.

Table 12: Sample example of TCs with the GT, the predicted WCD (averaged over the valid runs) and the absolute error of each flow.

TC	Flow	Ground Truth (μs)	Predicted (μs)	Abs. Error (μs)
TC1	f_0	200	212	12
TC1	f_1	150	180	30
TC1	f_2	500	490	10
TC2	f_0	100	108	8
TC2	f_1	300	255	45
TC3	f_0	400	420	20
TC3	f_1	250	265	15
TC3	f_2	600	600	0

Per-TC MAE: Suppose TC1, TC2, and TC3 contain three, two, and three flows, respectively.

$$\{f_0, f_1, f_2\} \subseteq TC1;$$
$$\{f_0, f_1\} \subseteq TC2;$$
$$\{f_0, f_1, f_2\} \subseteq TC3;$$

Let $\Gamma(f_0)$ denote the absolute error of flow f_0 in TC1, $\beta(f_0)$ denote the predicted WCD of flow f_0 given by the LLM model, and $\Omega(f_0)$ denote the GT of flow f_0. We calculate $\Gamma(f_0)$ as follows:

$$\Gamma(f_0) = |\beta(f_0) - \Omega(f_0)|$$

In the given example, let $\Gamma(f_0)$ = 12, $\Gamma(f_1)$ = 30, and $\Gamma(f_2)$ = 10 for TC1. Similarly, for TC2, $\Gamma(f_0)$ = 8 and $\Gamma(f_1)$ = 45 and for TC3, $\Gamma(f_0)$ = 20, $\Gamma(f_1)$ = 15, and $\Gamma(f_2)$ = 0. We calculate the MAE for TC1, TC2, and TC3 represented as MAE_{TC1}, MAE_{TC2}, and MAE_{TC3} as follows:

$$\text{MAE}_{\text{TC1}} = (12 + 30 + 10)/3 = 17.3\ \mu\text{s}$$
$$\text{MAE}_{\text{TC2}} = (8 + 45)/2 = 26.5\ \mu\text{s}$$
$$\text{MAE}_{\text{TC3}} = (20 + 15 + 0)/3 = 11.7\ \mu\text{s}$$

For every model, we have 100 TCs, and the final MAE is averaged across all valid TCs (in this example 3 TCs) and is represented as:

$$\text{MAE} = (17.3 + 26.5 + 11.7)/3 = 18.5\ \mu\text{s}$$

The reported standard deviation is the population standard deviation of the per-TC MAE across the valid TCs (6.1 μs in this example), and the median MAE is the median of the per-TC MAE (17.3 μs). A TC is valid only if at least 80% of its flows receive a prediction, and a model is reported only if it has at least 50 valid TCs.

In TSNBench, every TC contributes equally towards the model performance irrespective of the number of flows in the network. As per the network architecture, all flows are equally critical and need the same preference. This ensures that for each network scenario all the flows are weighted equally.

G.5 Changes from the submitted version

Following the reviews, we made the following changes to the open-ended evaluation. First, model outputs were re-extracted with a stricter parser that reads the final WCD answer of each response. When a response contained malformed JSON, the earlier parser could read numbers from the model's reasoning, such as CBS credit values or frame sizes, instead of its answer. A calculation without a final result, a value in a unit other than μs, and values above 10^9 μs now count as missing. Second, flow identifiers are matched to the GT regardless of their format (e.g., "15" and "F15"), and a run that uses a flow identifier that does not exist in the TC is discarded. Third, each model is scored at one temperature (0.0, or the vendor default for models without temperature support) instead of pooling all temperatures. Fourth, MAPE is removed because it is unstable for small GT delays, and the median MAE and error-threshold rates are added. The MAE formula is unchanged and all values

are recomputed. Models that return clean JSON with numeric values and the given flow identifiers are unaffected by the parser change, while the largest change is for Ministral 3 8B, whose earlier CBS error was dominated by extraction errors. Grok 4.1 Fast now has fewer than 50 valid CBS TCs and is no longer reported for CBS. Finally, we add the formula-in-context (FC) setting, a prompt-sensitivity study and a case study on the Orion CEV network topology.

Table 13: MAE (μs) $\pm$ standard deviation and median MAE (μs) across the valid TCs for CBS under CB and FC settings ($\downarrow$ lower is better). n is the number of valid TCs out of 100, using one temperature per model. Models with fewer than 50 valid TCs are reported as "–".

	CB			**FC**		
Model	MAE $\pm$ std	Median	n	MAE $\pm$ std	Median	n
Grok 4.1 Fast†,¶	–	–	48	–	–	–
Grok 4.1 Fast (Non-Reasoning)¶	3637.3 ± 3934.5	2992.5	91	–	–	–
DeepSeek-V3.2 (Non-Thinking)¶	–	–	28	–	–	–
GPT-4o*	–	–	49	–	–	17
GPT-4o mini	378.5 ± 189.7	337.7	89	461.6 ± 999.5	331.7	89
Llama 3.3 70B	301.2 ± 171.0	264.1	73	333.5 ± 171.3	308.5	63
Mistral Medium 3.1	339.6 ± 224.9	260.3	100	374.2 ± 232.7	315.7	100
Mistral Large 3	270.6 ± 180.2	231.2	100	273.5 ± 179.9	229.0	100
Claude Sonnet 4.5	293.0 ± 173.9	264.4	100	307.7 ± 178.3	269.2	100
o3†	255.8 ± 315.4	138.8	100	501.9 ± 2341.3	187.4	100
GPT-5†	150.2 ± 198.2	92.4	99	161.4 ± 188.0	82.5	99
DeepSeek-V3.2 (Thinking)†,¶	–	–	0	–	–	–
Gemini 2.5 Flash	554.6 ± 1831.6	223.2	84	258.0 ± 231.2	180.4	91
Llama 3.2 1B	–	–	0	–	–	0
Qwen3 8B	–	–	24	51855.3 ± 460034.8	359.8	97
Ministral 3 8B	1643.3 ± 2618.4	853.7	100	1551.7 ± 2735.6	374.8	100

n is the number of valid TCs out of 100, using one temperature per model.
"–" in the MAE columns: fewer than 50 valid TCs.
$n = 0$: evaluated, but no TC is valid.
"–" in the n column: not evaluated in this setting.
†, * and ¶ are explained in Table 3. The failure modes of models with few or no valid TCs are described in Appendix H.

Table 14: MAE (μs) $\pm$ standard deviation and median MAE (μs) across the valid TCs for CQF under CB and FC settings ($\downarrow$ lower is better). n is the number of valid TCs out of 100, using one temperature per model. Models with fewer than 50 valid TCs are reported as "–".

	CB			FC		
Model	MAE $\pm$ std	Median	n	MAE $\pm$ std	Median	n
Grok 4.1 Fast†,¶	130.1 ± 85.0	127.9	67	–	–	–
Grok 4.1 Fast (Non-Reasoning)¶	163.9 ± 37.8	167.7	99	–	–	–
DeepSeek-V3.2 (Non-Thinking)¶	168.1 ± 66.3	173.3	97	–	–	–
GPT-4o*	82.2 ± 264.7	1.2	100	11.3 ± 15.8	2.4	100
GPT-4o mini	180.5 ± 82.0	175.3	100	1377.1 ± 2818.7	323.4	88
Llama 3.3 70B	173.3 ± 73.5	173.7	93	5.1 ± 15.0	0.0	100
Mistral Medium 3.1	150.3 ± 42.2	141.0	100	33.0 ± 31.3	31.5	100
Mistral Large 3	54.2 ± 28.2	50.0	100	19.3 ± 17.8	15.7	100
Claude Sonnet 4.5	213.2 ± 1062.5	65.8	99	30.3 ± 37.2	16.0	100
o3†	101.6 ± 75.9	81.1	100	0.0 ± 0.0	0.0	100
GPT-5†	107.0 ± 69.0	107.0	99	0.0 ± 0.0	0.0	100
DeepSeek-V3.2 (Thinking)†,¶	–	–	1	–	–	–
Gemini 2.5 Flash	112.0 ± 90.1	91.7	99	0.1 ± 0.5	0.0	99
Llama 3.2 1B	–	–	0	–	–	0
Qwen3 8B	–	–	0	2.0 ± 6.2	0.0	57
Ministral 3 8B	3057.4 ± 4112.4	1046.0	99	1354.6 ± 2166.2	776.2	100

n is the number of valid TCs out of 100, using one temperature per model.
"–" in the MAE columns: fewer than 50 valid TCs.
$n = 0$: evaluated, but no TC is valid.
"–" in the n column: not evaluated in this setting.
†, * and ¶ are explained in Table 3. The failure modes of models with few or no valid TCs are described in Appendix H.

Table 15: Percentage of flows whose absolute WCD error falls below 50, 100, 500, and 1000 μs in the FC setting for CBS and CQF. For each model, the percentage is computed per TC and averaged over all valid TCs.

