mathematics

MDPI

Article

Performance Analysis of Switch Buffer Management Policy for Mixed-Critical Traffic in Time-Sensitive Networks

Ling Zheng [1,2,*], Yingge Feng [1], Weiqiang Wang [1] and Qianxi Men [1]

[1] School of Communication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China; fengyg@stu.xupt.edu.cn (Y.F.); 1556517203@stu.xupt.edu.cn (W.W.); mqx@stu.xupt.edu.cn (Q.M.)

[2] Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts and Telecommunications, Xi'an 710071, China

* Correspondence: lingzheng@xupt.edu.cn

Abstract: Time-sensitive networking (TSN), a cutting-edge technology enabling efficient real-time communication and control, provides strong support for traditional Ethernet in terms of real-time performance, reliability, and deterministic transmission. In TSN systems, although time-triggered (TT) flows enjoy deterministic delay guarantees, audio video bridging (AVB) and best effort (BE) traffic still share link bandwidth through statistical multiplexing, a process that remains nondeterministic. This competition in shared memory switches adversely affects data transmission performance. In this paper, a priority queue threshold control policy is proposed and analyzed for mixed-critical traffic in time-sensitive networks. The core of this policy is to set independent queues for different types of traffic in the shared memory queuing system. To prevent low-priority traffic from monopolizing the shared buffer, its entry into the queue is blocked when buffer usage exceeds a preset threshold. A two-dimensional Markov chain is introduced to accurately construct the system's queuing model. Through detailed analysis of the queuing model, the truncated chain method is used to decompose the two-dimensional state space into solvable one-dimensional sub-problems, and the approximate solution of the system's steady-state distribution is derived. Based on this, the blocking probability, average queue length, and average queuing delay of different priority queues are accurately calculated. Finally, according to the optimization goal of the overall blocking probability of the system, the optimal threshold value is determined to achieve better system performance. Numerical results show that this strategy can effectively allocate the shared buffer space in multi-priority traffic scenarios. Compared with the conventional schemes, the queue blocking probability is reduced by approximately 40% to 60%.

Keywords: time-sensitive network; queuing model; two-dimensional Markov chain; buffer management

MSC: 68M20; 68M10

check for updates

Academic Editors: Farag M. Sallabi, Mohammad Hayajneh and Marjan Mernik

Received: 24 August 2025
Revised: 16 October 2025
Accepted: 25 October 2025
Published: 29 October 2025

Citation: Zheng, L.; Feng, Y.; Wang, W.; Men, Q. Performance Analysis of Switch Buffer Management Policy for Mixed-Critical Traffic in Time-Sensitive Networks. *Mathematics* **2025**, *13*, 3443. https://doi.org/10.3390/math13213443

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Real-time communication is becoming increasingly critical in fields such as industrial automation, intelligent transportation, and healthcare. These fields impose stringent requirements on end-to-end delay and jitter in network transmission. Devices must not only achieve high-speed data transmission and immediate control responses but also ensure precise clock synchronization and high inter-device coordination. Typically, industrial

control systems require transmission delays below 10 ms, while autonomous driving demands latencies under 1 ms.

However, the maximum network latency of traditional Ethernet [1–3] can typically reach the level of hundreds of milliseconds, which is far from meeting the stringent requirements of these fields. The introduction of time-sensitive networking (TSN) [4–10] significantly enhances network transmission real-time performance and addresses the urgent need for low-latency communication in industrial automation. Building on traditional Ethernet, TSN delivers microsecond-level deterministic services via clock synchronization, data scheduling, and network configuration mechanisms, ensuring real-time communication and collaboration among automation devices to optimize production processes and enhance product quality. Simultaneously, advancements in autonomous driving technology have expanded TSN's potential in intelligent transportation, enabling precise communication support for autonomous vehicles and ensuring collaborative operations and safe driving among vehicles. In healthcare, TSN enables monitoring and regulation of medical devices, facilitates efficient coordination and information flow among them, and ultimately enhances patient safety.

TSN supports three types of mixed-critical traffic: time-triggered (TT) [11–13], audio video bridging (AVB) [14,15], and best-effort (BE) [16]. As shown in Figure 1, the traffic transmission and processing architecture of the TSN switch is explicitly designed in line with TSN core standards, ensuring seamless integration with existing scheduling and shaping mechanisms. The transmission and scheduling logics of the three types of traffic form a clear and collaborative system within this architecture. Among them, the highest-priority TT traffic for scenarios such as industrial control and periodic task transmission is allocated to an independent transmission partition, with its scheduling strictly interfaced with the time-aware shaper (TAS) technology [17–21] proposed in the IEEE 802.1Qbv [22,23] standard: through a pre-configured gate control list (GCL) [24–26], it periodically adjusts the on/off state of the output queue to achieve deterministic gating operations, ensuring TT traffic is transmitted within preset time windows and complying with the precise clock synchronization and scheduling requirements of this standard. Relying on this highest priority and sophisticated time synchronization and scheduling mechanism, TT traffic can effectively support periodic data transmission, perfectly matching the characteristics of periodic fixed operations in automated equipment. For example, in industrial scenarios, whether it is the regular movement of a robotic arm or the periodic processing of a production line, TT traffic provides accurate and reliable guarantees to ensure that these periodic tasks are performed accurately. Therefore, the transmission and scheduling strategy for TT traffic has been widely studied. However, in the GCL gating control method at the core of TAS technology, the offset difference between similar windows of adjacent nodes will affect network performance. Thus, Reference [27] proposed an optimized flexible window overlapping scheduling algorithm, which optimizes TT window offset based on delay evaluation while considering the overlap between windows of different priorities on the same node. In addition, under the condition of a simple network architecture, Reference [28] proposed an adaptive cycle compensation scheduling algorithm based on traffic classification. This study combines path selection and scheduling algorithms to improve scheduling success rate and reduce execution time, further optimizing the transmission of TT traffic. In addition, for AVB traffic that requires stable quality of service (QoS) guarantees, traffic shaping is accomplished in this architecture by integrating the IEEE 802.1Qav credit-based shaper (CBS) mechanism. This can effectively smooth traffic bursts, regulate transmission rates, and ensure compliance with preset bandwidth constraints. Meanwhile, AVB traffic shares the same transmission partition with BE traffic, which has the lowest priority and no strict latency requirements, and both are scheduled to transmit