Model	**CBS (%)**				**CQF (%)**			
	$<50\mu$**s**	$<100\mu$**s**	$<500\mu$**s**	$<1000\mu$**s**	$<50\mu$**s**	$<100\mu$**s**	$<500\mu$**s**	$<1000\mu$**s**
GPT-4o*	–	–	–	–	93.8	100.0	100.0	100.0
GPT-4o mini	0.9	8.8	75.8	96.4	8.5	20.4	65.0	82.5
Llama 3.3 70B	3.9	15.6	78.9	98.2	96.7	99.1	100.0	100.0
Mistral Medium 3.1	2.2	10.1	75.5	96.7	84.7	96.6	100.0	100.0
Mistral Large 3	13.5	25.0	87.9	98.0	89.3	100.0	100.0	100.0
Claude Sonnet 4.5	5.8	16.9	84.4	97.9	84.2	94.3	100.0	100.0
o3†	15.5	26.9	83.1	95.8	100.0	100.0	100.0	100.0
GPT-5†	37.4	61.9	93.0	97.1	100.0	100.0	100.0	100.0
Gemini 2.5 Flash	18.0	34.0	85.2	96.3	100.0	100.0	100.0	100.0
Qwen3 8B	6.7	15.2	65.1	82.2	98.2	99.0	100.0	100.0
Ministral 3 8B	4.7	11.3	59.9	76.9	1.6	5.9	31.6	77.8

"–", †, * and the models not shown are explained in Table 3. All thresholds are in μs.

Table 16: Sensitivity of the CB MAE (μs) for CBS to prompt wording and output format (100 TCs). CB is the original prompt of the main results, Variant A paraphrases the instructions and Variant B requests plain-text instead of JSON output. MAEs are computed per model on the TCs that are valid under all three prompts (n). Range, Std and CV (std/mean) are taken across the three prompts. Lower ↓ CV indicates a more stable model.

Model	CB	Variant A	Variant B	Range	Std	CV (%) ↓	n
GPT-4o	–	–	–	–	–	–	49
GPT-4o mini	371.4	357.9	117259.2	116901.3	55104.6	140.1	88
Llama 3.3 70B	278.3	268.9	768.3	499.4	233.2	53.2	60
Mistral Medium 3.1	339.6	384.6	499.8	160.2	67.5	16.5	100
Mistral Large 3	270.6	266.5	617.5	350.9	164.5	42.7	100
Claude Sonnet 4.5	293.0	291.0	280.0	13.0	5.7	2.0	100
o3†	255.8	253.6	248.8	7.0	2.9	1.2	100
GPT-5†	142.4	121.6	122.5	20.8	9.6	7.4	93
Gemini 2.5 Flash	570.2	5435.5	388.6	5046.9	2337.5	109.7	73
Llama 3.2 1B	–	–	–	–	–	–	0
Qwen3 8B	–	–	–	–	–	–	8
Ministral 3 8B	1621.9	1056.1	7315.5	6259.5	2826.8	84.9	97

† Temperature parameter not supported. Evaluated with default settings.
"–" Not reported: fewer than 50 valid TCs.
$n = 0$: evaluated, but no TC is valid (Llama 3.2 1B).
Not evaluated (retired before the prompt-sensitivity experiments): DeepSeek-V3.2 (Non-Thinking), DeepSeek-V3.2 (Thinking), Grok 4.1 Fast, Grok 4.1 Fast (Non-Reasoning).

Table 17: Sensitivity of the CB MAE (μs) for CQF to prompt wording and output format (100 TCs). CB is the original prompt of the main results, Variant A paraphrases the instructions and Variant B requests plain-text instead of JSON output. MAEs are computed per model on the TCs that are valid under all three prompts (n). Range, Std and CV (std/mean) are taken across the three prompts. Lower ↓ CV indicates a more stable model.

Model	CB	Variant A	Variant B	Range	Std	CV (%) ↓	n
GPT-4o	85.3	35.5	48.6	49.8	21.1	37.3	96
GPT-4o mini	188.9	197.2	442.7	253.8	117.7	42.6	92
Llama 3.3 70B	172.4	154.3	77.7	94.6	41.0	30.4	91
Mistral Medium 3.1	149.8	172.6	1952.9	1803.1	844.7	111.4	99
Mistral Large 3	54.2	37.6	95.3	57.6	24.2	38.8	100
Claude Sonnet 4.5	213.2	135.1	65.4	147.8	60.4	43.8	99
o3†	101.6	84.9	89.4	16.7	7.1	7.7	100
GPT-5†	107.0	74.6	101.5	32.3	14.1	15.0	99
Gemini 2.5 Flash	108.4	124.6	122.6	16.1	7.2	6.1	98
Llama 3.2 1B	–	–	–	–	–	–	0
Qwen3 8B	–	–	–	–	–	–	0
Ministral 3 8B	3057.4	260.0	995.2	2797.4	1184.1	82.4	99

† Temperature parameter not supported. Evaluated with default settings.
"–" Not reported: fewer than 50 valid TCs.
$n = 0$: no TC is valid under all three prompts (Llama 3.2 1B, Qwen3 8B).
Not evaluated (retired before the prompt-sensitivity experiments): DeepSeek-V3.2 (Non-Thinking), DeepSeek-V3.2 (Thinking), Grok 4.1 Fast, Grok 4.1 Fast (Non-Reasoning).

Table 18: MAE (μs) $\pm$ standard deviation and median MAE (μs) across the valid TCs for CBS under the CB (original), paraphrased (Variant A) and format-changed (Variant B) prompts ($\downarrow$ lower is better). n is the number of valid TCs out of 100, using one temperature per model.

	CB			Variant A			Variant B		
Model	MAE $\pm$ std	Median	n	MAE $\pm$ std	Median	n	MAE $\pm$ std	Median	n
GPT-4o	–	–	49	312.3 ± 263.4	258.6	98	755.0 ± 2178.2	292.1	97
GPT-4o mini	378.5 ± 189.7	337.7	89	352.8 ± 174.2	329.0	99	103364.3 ± 512945.1	1414.3	100
Llama 3.3 70B	301.2 ± 171.0	264.1	73	285.4 ± 186.3	271.6	91	756.6 ± 1343.0	351.7	90
Mistral Medium 3.1	339.6 ± 224.9	260.3	100	384.6 ± 276.3	303.5	100	499.8 ± 908.4	263.0	100
Mistral Large 3	270.6 ± 180.2	231.2	100	266.5 ± 181.3	225.2	100	617.5 ± 926.6	294.8	100
Claude Sonnet 4.5	293.0 ± 173.9	264.4	100	291.0 ± 169.2	259.5	100	280.0 ± 378.1	215.9	100
o3†	255.8 ± 315.4	138.8	100	253.6 ± 276.0	169.0	100	248.8 ± 263.5	169.1	100
GPT-5†	150.2 ± 198.2	92.4	99	129.1 ± 131.5	89.7	95	125.0 ± 154.6	77.0	99
Gemini 2.5 Flash	554.6 ± 1831.6	223.2	84	4318.4 ± 36981.7	250.7	94	375.5 ± 415.7	213.4	91
Llama 3.2 1B	–	–	0	–	–	0	–	–	0
Qwen3 8B	–	–	24	–	–	34	84655.7 ± 606649.4	370.7	55
Ministral 3 8B	1643.3 ± 2618.4	853.7	100	1083.2 ± 1733.6	454.1	100	7315.5 ± 33679.4	1873.7	97

"–" Not reported: fewer than 50 valid TCs.
$n = 0$: evaluated, but no TC is valid (Llama 3.2 1B).
† Temperature parameter not supported. Evaluated with default settings.
Not evaluated (retired before the prompt-sensitivity experiments): Grok 4.1 Fast, Grok 4.1 Fast (Non-Reasoning), DeepSeek-V3.2 (Non-Thinking), DeepSeek-V3.2 (Thinking).
Gemini 2.5 Flash's Variant A mean is dominated by TC72 (MAE 360,883.9 μs), where all three runs apply an incorrect CBS delay formula whose denominator is close to zero and report a WCD of 2,460,205.52 μs for one flow.

Table 19: MAE (μs) $\pm$ standard deviation and median MAE (μs) across the valid TCs for CQF under the CB (original), paraphrased (Variant A), and format-changed (Variant B) prompts ($\downarrow$ lower is better). n is the number of valid TCs out of 100, using one temperature per model.

	CB			Variant A			Variant B		
Model	MAE $\pm$ std	Median	n	MAE $\pm$ std	Median	n	MAE $\pm$ std	Median	n
GPT-4o	82.2 ± 264.7	1.2	100	34.6 ± 151.9	1.0	100	48.6 ± 46.5	34.7	96
GPT-4o mini	180.5 ± 82.0	175.3	100	197.2 ± 113.1	177.3	92	491.8 ± 1142.1	165.6	100
Llama 3.3 70B	173.3 ± 73.5	173.7	93	159.1 ± 59.8	160.8	100	83.2 ± 121.6	40.9	98
Mistral Medium 3.1	150.3 ± 42.2	141.0	100	173.1 ± 56.0	173.3	100	1952.9 ± 11090.1	84.3	99
Mistral Large 3	54.2 ± 28.2	50.0	100	37.6 ± 33.9	33.3	100	95.3 ± 30.4	98.3	100
Claude Sonnet 4.5	213.2 ± 1062.5	65.8	99	135.4 ± 57.4	143.1	100	65.1 ± 46.8	52.3	100
o3†	101.6 ± 75.9	81.1	100	84.9 ± 46.2	76.1	100	89.4 ± 51.0	80.8	100
GPT-5†	107.0 ± 69.0	107.0	99	75.0 ± 58.6	65.4	100	102.2 ± 65.2	92.9	100
Gemini 2.5 Flash	112.0 ± 90.1	91.7	99	125.6 ± 94.1	112.5	99	122.3 ± 86.3	120.2	100
Llama 3.2 1B	–	–	0	–	–	0	–	–	1
Qwen3 8B	–	–	0	183.4 ± 85.7	175.1	55	148.5 ± 85.5	147.5	100
Ministral 3 8B	3057.4 ± 4112.4	1046.0	99	258.3 ± 408.0	169.1	100	996.4 ± 4916.2	137.3	100

"–" Not reported: fewer than 50 valid TCs.
$n = 0$: evaluated, but no TC is valid (Llama 3.2 1B under CB and Variant A, and Qwen3 8B under CB).
† Temperature parameter not supported. Evaluated with default settings.
Not evaluated (retired before the prompt-sensitivity experiments): DeepSeek-V3.2 (Non-Thinking), DeepSeek-V3.2 (Thinking), Grok 4.1 Fast, Grok 4.1 Fast (Non-Reasoning).

Table 20: MAE (μs) for **CBS** open-ended evaluation across **One-Switch topology** TCs (TC41 to TC51) under the CB setting. The MAE of a TC is computed on the WCD averaged over the valid runs of the TC, using one temperature per model. "–" denotes an invalid, missing or partial response (fewer than 80% of the flows of the TC predicted). "0" denotes a trivial failure (all-zero WCD values). Best result per TC shown in **bold**.

Model	TC41	TC42	TC43	TC44	TC45	TC46	TC47	TC48	TC49	TC50	TC51
Grok 4.1 Fast	127.77	91.48	127.06	32.98	24.74	202.69	132.81	–	133.33	272.76	195.88
Grok 4.1 Fast (Non-Reasoning)	–	5283.22	–	–	3916.63	–	–	–	4248.83	–	5159.01
DeepSeek-V3.2 (Non-Thinking)	410.75	484.56	291.06	–	130.71	–	–	–	237.33	508.76	–
GPT-4o	505.94	486.53	294.39	85.41	124.37	265.02	204.64	63.72	237.22	510.61	415.26
GPT-4o mini	0	0	293.06	133.41	0	301.19	210.14	0	242.67	0	0
Llama 3.3 70B	516.89	322.23	301.96	140.34	43.20	303.73	213.14	58.87	250.67	391.42	427.25
Mistral Medium 3.1	509.17	488.48	294.34	130.85	35.19	295.19	205.13	69.52	240.64	512.68	418.62
Mistral Large 3	397.00	381.23	172.06	20.79	16.35	183.52	76.58	181.61	127.67	403.12	302.10
Claude Sonnet 4.5	499.76	484.85	286.34	109.67	122.86	294.20	202.77	61.65	216.11	499.51	413.98
o3	119.16	**61.05**	121.41	**10.57**	17.50	105.48	84.57	143.37	**19.64**	**125.73**	171.22
GPT-5	**60.23**	177.15	**86.44**	30.66	**15.86**	**50.59**	**40.84**	14.55	23.66	136.91	225.17
DeepSeek-V3.2 (Thinking)	–	–	–	–	–	–	–	–	–	–	–
Gemini 2.5 Flash	–	225.61	220.86	61.78	57.02	135.30	84.65	39.35	164.93	334.01	**158.82**
Small Models											
Llama 3.2 1B	–	–	–	–	–	–	–	0	0	–	–
Qwen3 8B	–	–	–	–	–	–	–	–	–	–	–
Ministral 3 8B	828.44	1183.11	448.31	858.59	863.29	1360.14	1283.86	**10.08**	882.67	713.24	1183.01

Table 21: MAE (μs) for **CQF** open-ended evaluation across **One-Switch topology** TCs (TC41 to TC51) under the CB setting. The MAE of a TC is computed on the WCD averaged over the valid runs of the TC, using one temperature per model. "–" denotes an invalid, missing or partial response (fewer than 80% of the flows of the TC predicted). "0" denotes a trivial failure (all-zero WCD values). Best result per TC shown in **bold**. "0.00" denotes an exact prediction (MAE of 0 µs).

Model	TC41	TC42	TC43	TC44	TC45	TC46	TC47	TC48	TC49	TC50	TC51
Grok 4.1 Fast	221.39	175.67	54.60	46.64	52.00	63.83	–	61.55	212.00	409.67	128.18
Grok 4.1 Fast (Non-Reasoning)	95.00	93.67	89.00	91.00	93.67	95.50	91.00	95.00	93.67	–	91.00
DeepSeek-V3.2 (Non-Thinking)	89.67	95.00	95.00	89.00	91.00	95.00	95.00	89.00	89.00	–	37.89
GPT-4o	316.33	283.33	633.00	**0.00**	**1.00**	32.33	313.67	15.67	949.00	317.00	283.67
GPT-4o mini	88.33	85.67	93.00	91.00	97.00	98.33	97.00	29.67	93.00	91.00	95.44
Llama 3.3 70B	67.72	97.72	29.26	66.82	99.33	37.74	57.94	90.25	717.31	171.29	109.71
Mistral Medium 3.1	92.67	93.00	91.00	90.33	91.00	93.00	91.17	90.00	93.00	92.67	93.00
Mistral Large 3	100.67	84.00	101.00	82.67	101.00	83.33	101.00	101.67	84.00	50.00	101.00
Claude Sonnet 4.5	87.43	**44.48**	**11.68**	46.29	47.44	43.38	**30.28**	48.89	46.54	**13.69**	**31.28**
o3	**34.33**	159.11	146.89	68.99	32.78	69.33	32.40	70.37	47.00	126.11	81.15
GPT-5	169.30	162.19	85.82	74.85	27.11	76.64	54.51	25.58	108.33	160.43	115.66
DeepSeek-V3.2 (Thinking)	–	–	–	–	–	–	–	–	–	–	–
Gemini 2.5 Flash	80.68	91.67	79.79	5.34	15.68	**21.28**	49.73	**8.42**	**44.36**	149.51	80.49
Small Models											
Llama 3.2 1B	0	–	0	–	–	–	–	–	–	–	0
Qwen3 8B	–	–	–	–	–	–	–	–	–	–	–
Ministral 3 8B	1297.00	1484.78	588.33	25.44	633.22	635.67	91.00	897.00	897.00	–	1200.19

Table 22: MAE (μs) for **CBS** open-ended evaluation across 10 **medium-mesh topology** TCs (TC71 to TC80), under the CB setting. The MAE of a TC is computed on the WCD averaged over the valid runs of the TC, using one temperature per model. "–" denotes an invalid, missing or partial response (fewer than 80% of the flows of the TC predicted). "0" denotes a trivial failure (all-zero WCD values). Best result per TC shown in **bold**.

Model	TC71	TC72	TC73	TC74	TC75	TC76	TC77	TC78	TC79	TC80
Grok 4.1 Fast	**20.83**	60.86	**16.10**	70.38	53.55	**157.00**	–	–	71.92	345.35
Grok 4.1 Fast (Non-Reasoning)	5545.59	4219.56	5320.84	2290.48	2848.59	4123.00	2749.35	4131.26	4171.83	2780.78
DeepSeek-V3.2 (Non-Thinking)	0	289.44	0	0	0	0	0	377.74	0	200.22
GPT-4o	2225.26	0	307.31	197.81	101.84	0	131.53	269.74	233.82	142.51
GPT-4o mini	166.41	295.44	415.31	331.66	209.84	387.00	237.65	379.74	340.17	206.22
Llama 3.3 70B	96.75	303.37	417.31	0	192.76	335.54	243.58	0	260.80	206.14
Mistral Medium 3.1	86.15	283.44	326.31	327.59	120.24	303.20	154.67	290.24	256.82	123.68
Mistral Large 3	73.25	192.89	295.31	231.69	99.22	289.40	138.89	271.74	239.77	107.56
Claude Sonnet 4.5	124.07	170.41	336.11	284.87	142.77	332.72	186.06	230.38	300.15	149.62
o3	690.12	55.96	109.72	381.87	62.76	192.38	**47.62**	179.95	194.04	78.94
GPT-5	77.08	**31.23**	80.38	**22.48**	**24.08**	171.81	70.22	**42.16**	**52.26**	151.57
DeepSeek-V3.2 (Thinking)	–	–	–	–	–	–	–	–	–	–
Gemini 2.5 Flash	42.76	57.38	258.11	–	3352.85	217.41	98.63	145.60	211.59	**57.35**
Small Models										
Llama 3.2 1B	0	–	–	–	–	–	–	–	–	–
Qwen3 8B	–	–	–	–	–	–	–	–	–	179.08
Ministral 3 8B	157.41	696.56	812.69	856.34	978.16	851.00	950.35	680.86	847.83	981.78

Table 23: MAE (μs) for **CQF** open-ended evaluation across 10 **medium-mesh topology** TCs (TC71 to TC80), under the CB setting. The MAE of a TC is computed on the WCD averaged over the valid runs of the TC, using one temperature per model. "–" denotes an invalid, missing or partial response (fewer than 80% of the flows of the TC predicted). "0" denotes a trivial failure (all-zero WCD values). Best result per TC shown in **bold**. "0.00" denotes an exact prediction (MAE of 0 µs).