during the transmission gaps of TT traffic. This design can fundamentally avoid resource competition between AVB, BE traffic, and TT traffic, and achieve maximum utilization of link bandwidth without compromising the deterministic transmission guarantees of TT traffic. Existing research demonstrates that TT traffic scheduling has been thoroughly investigated, ensuring high reliability and low latency during transmission. In contrast, the transmission optimization techniques for the event-triggered (ET) [29] traffic types, AVB and BE, are relatively less studied.

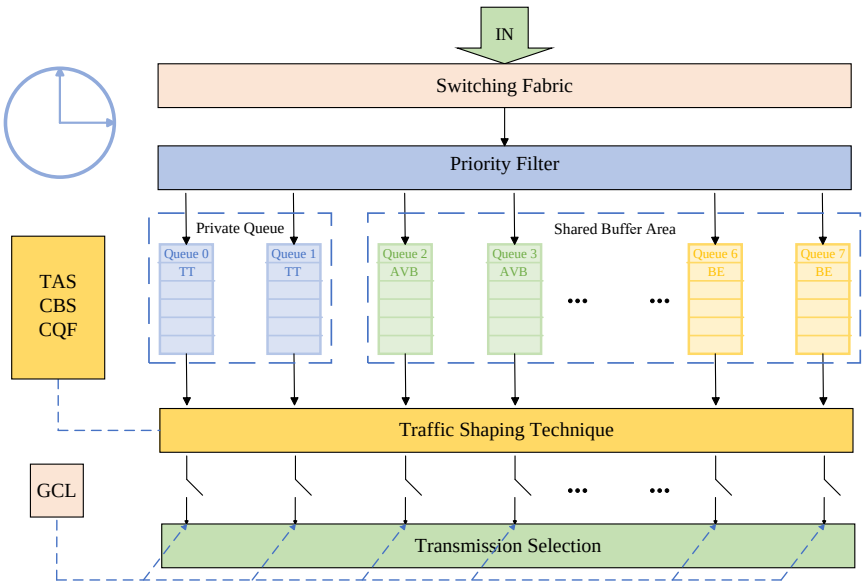


Figure 1. Traffic transmission processing architecture of TSN switch.

Recent representative studies on mixed traffic scheduling in the TSN domain have further provided important references for the advancement of this field. The OHDSR+ model proposed in [30] focuses on the joint scheduling and routing optimization of TT traffic and BE traffic. It enhances the reliability of TT traffic through redundant routing and time offset mechanisms, with its core innovation residing in the collaborative optimization of paths and scheduling under the constraint programming (CP) framework, and particularly emphasizes scalability in large-scale networks. As the precursor work of OHDSR+, Reference [31] first proposed the queue-gating time (QGT) mechanism, which precisely controls the transmission windows of TT and BE traffic via the GCL. Its core contribution lies in the establishment of a CP-based joint optimization framework for mixed traffic, verifying the feasibility of scheduling low-priority traffic. Both studies revolve around the top-level collaboration of "scheduling-routing" and provide effective solutions for the transmission optimization of TT traffic and low-priority traffic. However, they fail to delve into the resource competition issue at the shared buffer level. Even if temporal isolation of traffic is achieved through window scheduling, AVB and BE traffic in the shared buffer may still exhibit resource monopoly due to priority differences, thereby impairing transmission performance.

To address shared buffer unfairness, priority queue technology has been extensively studied as a core solution, with abundant research outcomes forming a mature theoretical and practical system. In the context of TSN switch buffer management, numerous studies have proposed targeted optimization schemes based on priority queue mechanisms. For example, Ref. [32] proposed a time-triggered switch-memory-switch architecture, which supports efficient transmission of time-triggered traffic through dedicated queue design and provides architectural references for the coordinated management of priority queues and buffer resources. Literature [33] introduced a dynamic per-flow queues (DFQ) method

in shared buffer TSN switches, improving the efficiency and determinism of time-triggered traffic transmission through refined queue partitioning and priority adaptation. Ref. [34] designed a flexible switching architecture with virtual queues, realizing logical isolation of different priority traffics via virtual queues and optimizing the allocation and scheduling of buffer resources. Ref. [35] proposed a size-based queuing (SBQ) approach, which ensures the transmission performance of high-priority traffic while improving bandwidth utilization through differentiated length management of priority queues. Ref. [36] combined virtual queues and time offset parameters to design a scheduling strategy, which reduces traffic interference and ensures low latency and high reliability of high-priority traffic through precise control of priority queues without increasing excessive hardware overhead.