Model	**TC71**	**TC72**	**TC73**	**TC74**	**TC75**	**TC76**	**TC77**	**TC78**	**TC79**	**TC80**
Grok 4.1 Fast	–	106.71	180.10	96.66	89.61	80.55	31.38	–	84.00	55.00
Grok 4.1 Fast (Non-Reasoning)	139.00	142.00	142.00	139.00	136.00	139.00	139.00	139.00	139.00	139.00
DeepSeek-V3.2 (Non-Thinking)	142.00	140.00	142.00	98.00	142.00	142.00	142.00	140.00	142.00	142.00
GPT-4o	**1.00**	**1.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**	**1.00**	**1.00**	**0.00**
GPT-4o mini	146.00	142.00	142.00	142.00	142.00	142.00	142.00	148.00	146.00	142.00
Llama 3.3 70B	141.00	141.00	141.00	141.00	141.00	141.00	141.00	130.80	141.00	141.00
Mistral Medium 3.1	141.00	143.00	141.00	141.00	141.00	141.00	144.00	141.00	141.00	141.00
Mistral Large 3	50.00	50.00	52.00	50.00	50.00	50.00	49.00	49.00	50.00	50.00
Claude Sonnet 4.5	74.19	104.53	34.18	3.51	19.02	2.48	2.13	28.42	13.70	2.17
o3	2.00	153.00	2.00	28.07	41.03	48.56	129.11	137.80	44.57	113.05
GPT-5	39.81	89.97	183.10	107.01	43.71	52.00	47.06	144.60	49.25	63.05
DeepSeek-V3.2 (Thinking)	–	–	–	–	–	–	–	–	–	–
Gemini 2.5 Flash	21.82	32.29	188.30	113.73	19.34	192.68	81.10	44.70	99.48	71.29
Small Models										
Llama 3.2 1B	–	–	–	–	–	–	–	0	–	–
Qwen3 8B	–	–	–	–	–	–	–	–	–	–
Ministral 3 8B	1046.00	11846.00	1116.00	846.00	1096.00	1007.75	846.00	844.80	1046.00	846.00

G.6 Case study: Orion CEV network topology

To test whether our findings carry over from the synthetic topologies to a real-world topology, we evaluate the models on TCs derived from the Orion Crew Exploration Vehicle (CEV) network topology [Zhao et al., 2022]. This topology is a widely used reference in TSN research. It consists of 31 end systems and 15 switches connected by 100 Mbps links (Figure 15).

Test cases: We generated 10 TCs per mechanism on this topology. Each TC follows the same format as the synthetic TCs.

Evaluation: All models are evaluated with the same prompts, the same CB and FC settings, up to three runs per TC and the same scoring as in Section 4.2. The GT WCD is computed with the same NC analysis for CBS and the same closed-form bound (Eq. 2) for CQF as for the main benchmark. Models retired by their providers before these experiments (Grok 4.1 Fast in reasoning and non-reasoning mode, and DeepSeek-V3.2 in thinking and non-thinking mode) are not included. With only 10 TCs, we replace the minimum of 50 valid TCs by a minimum of 5 valid TCs and report the number of valid TCs (n) for every model. For CBS, Table 26 reports the mean and median MAE and Δ for each model under CB and FC and Table 27 the MAE with its standard deviation. Tables 30 and 31 report the same for CQF, and Tables 28, 29, 32, and 33 give the per-TC results.

CBS results: As on the synthetic TCs, FC lowers the mean MAE for only a minority of models. Ten models have at least 5 valid TCs in both settings. The valid TCs of GPT-4o mini and Llama 3.3 70B differ between CB (Table 28) and FC (Table 29). As in Table 3, Δ compares the mean MAE over the TCs valid in each setting. FC lowers the mean MAE for four, namely GPT-4o mini ($+8.7\%$), o3 ($+4.3\%$), GPT-5 ($+43.8\%$), and Qwen3 8B ($+62.0\%$), and raises it for the other six (Δ between -0.4% and -665.3%). The gain of GPT-5 rests on a single TC. Its CB mean is driven by TC110 (7,020.8 μs), while its other nine TCs stay between 43.5 and 145.4 μs, and FC lowers its error in only 1 of its 10 TCs (Tables 28 and 29). Ministral 3 8B, which FC helps on the synthetic TCs, gets worse with FC in all 10 Orion TCs. In the CB setting, o3 has the lowest MAE (256.1 μs) and GPT-5 the lowest median MAE (99.5 μs).

CQF results: Unlike CBS, FC lowers the mean MAE for most models on CQF (7 of 11, Table 25). o3, GPT-5, and Gemini 2.5 Flash reach an MAE of 0.0 μs in all 10 TCs, and Qwen3 8B reaches 4.7 μs (Table 30). FC raises the mean MAE for GPT-4o mini, Llama 3.3 70B, Claude Sonnet 4.5, and Ministral 3 8B. For Llama 3.3 70B, the increase comes from two TCs (TC104 and TC108), while its median falls from 164.1 to 26.3 μs. In the CB setting, GPT-4o has the lowest MAE (1.5 μs).

Table 24: CBS WCD estimation on the Orion CEV topology (10 TCs) over the models with at least 5 valid TCs (↓ lower is better). Median MAE is the median of the models' mean MAEs. FC lowers the mean MAE for 4 of the 10 models that have at least 5 valid TCs in both settings.

Setting	Models	Median MAE (μs)	Best model (MAE, μs)
CB	10	406.7	o3 (256.1)
FC	10	431.0	o3 (245.1)

Table 25: CQF WCD estimation on the Orion CEV topology (10 TCs) over the models with at least 5 valid TCs (↓ lower is better). Median MAE is the median of the models' mean MAEs. FC lowers the mean MAE for 7 of the 11 models that have at least 5 valid TCs in both settings.

Setting	Models	Median MAE (μs)	Best model (MAE, μs)
CB	11	136.3	GPT-4o (1.5)
FC	11	21.3	o3* (0.0)

* Tied with GPT-5 and Gemini 2.5 Flash (0.0 μs).

Table 26: Open-ended WCD estimation results for CBS on the Orion CEV topology (10 TCs), reported as mean and median MAE across the valid TCs (μs, $\downarrow$ lower is better). Δ is the relative change of the mean MAE from CB to FC, with each mean taken over the valid TCs of its setting. Positive values mean the formula lowers the error. For GPT-4o mini and Llama 3.3 70B, the valid TCs differ between CB and FC (Tables 28 and 29), so their Δ compares means over different TCs.

	CB		FC		
Model	**MAE**	**Median**	**MAE**	**Median**	**Δ (%)**
GPT-4o mini$^\sharp$	412.4	420.1	376.7	412.2	$+8.7$
Llama 3.3 70B$^\sharp$	324.2	293.8	487.4	251.6	-50.3
Mistral Medium 3.1	926.7	901.1	1304.5	872.4	-40.8
Mistral Large 3	339.6	335.4	340.9	337.5	-0.4
Claude Sonnet 4.5	358.2	350.3	366.9	352.7	-2.4
o3†	256.1	223.8	245.1	190.6	$+4.3$
GPT-5†	790.0	99.5	444.0	142.4	$+43.8$
Gemini 2.5 Flash	400.9	312.2	418.1	435.6	-4.3
Qwen3 8B	19058.4	7477.9	7249.0	2074.1	$+62.0$
Ministral 3 8B	3031.6	2410.3	23200.1	9906.9	-665.3

† Temperature parameter not supported, evaluated with default settings.
$^\sharp$ Valid TCs differ between CB and FC (Tables 28 and 29). As in Table 3, each mean is taken over the TCs valid in that setting.
Not shown (fewer than 5 valid TCs in both settings): GPT-4o (1 valid TC), Llama 3.2 1B (no valid TC).
Not evaluated (retired before the Orion experiments): Grok 4.1 Fast, Grok 4.1 Fast (Non-Reasoning), DeepSeek-V3.2 (Non-Thinking), DeepSeek-V3.2 (Thinking).

Table 27: MAE (μs) $\pm$ standard deviation and median MAE (μs) across the valid TCs for CBS on the Orion CEV topology under CB and FC ($\downarrow$ lower is better). n is the number of valid TCs out of 10.

	CB			FC		
Model	MAE $\pm$ std	Median	n	MAE $\pm$ std	Median	n
GPT-4o	–	–	1	–	–	1
GPT-4o mini	412.4 ± 116.0	420.1	10	376.7 ± 126.9	412.2	7
Llama 3.3 70B	324.2 ± 88.3	293.8	8	487.4 ± 622.4	251.6	8
Mistral Medium 3.1	926.7 ± 322.6	901.1	10	1304.5 ± 1019.3	872.4	10
Mistral Large 3	339.6 ± 95.2	335.4	10	340.9 ± 93.2	337.5	10
Claude Sonnet 4.5	358.2 ± 97.7	350.3	10	366.9 ± 98.2	352.7	10
o3†	256.1 ± 154.7	223.8	10	245.1 ± 159.7	190.6	10
GPT-5†	790.0 ± 2077.1	99.5	10	444.0 ± 914.0	142.4	10
Gemini 2.5 Flash	400.9 ± 225.6	312.2	9	418.1 ± 96.7	435.6	9
Llama 3.2 1B	–	–	0	–	–	0
Qwen3 8B	19058.4 ± 31460.9	7477.9	10	7249.0 ± 11276.2	2074.1	10
Ministral 3 8B	3031.6 ± 2486.2	2410.3	10	23200.1 ± 27611.0	9906.9	10

"–" Not reported: fewer than 5 valid TCs.
$n = 0$: evaluated, but no TC is valid.
† Temperature parameter not supported, evaluated with default settings.
Not evaluated (retired before the Orion experiments): Grok 4.1 Fast, Grok 4.1 Fast (Non-Reasoning), DeepSeek-V3.2 (Non-Thinking), DeepSeek-V3.2 (Thinking).

Table 28: MAE (μs) of each TC for CBS on the Orion CEV topology under the CB setting (↓ lower is better).

Model	TC101	TC102	TC103	TC104	TC105	TC106	TC107	TC108	TC109	TC110
GPT-4o	–	–	–	–	–	–	361.1	–	–	–
GPT-4o mini	209.1	548.4	538.3	547.9	239.5	425.5	472.1	331.5	414.7	396.8
Llama 3.3 70B	285.4	–	485.3	399.8	176.9	276.8	300.9	–	381.8	286.7
Mistral Medium 3.1	1034.3	631.2	991.4	415.2	1293.9	672.6	1603.0	960.4	823.3	841.8
Mistral Large 3	345.0	450.7	441.4	455.0	139.0	325.7	383.8	238.6	313.4	302.9
Claude Sonnet 4.5	354.3	484.6	449.0	472.0	150.5	346.3	407.0	252.3	339.2	326.8
o3†	209.5	361.5	183.1	359.2	88.9	93.9	274.4	238.1	627.6	124.8
GPT-5†	88.2	103.4	120.8	145.4	43.5	81.1	94.6	95.7	106.1	7020.8
Gemini 2.5 Flash	442.5	472.7	285.4	299.2	162.8	–	250.2	984.1	399.3	312.2
Llama 3.2 1B	–	–	–	–	–	–	–	–	–	–
Qwen3 8B	13989.3	677.6	25724.8	556.9	109081.0	12906.5	584.1	482.2	2049.3	24532.3
Ministral 3 8B	765.6	7630.4	3820.4	555.2	1000.3	4339.6	6428.3	778.5	4195.6	802.4

"–" No valid answer.
† Temperature parameter not supported, evaluated with default settings.
Not evaluated (retired before the Orion experiments): Grok 4.1 Fast, Grok 4.1 Fast (Non-Reasoning), DeepSeek-V3.2 (Non-Thinking), DeepSeek-V3.2 (Thinking).

Table 29: MAE (μs) of each TC for CBS on the Orion CEV topology under the FC setting (↓ lower is better).

Model	TC101	TC102	TC103	TC104	TC105	TC106	TC107	TC108	TC109	TC110
GPT-4o	–	–	–	–	–	–	361.1	–	–	–
GPT-4o mini	441.6	–	–	539.9	236.8	–	465.1	146.6	412.2	394.4
Llama 3.3 70B	234.9	2124.4	297.5	251.9	149.5	195.6	–	–	251.3	393.8
Mistral Medium 3.1	816.9	933.1	1489.5	415.2	852.2	409.4	3465.2	3040.6	730.2	892.6
Mistral Large 3	349.7	448.3	442.6	452.9	146.4	325.2	381.9	237.0	321.1	303.9
Claude Sonnet 4.5	321.6	492.9	492.6	487.1	181.8	367.1	396.4	261.9	338.2	329.5
o3†	186.6	165.2	357.2	319.1	65.7	158.7	233.7	115.5	194.7	654.9
GPT-5†	148.8	122.8	221.3	188.1	57.5	194.6	3182.2	104.4	136.1	84.3
Gemini 2.5 Flash	443.3	478.1	540.8	549.9	241.8	–	435.6	334.4	415.9	322.6
Llama 3.2 1B	–	–	–	–	–	–	–	–	–	–
Qwen3 8B	3670.5	38235.5	11846.8	194.2	4697.0	121.2	477.8	12596.7	249.5	400.9
Ministral 3 8B	4483.5	61053.0	12836.5	36939.8	1473.7	86406.2	9704.0	1019.6	10109.8	7974.9

"–" No valid answer.
† Temperature parameter not supported, evaluated with default settings.
Not evaluated (retired before the Orion experiments): Grok 4.1 Fast, Grok 4.1 Fast (Non-Reasoning), DeepSeek-V3.2 (Non-Thinking), DeepSeek-V3.2 (Thinking).

Table 30: Open-ended WCD estimation results for CQF on the Orion CEV topology (10 TCs), reported as mean and median MAE across the valid TCs (μs, ↓ lower is better). Δ is the relative change of the mean MAE from CB to FC, with each mean taken over the valid TCs of its setting. Positive values mean the formula lowers the error.

	CB		**FC**		
Model	**MAE**	**Median**	**MAE**	**Median**	**Δ (%)**
GPT-4o	1.5	1.5	1.5	1.2	+0.6
GPT-4o mini	190.9	174.0	401.0	405.1	−110.1
Llama 3.3 70B	164.4	164.1	8029.5	26.3	−4785.1
Mistral Medium 3.1	189.7	121.3	21.3	9.7	+88.7
Mistral Large 3	45.4	45.4	22.7	30.2	+49.9
Claude Sonnet 4.5	24.7	21.1	43.8	50.3	−77.7
o3†	118.3	97.2	0.0	0.0	+100.0
GPT-5†	127.4	114.8	0.0	0.0	+100.0
Gemini 2.5 Flash	136.3	131.9	0.0	0.0	+100.0
Qwen3 8B	721.0	168.2	4.7	4.9	+99.3
Ministral 3 8B	10079.0	867.5	26619.3	819.9	−164.1

† Temperature parameter not supported, evaluated with default settings.
Not shown (fewer than 5 valid TCs in both settings): Llama 3.2 1B (no valid TC).
Not evaluated (retired before the Orion experiments): Grok 4.1 Fast, Grok 4.1 Fast (Non-Reasoning), DeepSeek-V3.2 (Non-Thinking), DeepSeek-V3.2 (Thinking).
FC raises Llama 3.3 70B's mean MAE ($-4{,}785.1\%$) because of two TCs (TC104 and TC108, MAE 32,031.8 and 48,032.4 μs), while its median FC error falls from 164.1 to 26.3 μs.

Table 31: MAE (μs) $\pm$ standard deviation and median MAE (μs) across the valid TCs for CQF on the Orion CEV topology under CB and FC ($\downarrow$ lower is better). n is the number of valid TCs out of 10.