Building on the aforementioned priority queue-related research, the dynamic threshold (DT) strategy [37] stands out for its high adaptability. However, the DT strategy struggles to achieve efficient buffer utilization, as it necessitates reserving space in the shared buffer. Building on the dynamic threshold management strategy, Ref. [38] proposed an efficient thresholding-based filter buffer management scheme, which compares the queue length of the output port with a specific buffer allocation factor, classifies the output port as active or inactive, and filters out the incoming packets of the inactive output port when the queue length exceeds a threshold. It can reduce the overall packet loss rate of the shared buffer packet switch and improve the fairness of buffer usage. Ref. [39] proposes a new strategy combining evolutionary computing and fuzzy logic. The core idea of this strategy is to adaptively adjust the threshold of each logic output queue through a fuzzy reasoning system to match the actual network traffic conditions. Ref. [40] proposes a double threshold method considering traffic scenarios, simplifies the calculation formula of traffic intensity, and dynamically adjusts the threshold value of each port to maintain common traffic intensity, maintain system equilibrium, and achieve global optimization. Refs. [41,42] present an enhanced dynamic threshold (EDT) strategy, mitigating micro-burst-induced packet loss via buffer optimization and temporary fairness relaxation.

Recently, due to the popularity of artificial intelligence algorithms, a series of AI-based buffer management strategies have been proposed. Refs. [43,44] propose a traffic-aware dynamic threshold (TDT) strategy, dynamically adjusting buffer allocation based on real-time port traffic detection. The wide application of machine learning [45] technology promotes its combination with network traffic control, and realizes the intelligence and refinement of traffic management through accurate analysis and prediction of traffic patterns. Ref. [46] proposed a buffer management strategy method based on deep reinforcement learning, developed corresponding domain-specific technologies to successfully apply deep reinforcement learning to the buffer management problem of network switches, and verified its effectiveness and superiority through experiments. However, these methods demand complex training processes and significant computational resources, posing challenges for deployment in high-speed switching environments.

This paper proposes a switch buffer management strategy for mixed-critical traffic in TSN. The core idea is to configure the output port of the TSN switch with multiple priority queues. A dedicated highest-priority queue is allocated for TT traffic, while the remaining queues are organized into a two-tier priority system. The high-priority queue is used to transmit AVB traffic, and the low-priority queue is used to transmit BE traffic. TT traffic occupies an independent buffer space inaccessible to other traffic types. Two types of the ET traffic share the remaining buffer space. To guarantee quality of service (QoS) for AVB traffic, a priority threshold is defined for lower-priority queues. When the shared buffer's total queue length exceeds the preset threshold, only high-priority AVB frames are admitted, while low-priority BE frames are blocked. Existing low-priority frames in the buffer are transmitted following default egress rules. This ensures the real-time

transmission of AVB traffic, reduces the delay of BE traffic, and reduces the overall blocking probability of the system.

Using a two-dimensional Markov chain [47,48] as the state space, this paper analyzes the performance of the priority queue threshold control strategy for mixed-critical traffic in TSN switches with shared buffers. Firstly, the two-dimensional Markov chain model is used to analyze and deduce the system performance, through metrics such as queue blocking probability, average queue length, and average queue delay. Compared to conventional one-dimensional methods, this method can greatly enhance the richness and detail of information in the queueing system, and show significant advantages in analytical precision. Secondly, the system model is analyzed through closed-form solutions of key performance indicators under multi-parameter conditions, exploring how parameters govern overall system behavior. Finally, mathematical tools establish a performance analysis model to uncover critical challenges in optimizing mixed-critical traffic performance. Based on this analysis, the optimal priority threshold for enhancing mixed-critical traffic transmission efficiency in time-sensitive networks is determined.

The remaining parts of the paper are organized as follows. Section 2 gives the system model and provides the detailed process of the analysis. Section 3 provides the numerical analysis results. Finally, Section 4 concludes the paper.

2. System Model and Analysis

This section systematically models the transmission process of mixed-critical traffic at the output port of the TSN switch. First, the necessity of introducing the priority threshold mechanism into the system is analyzed. Then, a two-dimensional Markov chain is used as a mathematical tool to thoroughly discuss the system performance. Finally, through detailed calculations, the steady-state distribution of the system is derived, laying a theoretical foundation for system performance analysis and optimization in subsequent sections.

2.1. System Model

In the TSN switch, once a data frame arrives, it is directly buffered in the preset priority queue and waits for egress scheduling. The switch follows an established scheduling policy for dequeuing to ensure prioritized transmission of data frames in high-priority queues. The TSN network handles three traffic types: TT traffic occupies a private queue with a dedicated scheduling algorithm to guarantee transmission determinism, while the remaining queues transmit AVB and BE traffic. To optimize resource allocation, we consolidate these remaining queues into a two-priority system: a high-priority queue for AVB traffic and a low-priority queue for BE traffic, which share the memory space. Thus, the AVB and BE queues can be modeled as a shared-buffer switch supporting QoS differentiation [49,50].

The queuing model of the TSN switch with multiple priority output queues is shown in Figure 2. The size of the shared buffer space of the entire output queue is B. The packet arrival rate of the high-priority queue is a Poisson process with rate λ_1, and the service time follows an exponential distribution with parameter $1/\mu_1$. Similarly, the packet arrival rate of the low-priority queue is a Poisson process with rate λ_2, and the service time follows an exponential distribution with parameter $1/\mu_2$. In addition, n_1 indicates the length of the high-priority queue, n_2 indicates the length of the low-priority queue, $\rho_1 = \lambda_1/\mu_1$ indicates the load of the high-priority queue, and $\rho_2 = \lambda_2/\mu_2$ indicates the load of the low-priority queue. A priority threshold with value T is set in the system. If the total queue length $n = n_1 + n_2$ exceeds T, low-priority traffic is blocked from entering the system.