	CB			FC		
Model	MAE $\pm$ std	Median	n	MAE $\pm$ std	Median	n
GPT-4o	1.5 ± 0.0	1.5	10	1.5 ± 0.8	1.2	10
GPT-4o mini	190.9 ± 52.3	174.0	10	401.0 ± 215.6	405.1	7
Llama 3.3 70B	164.4 ± 1.1	164.1	10	8029.5 ± 16396.4	26.3	10
Mistral Medium 3.1	189.7 ± 201.3	121.3	10	21.3 ± 21.7	9.7	10
Mistral Large 3	45.4 ± 0.0	45.4	10	22.7 ± 15.0	30.2	10
Claude Sonnet 4.5	24.7 ± 18.2	21.1	10	43.8 ± 11.2	50.3	10
o3†	118.3 ± 62.2	97.2	10	0.0 ± 0.0	0.0	10
GPT-5†	127.4 ± 53.0	114.8	10	0.0 ± 0.0	0.0	10
Gemini 2.5 Flash	136.3 ± 55.3	131.9	10	0.0 ± 0.0	0.0	10
Llama 3.2 1B	–	–	0	–	–	0
Qwen3 8B	721.0 ± 1197.5	168.2	10	4.7 ± 3.7	4.9	10
Ministral 3 8B	10079.0 ± 17916.9	867.5	10	26619.3 ± 35065.0	819.9	10

"–" Not reported: fewer than 5 valid TCs.
$n = 0$: evaluated, but no TC is valid.
† Temperature parameter not supported, evaluated with default settings.
Not evaluated (retired before the Orion experiments): Grok 4.1 Fast, Grok 4.1 Fast (Non-Reasoning), DeepSeek-V3.2 (Non-Thinking), DeepSeek-V3.2 (Thinking).

Table 32: MAE (μs) of each TC for CQF on the Orion CEV topology under the CB setting ($\downarrow$ lower is better).

Model	TC101	TC102	TC103	TC104	TC105	TC106	TC107	TC108	TC109	TC110
GPT-4o	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
GPT-4o mini	347.8	174.4	171.4	171.5	174.6	174.5	173.6	174.6	173.6	172.7
Llama 3.3 70B	164.1	164.0	164.1	163.9	164.1	163.6	164.0	163.9	164.1	167.7
Mistral Medium 3.1	212.2	325.4	59.8	736.8	39.3	103.5	173.8	55.4	139.1	51.4
Mistral Large 3	45.4	45.4	45.4	45.4	45.4	45.4	45.4	45.4	45.4	45.4
Claude Sonnet 4.5	30.8	14.8	3.9	45.5	22.1	20.1	67.7	11.7	22.0	8.1
o3†	87.7	53.2	260.0	148.9	33.5	96.2	94.2	98.2	179.5	131.2
GPT-5†	120.1	104.9	125.0	197.1	73.6	77.7	142.3	76.9	246.5	109.5
Gemini 2.5 Flash	154.7	227.2	95.8	137.7	152.3	230.6	126.1	75.7	54.9	107.8
Llama 3.2 1B	–	–	–	–	–	–	–	–	–	–
Qwen3 8B	120.3	3814.5	178.3	128.5	166.3	170.1	136.5	2191.1	178.3	125.6
Ministral 3 8B	29503.5	57545.7	4268.8	4684.1	806.8	887.6	817.3	827.8	847.4	601.1

"–" No valid answer.
† Temperature parameter not supported, evaluated with default settings.
Not evaluated (retired before the Orion experiments): Grok 4.1 Fast, Grok 4.1 Fast (Non-Reasoning), DeepSeek-V3.2 (Non-Thinking), DeepSeek-V3.2 (Thinking).

Table 33: MAE (μs) of each TC for CQF on the Orion CEV topology under the FC setting ($\downarrow$ lower is better).

Model	TC101	TC102	TC103	TC104	TC105	TC106	TC107	TC108	TC109	TC110
GPT-4o	1.0	1.0	0.5	1.3	1.0	0.5	2.0	3.0	2.1	2.1
GPT-4o mini	560.7	730.0	–	210.0	270.8	563.5	66.9	405.1	–	–
Llama 3.3 70B	59.9	27.3	14.3	32031.8	15.6	25.3	23.4	48032.4	18.4	46.4
Mistral Medium 3.1	51.9	50.0	3.6	52.0	15.9	2.1	2.0	2.1	0.7	33.0
Mistral Large 3	30.5	31.8	2.9	33.5	30.8	30.0	47.0	1.9	16.9	2.0
Claude Sonnet 4.5	50.3	51.0	50.3	33.7	51.0	51.0	50.7	17.0	50.0	33.3
o3†	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GPT-5†	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Gemini 2.5 Flash	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Llama 3.2 1B	–	–	–	–	–	–	–	–	–	–
Qwen3 8B	8.1	0.8	6.5	0.0	6.5	0.0	3.2	9.7	2.4	9.7
Ministral 3 8B	634.1	86861.6	814.4	699.0	438.8	86859.8	30067.2	814.4	825.4	58178.7

"–" No valid answer.
† Temperature parameter not supported, evaluated with default settings.
Not evaluated (retired before the Orion experiments): Grok 4.1 Fast, Grok 4.1 Fast (Non-Reasoning), DeepSeek-V3.2 (Non-Thinking), DeepSeek-V3.2 (Thinking).

Table 34: CBS Error Analysis Case 1: Lack of Specific Knowledge.

Test Case: TC1
TSN mechanism: CBS

You are an expert Time-Sensitive Networking (TSN) orchestrator. Your task is to calculate the worst case delay (WCD) for each TSN flow.

Input:
Network Topology ($TC1_topo.txt$)
Flow Information ($TC1_flows.txt$)
Routing of the Flow ($TC1_route.txt$)

Constant:
Bandwidth link = 100 Mbps; Propagation delay = 1 μs; Switching delay = 1 μs;
Time synchronization error = 1 μs; The switches of the network are cut-through switches.

TSN Mechanism:
Only Credit-Based Shaper (CBS, IEEE 802.1Qav) is allowed;
All flows are AVB Class A, PCP = 6, using queue 6 only.

Task:

1. Map each egress port's queues and collect the set of flows traversing from that port, using the given topology, flows, and route of the flow.
2. For each egress port, use the given IdleSlope and then compute the SendSlope.
3. For each flow, construct an arrival curve from its frame size and periodicity.
4. For each port, derive a lower-bounded CBS service curve.
5. Calculate the worst case delay (WCD) in microseconds (μs) for each flow using Network Calculus method.
6. Provide the confidence score between 0.0 and 1.0 from your answers. 1.0 means mathematically or procedurally provable from given info with zero ambiguity. 0.0 means zero confidence.

Table 34: CBS Error Analysis Case 1: Lack of Specific Knowledge. (continued)

Grok 4.1 Fast (Non-Reasoning) output:

F0: 1452.0, F1: 1124.0, F2: 678.0, F3: 1234.0, F4: 1567.0, F5: 689.0, F6: 456.0, F7: 789.0, F8: 567.0, F9: 589.0, F10: 678.0, F11: 645.0, F12: 1123.0, F13: 789.0, F14: 456.0, F15: 678.0, F16: 789.0, F17: 1567.0, F18: 456.0, F19: 1234.0

Model Reasoning and Explanation Output:

Parameters used:
send_slope: $75\% \times 100$ Mbps $= 75$ Mbps $= 75000$ bits/μs
frame_overhead: 24 bytes (Ethernet header + FCS),
mtu: 1522 bytes (max frame size),
queue: 6 (PCP=6, AVB Class A),
cbs_credit_max: MTU $\times$ 8 $= 12176$ bits

WCD equation given by the model:

$$\alpha(t) = \frac{L_{\max}}{\text{Period}} + \frac{t}{\text{Period}} \quad \text{(arrival curve)}$$

$$\beta(t) = R_{\text{send}} \times t - C \quad \text{(CBS service curve)}$$

$$WCD = h \times (\text{prop_delay} + \text{switch_delay} + \text{sync_error}) + \max(\alpha \cap \beta)$$

$$\text{SendSlope} = \text{IdleSlope} \times \text{Bandwidth} = 0.75 \times 100 \text{ Mbps}$$

$$R_{\text{send}} = \text{SendSlope} = 75 \text{ Mbps}$$

$$C = \text{credit}_{\max} = \text{MTU} \times 8 \text{ bits}$$

$$h = \text{hop count from path}$$

Per-flow WCD:

$$WCD_{F_i} = \sum_{h \in \text{path}(F_i)} \left(d^h_{\text{prop}} + d^h_{\text{switch}} + d^h_{\text{sync}}\right) + \max_{t \geq 0} \left(\alpha_{F_i}(t) \cap \beta_{\text{port}}(t)\right)$$

Table 34: CBS Error Analysis Case 1: Lack of Specific Knowledge. (continued)

Run 2 of the model (Grok 4.1 Fast (Non-Reasoning)):

WCD equation given by the model during run 2:

$$\begin{aligned}
\text{Frame size:} \quad & L_i = \text{payload_}i + 24 \text{ bytes}, \\
\text{Transmission time:} \quad & T_{x_i} = \frac{8L_i}{R}, \\
\text{Arrival curve:} \quad & \alpha_i(t) = \frac{L_i}{p_i} \cdot t, \\
\text{CBS service curve:} \quad & \beta(t) = \min\left\{\alpha_{h_i}(t),\ \text{sendSlope} \cdot t + \text{credit}_{\max}\right\}, \\
\text{Worst-case delay:} \quad & WCD_i = \max_{h \in \text{path}} \sum \left(T_{\text{prop},h} + T_{\text{switch},h} + T_{\text{queue},h}\right) + T_{\text{sync}}, \\
\text{Queueing delay:} \quad & T_{\text{queue},h} = \frac{L_i}{\alpha C} + \text{burst_interference}, \\
\text{Send slope:} \quad & \text{sendSlope} = \text{idleSlope} \cdot R - \text{idleSlope} \cdot R = \alpha R(1 - 2\alpha);
\end{aligned}$$

Per-flow WCD:

$$WCD_i = \text{hop_count}_i \cdot (1 + 1) + \text{max_queue_delay} + 1;$$

Expert Explanation:

The model uses different equations across different runs even with temperature set to 0.0. When analyzing the first run, the model makes several fundamental errors in evaluating the interference/blocking experienced by the queue under analysis, i.e., both the arrival curve and CBS service curve.
First, it assumes the wrong maximum burst size and does not consider the variant output arrival curve at subsequent hops along the path.
Second, it assumes a non-zero maximum credit, which is possible in this scenario due to the non-preemption frame of lower priority of non-CBS traffic. However, the model applies the wrong maximum credit to the CBS service curve, incorrectly understanding the relation of the corresponding rate (75%).
Furthermore, in the final WCD calculation, the model considers only the arrival curve of the frame under analysis. This is incorrect: the proper approach requires using the aggregate arrival curve of all frames transmitted through the queue. It is also not clear whether the $max(\alpha \cap \beta)$ operation represents the maximum horizontal deviation between α and β. The output arrival curve is not considered in the following hops' calculation.
On the other hand, the second run exhibits even more severe hallucinations, producing calculations of the wrong arrival curve without considering any burst size, and the wrong CBS service curve considering the wrong service rate (sendSlope) and wrong relation between the corresponding rate and the maximum credit, as well as without considering the output arrival curve.

GPT-5 output:

F0: 714.65, F1: 821.79, F2: 171.48, F3: 687.17, F4: 907.26, F5: 171.48, F6: 183.9, F7: 453.91, F8: 611.08, F9: 181.3, F10: 611.08, F11: 321.35, F12: 1124.12, F13: 483.67, F14: 172.43, F15: 489.26, F16: 521.33, F17: 918.73, F18: 181.3, F19: 624.12

Table 34: CBS Error Analysis Case 1: Lack of Specific Knowledge. (continued)

Model Reasoning and Explanation Output:

$$\text{sendSlope} = -25 \text{ Mbps},$$

$$\text{Arrival curve:} \quad b = \sum_{\forall \text{F}_\text{i} \mid \text{ share output port}} L_i \text{ bits},$$

$$r = \sum_{\forall \text{F}_\text{i} \mid \text{ share output port}} \frac{L_i}{p_i} \text{ bits}/\mu\text{s},$$

$$\alpha(t) = b + rt,$$

$$\begin{aligned} T_{\text{tech,total}} &= T_{\text{prop}} + T_{\text{switching}} + T_{\text{sync}} \\ &= hops \times 1\ \mu\text{s} + (hops - 1) \times 1\ \mu\text{s} + (hops - 1) \times 1\ \mu\text{s} = 13\ \mu\text{s}, \end{aligned}$$

$$\begin{aligned} WCD_{\text{total}} &= WCD_{\text{sum}} + T_{\text{tech,total}} \\ &= 701.65\ \mu\text{s} + 13\ \mu\text{s} = 714.65\ \mu\text{s}. \end{aligned}$$

Expert Explanation:

GPT-5 uses the correct arrival and service curves. Its main error is that it fails to include the output port (output arrival curves) in the analysis.

Ministral 3 8B output:

F0: 1250.0, F1: 1100.0, F2: 500.0, F3: 1500.0, F4: 600.0, F5: 1100.0, F6: 2500.0, F7: 1200.0, F8: 500.0, F9: 500.0, F10: 2000.0, F11: 500.0, F12: 1000.0, F13: 1000.0, F14: 2000.0, F15: 500.0, F16: 1500.0, F17: 1000.0, F18: 500.0, F19: 2000.0

Model Reasoning and Explanation Output:

$$\text{Bandwidth:} \quad R = 100 \text{ Mbps} = 12.5 \text{ MB/s},$$

$$\text{Propagation delay:} \quad T_{\text{prop}} = 1\ \mu\text{s} \quad \text{per hop},$$

$$\text{Switching delay:} \quad T_{\text{switch}} = 1\ \mu\text{s} \quad \text{per switch},$$

$$\text{Time synchronization error:} \quad T_{\text{sync}} = 1\ \mu\text{s},$$

$$\text{IdleSlope:} \quad \text{IdleSlope} = 0.75R = 75 \text{ Mbps} = 9.375 \text{ MB/s},$$

$$\text{Cut-through switch:} \quad T_{\text{buffer}} = 0,$$

$$\text{Payload overhead:} \quad L_{\text{overhead}} = 18 \text{ B} + 4 \text{ B} = 22 \text{ B},$$

$$\text{Maximum frame size:} \quad L_{\text{max}} = 1433 \text{ B} + 22 \text{ B} = 1455 \text{ B},$$

$$\text{Maximum transmission time:} \quad T_{\text{tx,max}} = \frac{1455 \times 8}{100 \text{ Mbps}} = 9.24\ \mu\text{s}.$$

Table 34: CBS Error Analysis Case 1: Lack of Specific Knowledge. (continued)

$$\begin{aligned}\text{SendSlope} &= \text{IdleSlope}\,(1 - \text{IdleSlope}) \\ &= 9.375\ \text{MB/s} \times 0.25 \\ &= 2.34375\ \text{MB/s},\end{aligned}$$

Burst size: $B_{\text{burst},i} = \text{SendSlope} \cdot p_i,$

CBS service curve: $S(t) = \text{SendSlope}\left(t - \dfrac{B_{\text{burst}}}{\text{SendSlope}}\right), \quad t \geq \dfrac{B_{\text{burst}}}{\text{SendSlope}}.$

WCD Calculation:

Arrival curve for flow F_x : $R(t) = \min\{B_x,\ S_x t\},$

$$B_x = L_{\text{payload},x} + L_{\text{overhead}},$$

$$S_x = \frac{8L_{\text{payload},x}}{p_x},$$

Service curve for each port: $S(t) = \text{SendSlope}\left(t - \dfrac{B_{\text{burst}}}{\text{SendSlope}}\right), \quad t \geq \dfrac{B_{\text{burst}}}{\text{SendSlope}},$

$$WCD = \max\left(0,\ \frac{B_{\text{burst}} - \text{SendSlope} \cdot Period}{\text{SendSlope}}\right) + T_{\text{prop}} + T_{\text{switch}} + T_{\text{sync}}.$$

Expert Explanation:

As a small model, Ministral 3 8B delivers a poor result. Similar to what was observed with Grok 4.1 Fast (Non-Reasoning), it fails to account for the arrival curves of flows and the wrong CBS service curve.
In addition, it analyzes only a single queue, instead of computing the delay at the output port of each device along the end-to-end path.
Moreover, it introduces unit inconsistencies such as dividing the frame size expressed in bytes by a link speed given in bits/μs, which leads to incorrect results.

Table 35: CQF Error Analysis Case 1: Lack of Specific Knowledge.

Test Case: TC1
TSN mechanism: CQF

You are an expert Time-Sensitive Networking (TSN) orchestrator. Your task is to calculate the worst case delay (WCD) for each TSN flow.

Input:
Network Topology ($TC1_topo.txt$)
Flow Information ($TC1_flows.txt$)
Routing of the Flow ($TC1_route.txt$)

Constant:
Bandwidth link = 100 Mbps; Propagation delay = 1 μs; Switching delay = 1 μs;
Time synchronization error = 1 μs; The switches of the network are cut-through switches; cycle duration is 50 μs.

TSN Mechanism:
Only Cyclic Queuing and Forwarding (CQF, IEEE 802.1Qch) is allowed;
All flows are TT, PCP = 7, using queue 7 (odd) and 6 (even) only.

Task:

1. Map each egress port's queues and collect the set of flows traversing that port, using the given topology, flows, and route of the flow.
2. For the entire network, use the given cycle duration and compute the Hypercycle.
3. For each flow, set the offset or the start time of the flow from the sending node as 0.
4. Calculate the worst case delay (WCD) in microseconds (μs) for each flow.
5. Provide the confidence score between 0.0 and 1.0 from your answers. 1.0 means mathematically or procedurally provable from given info with zero ambiguity. 0.0 means zero confidence.