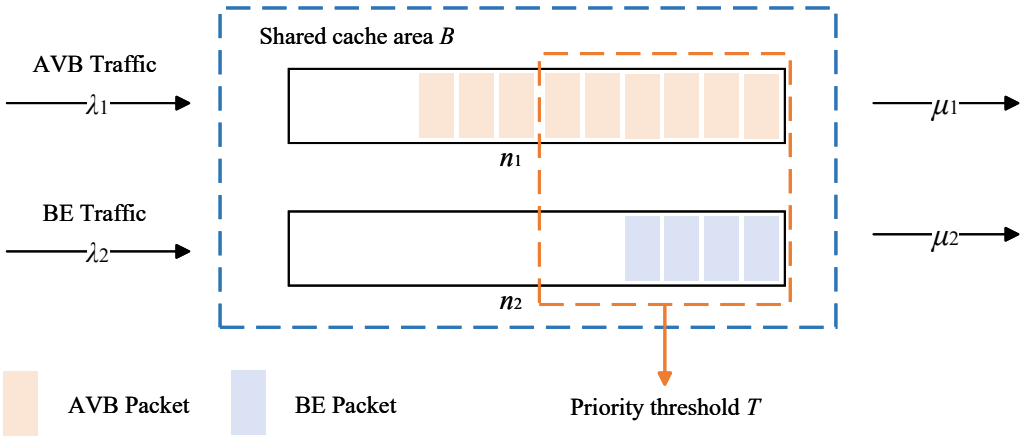


Figure 2. System queuing model.

The system can be modeled as a two-dimensional Markov chain, whose state space can be expressed as (n_1, n_2). The steady-state distribution of the system is

$$P(n_1, n_2) = \frac{\rho_1^{n_1} \rho_2^{n_2}}{G} \tag{1}$$

where G is a normalization constant that ensures $P(n_1, n_2)$ is a probability distribution

$$G = \sum_{(n_1, n_2) \in S} \rho_1^{n_1} \rho_2^{n_1} \tag{2}$$

2.2. Buffer Occupancy Without Priority Threshold

In this model, the high-priority queue is assigned a higher service rate than the low-priority queue, thus high-priority queues are dequeued faster than low-priority queues. When the shared buffer lacks a priority threshold, after the system reaches steady state, it can be expected that low-priority data frames will accumulate in the shared buffer, reducing the available buffer space for high-priority packets. Consequently, the transmission of high-priority packets is severely degraded. Additionally, the queuing delay of the low-priority queue will significantly increase. While high-priority data frames are designed to achieve low latency and high service quality, a shared buffer without a priority threshold directly contradicts this objective.

Here, we will illustrate the necessity of setting a priority threshold for low-priority queues through two parts of numerical analysis.

In the first part of numerical analysis, we analyze the occupancy ratio of low-priority data frames in the shared buffer (without a priority threshold) when the total queue length reaches the maximum threshold B. When the shared buffer is full (i.e., $n_1 + n_2 = B$), combined with the analysis method for the steady-state distribution of Markov chains in [51], the steady-state distribution of the system is:

$$\begin{aligned} P(n_1, n_2 | n_1 + n_2 = B) &= \frac{\rho_1^{n_1} \cdot \rho_2^{n_2}}{\Sigma_{n_1+n_2=B} \rho_1^{n_1} \cdot \rho_2^{n_2}} \\ &= \frac{\rho_1^{B-n_2} \cdot \rho_2^{n_2}}{\sum_{n_2=0}^{B} \rho_1^{B-n_2} \cdot \rho_2^{n_2}} \\ &= \frac{\rho_2^{n_2} / \rho_1^{n_2}}{\left(1 - (\rho_2/\rho_1)^{B+1}\right) / (1 - \rho_2/\rho_1)} \\ &= \frac{\rho_1^{B-n_2+1} \rho_2^{n_2} - \rho_1^{B-n_2} \rho_2^{n_2+1}}{\rho_1^{B+1} - \rho_2^{B+1}}. \end{aligned} \tag{3}$$

Then, the average queue length of low priority in the shared buffer is

$$E(n_2|n_1+n_2=B)=\sum_{n_2=0}^{B} n_2 \cdot P(n_1,n_2|n_1+n_2=B). \tag{4}$$

When no priority threshold is applied, the relationship between the low-priority average queue length and the proportion of low-priority queue load is shown in Figure 3, where the proportion of low-priority queue load is $\rho_2/(\rho_1+\rho_2)$ and the shared buffer space is $B=10$. As can be seen from the figure, the low-priority average queue length increases with the proportion of low-priority loads. The results indicate that without controlling the enqueuing of low-priority traffic, a large amount of ET traffic flooding into the system causes low-priority BE traffic to occupy significant shared buffer space due to slow dequeueing speed, thereby drastically reducing the buffer space available for high-priority AVB traffic and degrading the transmission of AVB traffic with higher real-time requirements.

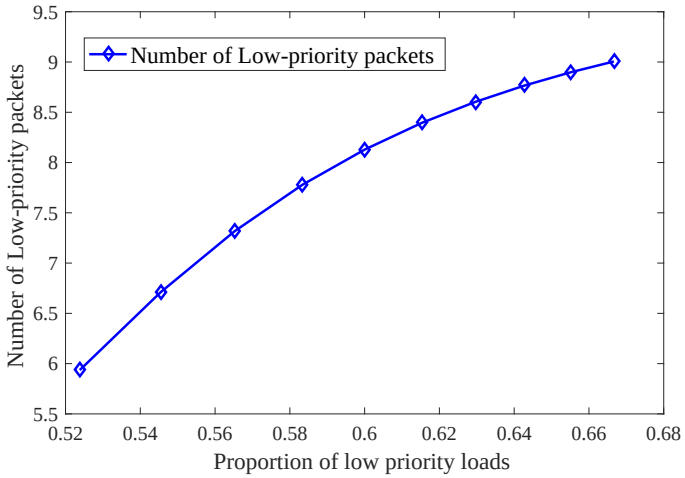


Figure 3. Relationship between low-priority queue length and its load ratio.

In the second part of the numerical analysis, we analyzed the average queuing delay of high-priority and low-priority queues without a priority threshold, and obtained the numerical analysis results shown in Figure 4. When no priority threshold is applied and the whole shared buffer is completely shared by high- and low-priority data frames, the average queue length of the high-priority queue is Q_{High} and the average queue length of the low-priority queue is Q_{Low}, which is expressed as follows:

$$\begin{aligned} Q_{High} &= E(n_2|n_1+n_2=B) \\ &= \sum_{n_2=0}^{B} n_2 \cdot P(n_1,n_2|n_1+n_2=B,) \end{aligned} \tag{5}$$

$$\begin{aligned} Q_{Low} &= B - Q_{High} \\ &= B - \sum_{n_2=0}^{B} n_2 \cdot P(n_1,n_2|n_1+n_2=B.) \end{aligned} \tag{6}$$

According to Little's Law [52], the average queuing delay of the high-priority queue D_{High} and the average queuing delay of the low-priority queue D_{Low} can be expressed as follows:

$$D_{High} = Q_{High}/\lambda_1 \tag{7}$$

$$D_{Low} = Q_{Low}/\lambda_2 \tag{8}$$

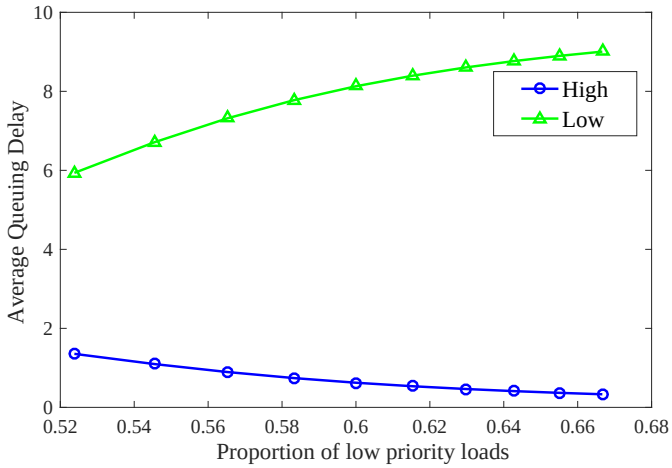


Figure 4. Relationship between the average queuing delay of each priority queue and the proportion of low-priority load.

Figure 4 illustrates the relationship between the average queuing delay of the two-priority queues and the proportion of low-priority load. As shown in the figure, the average queuing delay of low-priority queues increases as the proportion of low-priority load rises. The results demonstrate that without restricting low-priority data frames' access to the shared buffer, their queuing delay increases dramatically, reaching values tens of times higher than those of high-priority queues.

In summary, setting a priority threshold for low-priority queues to regulate their access to the shared buffer is both critically important and theoretically justified.

2.3. Analysis of Priority Threshold Mechanism

In this section, we analyze the behavior of a low-priority queue with a priority threshold T. A two-dimensional Markov chain is employed to model the queuing system, where the state space (n_1, n_2) represents the lengths of the high-priority queue n_1 and the low-priority queue n_2 within the shared buffer.

As shown in Figure 5, the horizontal axis represents state transitions of high-priority data frames, and the vertical axis represents state transitions of low-priority data frames. When the total queue size is $n < T$, both low- and high-priority data frames can be queued in the buffer. When the total queue size is $T < n < B$, the low-priority data frame cannot enter the buffer queue, and the low-priority data frame already in the buffer is dequeued normally according to the original dequeuing rule. When the total queue length is $n = B$, all data frames cannot enter the buffer.

Based on the queuing rules described above, key system performance metrics—including queue blocking probability, average queue length, and average queue delay—are analyzed and derived through a two-dimensional Markov chain model. By calculating the system's steady-state distribution and incorporating all state space behaviors, the blocking probability of each priority queue is derived. An average weighted packet loss rate is then introduced to quantify the system's overall blocking probability. The average queue length for each priority queue is calculated using the marginal distribution probability of the queue length in the steady state. Similarly, the average delay for each priority queue is derived via Little's theorem based on the corresponding average queue length. These results demonstrate that determining the system's steady-state distribution is the critical step for computing its core performance metrics.