Table 35: CQF Error Analysis Case 1: Lack of Specific Knowledge. (continued)

Claude Sonnet's output:

F0: 257.72, F1: 206.8, F2: 105.096, F3: 218.704, F4: 253.904, F5: 104.008, F6: 103.808, F7: 153.52, F8: 119.408, F9: 119.752, F10: 122.896, F11: 118.464, F12: 222.04, F13: 156.552, F14: 103.808, F15: 156.576, F16: 155.68, F17: 256.616, F18: 101.696, F19: 208.328

Model Reasoning and Explanation Output:

$$\begin{aligned} &\text{Cycle duration: } = 50\ \mu\text{s}, \\ \text{LCM of all periods and cycle:}\quad &\text{LCM}(1000, 2500, 5000, 50) = 5000\ \mu\text{s}, \\ &\text{Hypercycle: } = 5000\ \mu\text{s}. \end{aligned}$$

$$\text{WCD equation:}\quad WCD = \sum_{\text{hops}} (T_{\text{trans}} + T_{\text{prop}} + T_{\text{sw}} + T_{\text{sync}}) + T_{\text{buffering}}.$$

$$\begin{aligned} T_{\text{trans}} &= \frac{(\text{Payload} + \text{Ethernet_overhead}) \cdot 8}{R}, \\ T_{\text{prop}} &= 1\ \mu\text{s} \quad \text{per link}, \\ T_{\text{sw}} &= 1\ \mu\text{s} \quad \text{per switch}, \\ T_{\text{sync}} &= 1\ \mu\text{s} \quad \text{per hop}, \\ T_{\text{buffering}} &= (\text{hop_count} - 1) \cdot T_{\text{cycle}} \end{aligned}$$

In CQF, frames may wait up to one cycle duration at each intermediate switch if they arrive just after the transmission window closes. For cut-through with CQF, worst case adds $(N-1) * \text{cycle_duration}$ where N is hop count.

$$\begin{aligned} \text{Hop count} &= 5, \\ \text{Payload_bytes} &= 965\ \text{bytes}, \\ \text{Frame_size_bytes} &= 1003\ \text{bytes}, \\ \text{Transmission_delay} &= 80.24\ \mu\text{s}, \\ \text{Total propagation delay} &= 5\ \mu\text{s}, \\ \text{Total switching delay} &= 4\ \mu\text{s}, \\ \text{Total synchronization error} &= 5\ \mu\text{s}, \\ \text{CQF buffering} &= 200\ \mu\text{s}, \\ \text{WCD} &= 257.72\ \mu\text{s}. \end{aligned}$$

Ground truth WCD:

$$\begin{aligned} &\text{Route hop count of } f_0 : 5, T = 50\ \mu\text{s}, \\ WCD_{f_0} &= f_0.\phi + (SW_{\text{num}} + 1) \cdot T + \xi \\ &= (4+1) \times 50 + (4 \times 1) + (1+1) \\ &= 256\ \mu\text{s}. \end{aligned}$$

Expert Explanation:

The model knows the working mechanism of CQF, but its WCD equation is a generic per-hop delay sum, not the CQF bound. It also calculates the wrong Hypercycle by including cycle duration with the periods.

Table 35: CQF Error Analysis Case 1: Lack of Specific Knowledge. (continued)

GPT-4o output:

F0: 260.0, F1: 210.0, F2: 60.0, F3: 210.0, F4: 260.0, F5: 60.0, F6: 60.0, F7: 110.0, F8: 60.0, F9: 60.0, F10: 60.0, F11: 60.0, F12: 210.0, F13: 110.0, F14: 60.0, F15: 110.0, F16: 110.0, F17: 260.0, F18: 60.0, F19: 210.0

Model Reasoning and Explanation Output:

$$\text{Hypercycle} = 100\ \mu\text{s}.$$

$$\text{WCD} = (\text{hop_count} \times (\text{cycle_duration} + T_{\text{switch}} + T_{\text{prop}} + T_{\text{sync}})) + \frac{\text{Payload}}{\text{Bandwidth}}$$

Each flow's path, cycle duration, hypercycle, and CQF parameters are detailed, along with the equation used for WCD calculation. No missing inputs are identified, but assumptions are made based on typical network configurations.

Expert Explanation:

The Hypercycle calculated by the model is wrong. Moreover, the WCD equation used by the model has an extra wrong component $\frac{\text{Payload}}{\text{Bandwidth}}$.

One of the most interesting findings is that all the models provide very different WCD equations across different runs and different test cases.

Ministral 3 8B output:

F0: 1160.0, F1: 1020.0, F2: 100.0, F3: 1160.0, F4: 1260.0, F5: 100.0, F6: 100.0, F7: 100.0, F8: 100.0, F9: 100.0, F10: 100.0, F11: 100.0, F12: 1160.0, F13: 100.0, F14: 100.0, F15: 100.0, F16: 100.0, F17: 1260.0, F18: 100.0, F19: 1160.0

Model Reasoning and Explanation Output:

$$\begin{aligned}
\text{Hypercycle:} \quad H &= 10000\ \mu\text{s}, \\
\text{Max frame size} &= 1526\ \text{bytes} = 1500 + 26\ \text{bytes}, \\
\text{Max transmission time} &= 12.208\ \mu\text{s}, \\
\text{Max queuing delay} &= 48\ \mu\text{s}, \\
WCD &= h\left(T_{\text{prop}} + T_{\text{switch}} + T_{\text{sync}}\right) \\
&\quad + T_{\text{tx,max}} \left\lceil \frac{\text{payload}}{\text{max_frame_size}} \right\rceil \\
&\quad + T_{\text{queue,max}}(h-1),
\end{aligned}$$

Table 35: CQF Error Analysis Case 1: Lack of Specific Knowledge. (continued)

$$\text{Hop count:} \quad h = 5,$$

$$\text{Propagation + switching + sync} = 5 \times (1 + 1 + 1) = 15\ \mu\text{s},$$

$$\text{Transmission time} = 12.208 \times \left\lceil \frac{2500}{1526} \right\rceil = 12.208 \times 2 = 24.416\ \mu\text{s},$$

$$\text{Queueing delay} = 48 \times (5 - 1) = 192\ \mu\text{s},$$

$$\text{Total} = 15 + 24.416 + 192 = 231.416\ \mu\text{s},$$

$$\text{Adjusted worst-case} = 1160\ \mu\text{s}.$$

Expert Explanation:

The WCD equation provided by the model is wrong. Although the model considers the number of hops in the route and the delays accumulated at each hop, and also calculates the hop count, it misses the most crucial term of the WCD equation, which is the cycle duration. Furthermore, the two components of the WCD equation ($T_{\text{tx,max}} \left\lceil \frac{\text{payload}}{\text{max_frame_size}} \right\rceil$) and ($T_{\text{queue,max}}(h-1)$) considered by the model are entirely hallucinated. These two components are mainly contributing to the large WCD values of this model.

H Failure Mode Analysis

To understand the nature of WCD computation failures, we identify three distinct failure modes observed across models and mechanisms. Some models show more than one of them. Tables 34 and 35 show example CBS and CQF outputs with expert explanations.

Trivial Zero Failure: The model returns WCD = 0 for all flows, producing a structurally valid JSON response but with no computational content. This failure mode affects GPT-4o for 51 of the 100 CBS TCs in the CB setting, leaving 49 valid TCs, DeepSeek-V3.2 (Non-Thinking) on CBS, and Llama 3.2 1B in about half of its CB runs for CBS and CQF. It also occurs in 18% of the CB runs of GPT-4o mini and 30% of those of Llama 3.3 70B on CBS. This suggests these models recognize the output format requirement but do not carry out the WCD computation in these responses.

Invalid Flow Identifiers: The model returns WCD values for flows that do not exist in the TC. Such runs are discarded by the flow-ID rule (Appendix G.5). Llama 3.2 1B lists up to 1,013 invented flows (F0 to F1012) on CBS and CQF instead of the flows of the TC and returns WCD = 0 for all of them. For Qwen3 8B, the rule discards 209 CB runs on CBS, which leaves 24 valid CBS TCs. On CQF, all 36 of its answered CB runs use flow IDs that are not in the TC. For Grok 4.1 Fast, the rule discards 36 CB runs on CBS and 45 on CQF.

Missing or Truncated Output: The model returns no answer, or an answer that ends before the WCD values. DeepSeek-V3.2 (Thinking), evaluated in the CB setting only, returns empty responses for all CBS TCs and all but one CQF TC. Grok 4.1 Fast returns truncated JSON with flow profile metadata but no WCD values, and has only 48 valid CBS TCs. Llama 3.2 1B returns no parsable WCD value in 56 of its 123 CB runs on CBS. For Qwen3 8B, 16 of its 300 CB runs on CBS and 87 of its 123 CB runs on CQF failed at the inference endpoint, most of them with a gateway timeout. As a result, neither Qwen3 8B nor Llama 3.2 1B reaches the 50 valid TCs required for reporting on the synthetic topologies in the CB setting with the original prompts. Qwen3 8B is therefore reported in

the FC setting (97 valid CBS TCs and 57 valid CQF TCs), for the CBS Variant B prompt (55 valid TCs), and for the CQF Variant A and B prompts (55 and 100 valid TCs), but not in the CB setting.