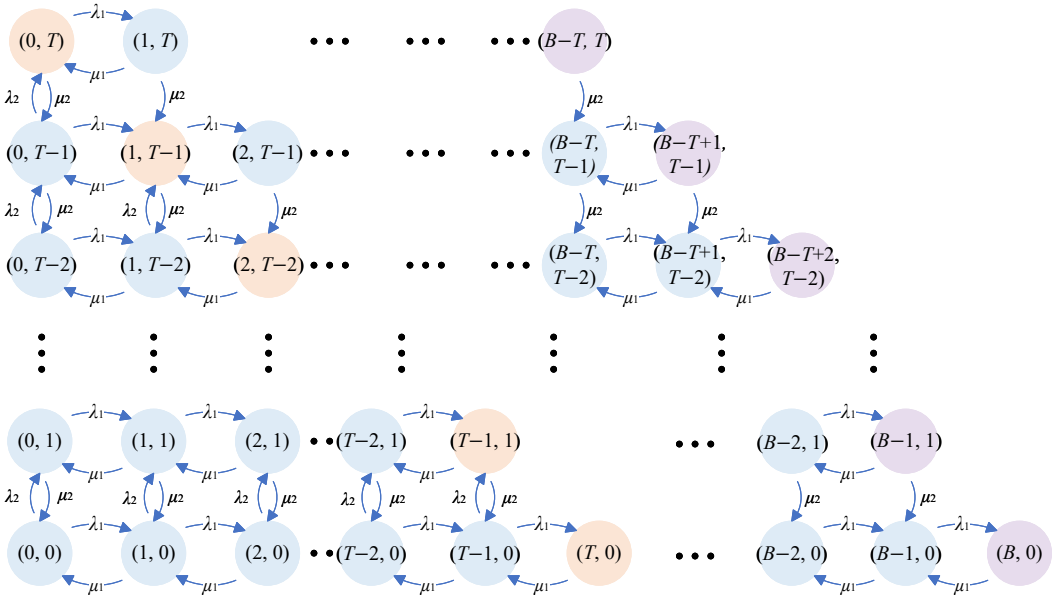


Figure 5. State transition diagram for two-dimensional Markov chain.

2.3.1. Steady-State Distribution of the System

Theoretically, the exact steady-state distribution of a two-dimensional Markov chain under a finite buffer size can be obtained by constructing a global balance equation system and solving the corresponding linear system. Taking a specific buffer size $B = 10$ as an example, $\pi(n_1, n_2)$ denotes the steady-state probability of the state (n_1, n_2). The system contains nearly 100 independent states in total. The corresponding linear system to be constructed must satisfy the global balance conditions, namely "the sum of steady-state probabilities of all states equals 1" and "the inflow probability of each state equals its outflow probability", and eventually forms a linear system with approximately 100 equations, as shown below:

$$\begin{cases} (\lambda_1 + \lambda_2)\pi(0,0) = \mu_1\pi(1,0) + \mu_2\pi(0,1) \\ (\lambda_1 + \lambda_2 + \mu_1)\pi(1,0) = \lambda_1\pi(0,0) + \mu_1\pi(2,0) + \mu_2\pi(1,1) \\ \quad \vdots \\ \mu_1\pi(B,0) = \lambda_1\pi(B-1,0) \\ \quad \vdots \\ (\lambda_1 + \mu_2)\pi(0,T) = \lambda_2\pi(0,T-1) + \mu_1\pi(1,T) \\ (\mu_1 + \mu_2)\pi(1,T) = \lambda_1\pi(0,T) \\ \quad \vdots \\ (\mu_1 + \mu_2)\pi(B-T,T) = \lambda_1\pi(B-T-1,T) \\ \sum_{n_1=0}^{B} \sum_{n_2=0}^{B-n_1} \pi(n_1, n_2) = 1 \end{cases} \tag{9}$$

However, although such a solution method based on the global balance equation system can yield a theoretical solution through numerical methods like Gaussian elimination, as the buffer size B increases, the number of system states grows quadratically. This directly leads to a sharp expansion in the scale of the linear system, which in turn causes severe computational complexity issues in the solution process. It not only requires a large amount of storage space to store the coefficient matrix and variable vectors of the equation system, but also demands

extremely high computational resources to complete matrix operations and iterative solving. Ultimately, this results in extremely low engineering feasibility for exact solutions.

Therefore, to solve the problem of finding the system's steady-state distribution of the two-dimensional Markov chain, this paper adopts the truncated chain method to achieve an efficient approximate solution. By doing so, it decomposes the two-dimensional state space (which originally contains two variables) into two one-dimensional state spaces, each containing only a single variable. The two-dimensional Markov chain is horizontally truncated into T one-dimensional truncated chains, each with a single variable n_1 as its state space. Subsequently, each of these truncated chains is regarded as a state point, and all truncated chains are synthesized into a single aggregated one-dimensional chain with a single variable n_2 as its state space. With the help of the properties of one-dimensional Markov chains, we can calculate the stationary probability U_{n_2} of a truncated chain in the aggregate chain and the stationary probability $\pi_{n_1}^{(n_2)}$ of a state of the truncated chain. By multiplication of the two, we can get the stationary probability of the state point (n_1, n_2) in the two-dimensional Markov chain. When the system reaches a stable state after a long period of operation, the stationary probability of the point is converted to the expected proportion of the point in the two-dimensional state chain and can be further approximated as the steady-state distribution of the system. Therefore, the problem of the system steady-state distribution of two-dimensional Markov chains is transformed into the problem of solving the stationary probability U_{n_2} of a certain truncated chain and the stationary probability $\pi_{n_1}^{(n_2)}$ of a certain state of the truncated chain in the aggregate chain.

Firstly, we solve for the stationary probability $\pi_{n_1}^{(n_2)}$ of a certain state of the truncation chain. The two-dimensional Markov chain is truncated horizontally, as shown in Figure 6, and T truncated chains with state space S_k are obtained.

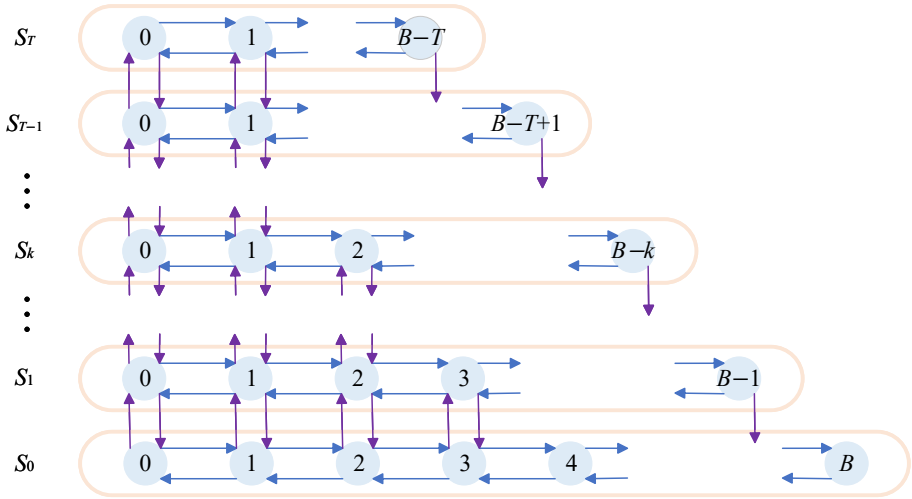


Figure 6. Truncation diagram of two-dimensional Markov chain.

Defining S_k as the set of states $(k,0),(k,1),...,(k,B-k)$, the horizontally truncated chain with state space S_k is shown in Figure 7. S_k is a one-dimensional Markov chain, with its state space determined by the length n_1 of a high-priority queue.

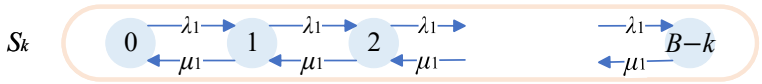


Figure 7. State transition diagram of the k-th truncated chain S_k.

We denote by $\pi_{\mathrm{j}}^{(\mathrm{k})}$ the stationary probability of state j of the truncated chain S_k. From the detailed balance equation and the equation indicating that the sum of stationary probabilities of all states is 1, we obtain

$$\begin{cases} \pi_j^{(k)} \cdot \lambda_1 = \pi_{j+1}^{(k)} \cdot \mu_1, \\ \sum_{j=0}^{B-k} \pi_j^{(k)} = 1. \end{cases} \tag{10}$$

Thus, we can obtain the stationary probability of any state within the truncated chain S_k as

$$\pi_j^{(k)} = \rho_1^j \cdot \pi_0^{(k)} = \rho_1^j \cdot \frac{1-\rho_1}{1-\rho_1^{B-k+1}}, k = 0, 1, ..., T, j = 0, 1, ..., B. \tag{11}$$

Secondly, we solve for the stationary probability U_{n_2} of a certain truncated chain in the aggregated chain. As shown in Figure 8, the state transition diagram of the aggregated chain for this system model is a one-dimensional Markov chain with only one state variable, which is generated by state transitions between T truncated chains. The state space variable of this aggregated chain is the length n_2 of the low-priority queue.

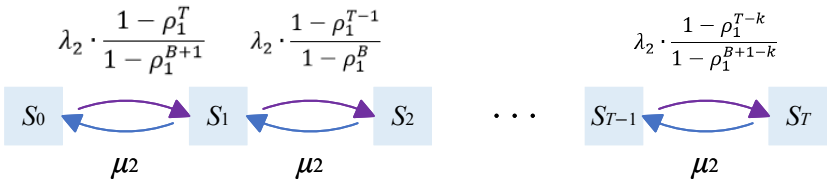


Figure 8. State transition diagram for aggregated chains.

We denote by $\tilde{\mathrm{q}}_{k+1,k}$ the transition probability from the truncated chain S_{k+1} to the truncated chain S_k, and by $\tilde{\mathrm{q}}_{k,k+1}$ the transition probability from the truncated chain S_k to the truncated chain S_{k+1}, which is also the transition probability of the aggregated chain. Then,

$$\begin{aligned} \tilde{\mathrm{q}}_{\mathrm{k+1,k}} &= \sum_{j \in S_{k+1}, i \in S_k} \pi_j^{(k+1)} \cdot \mathrm{q}_{ji} \\ &= \mu_2 \sum_{j \in S_{k+1}} \pi_j^{(k+1)} \\ &= \mu_2. \end{aligned} \tag{12}$$

$$\begin{aligned} \tilde{\mathrm{q}}_{\mathrm{k,k+1}} &= \sum_{j \in S_k, i \in S_{k+1}} \pi_j^{(k)} \cdot \mathrm{q}_{ji} \\ &= \sum_{j=0}^{T-k-1} \pi_j^{(k)} \cdot \mathrm{q}_{ji} \\ &= \lambda_2 \cdot \frac{1-\rho_1^{T-k}}{1-\rho_1^{B-k+1}}. \end{aligned} \tag{13}$$

where q_{ji} represents the one-step transition probability from the truncated chain in which state j is located to the truncated chain in which state i is located.

We denote by U_k the stationary probability of the truncated chain S_k within the aggregated chain. When the aggregated chain reaches its stationary state, we can establish equations based on the detailed balance equation and the constraint that the sum of the stationary probabilities of all truncated chains equals 1. Therefore, we have

$$\begin{cases} U_k \cdot \tilde{\mathrm{q}}_{\mathrm{k,k+1}} = U_{k+1} \cdot \tilde{\mathrm{q}}_{\mathrm{k+1,k}} \\ \sum_{k=0}^{T} U_k = 1. \end{cases} \tag{14}$$

Thus, we can obtain the stationary probability of any truncated chain within the aggregated chain as follows:

$$U_k = \frac{\rho_2^k \cdot \prod_{m=0}^{k-1} \frac{1-\rho_1^{T-m}}{1-\rho_1^{B-m+1}}}{\sum_{k=0}^{T} \rho_2^k \cdot \prod_{m=0}^{k-1} \frac{1-\rho_1^{T-m}}{1-\rho_1^{B-m+1}}}. \tag{15}$$

Finally, we solve for the steady-state distribution of the system. By multiplying the stationary probability U_{n_2} of a particular truncated chain within the aggregated chain by the stationary probability $\pi_{\mathrm{n_1}}^{(\mathrm{n_2})}$ of a specific state within that truncated chain, we can obtain the steady-state distribution of the entire system as follows:

$$\begin{aligned} P(n_1, n_2) &= \pi_{n_1}^{(n_2)} \cdot U_{n_2} \\ &= \frac{\rho_1^{n_1} \cdot \frac{1-\rho_1}{1-\rho_1^{B-n_2+1}} \cdot \rho_2^{n_2} \cdot \prod_{m=0}^{n_2-1} \frac{1-\rho_1^{T-m}}{1-\rho_1^{B-m+1}}}{\sum_{n_2=0}^{T} \rho_2^{n_2} \cdot \prod_{m=0}^{n_2-1} \frac{1-\rho_1^{T-m}}{1-\rho_1^{B-m+1}}}. \end{aligned} \tag{16}$$

2.3.2. Blocking Probability for Two Priority Queues

When the total queue length reaches the maximum threshold B, blocking occurs in the high-priority queue, and L_1 is used to denote the blocking probability of the high-priority queue. When the total queue length reaches the priority threshold T, blocking occurs in the low-priority queue, and L_2 is used to denote the blocking probability of the low-priority queue.

$$\begin{aligned} L_1 &= P(n_1 = B - n_2, 0 \le n_2 \le T) \\ &= \frac{\sum_{n_2=0}^{T} \rho_1^{B-n_2} \cdot \frac{1-\rho_1}{1-\rho_1^{n_1+1}} \cdot \rho_2^{n_2} \cdot \prod_{m=0}^{n_2-1} \frac{1-\rho_1^{T-m}}{1-\rho_1^{B-m+1}}}{\sum_{n_2=0}^{T} \rho_2^{n_2} \cdot \prod_{m=0}^{n_2-1} \frac{1-\rho_1^{T-m}}{1-\rho_1^{B-m+1}}}, \end{aligned} \tag{17}$$

$$\begin{aligned} L_2 &= 1 - P(0 \le n_1 \le T, 0 \le n_2 \le T - 1 - n_1) \\ &= 1 - \frac{\sum_{n_1=0}^{T-1} \sum_{n_2=0}^{T-1-n_1} \rho_1^{n_1} \cdot \frac{1-\rho_1}{1-\rho_1^{B-n_2+1}} \cdot \rho_2^{n_2} \cdot \prod_{m=0}^{n_2-1} \frac{1-\rho_1^{T-m}}{1-\rho_1^{B-m+1}}}{\sum_{n_2=0}^{T} \rho_2^{n_2} \cdot \prod_{m=0}^{n_2-1} \frac{1-\rho_1^{T-m}}{1-\rho_1^{B-m+1}}}. \end{aligned} \tag{18}$$

2.3.3. Average Queue Length and Delay

We need to obtain the average queue length and queue delay of the high-priority queue and the low-priority queue, respectively. The marginal distribution of the high-priority queue length is

$$P(n_1 = \mathrm{j}) = \begin{cases} \sum_{n_2=0}^{T} P(n_1, n_2), 0 \le j \le B - T \\ \sum_{n_2=0}^{B-j} P(n_1, n_2), B - T + 1 \le j \le B. \end{cases} \tag{19}$$

The marginal distribution of the low-priority queue length is

$$P(n_2 = \mathrm{j}) = \sum_{n_1=0}^{B-j} P(n_1, n_2), 0 \le j \le T. \tag{20}$$

After obtaining the marginal distributions of the queue lengths for each priority, we can then calculate the average queue lengths for the high-priority queue and the low-priority queue, respectively, using the following formulas:

$$\begin{aligned} \mathrm{E}[n_1] &= \sum_{j=0}^{B} j \times P(n_1 = \mathrm{j}) \\ &= \sum_{j=0}^{B-T} j \times \sum_{n_2=0}^{T} P(j, n_2) + \sum_{j=B-T+1}^{B} j \times \sum_{n_2=0}^{B-j} P(j, n_2), \end{aligned} \tag{21}$$

$$\begin{aligned} \mathrm{E}[n_2] &= \sum_{j=0}^{T} j \times \mathrm{P}(n_2 = \mathrm{j}) \\ &= \sum_{j=0}^{T} j \times \sum_{n_1=0}^{B-j} P(n_1, j). \end{aligned} \tag{22}$$

By applying Little's theorem, we can obtain the average delays for both the high-priority queue and the low-priority queue as follows:

$$D_1 = \mathrm{E}[n_1]/\lambda_1, \tag{23}$$

$$D_2 = \mathrm{E}[n_2]/\lambda_2. \tag{24}$$

2.3.4. Overall Blocking Probability

We introduce the system's average weighted packet loss rate as a measure of the overall blocking probability of the system, denoted as WL, as follows:

$$WL = \frac{w_1 L_1 + w_2 L_2}{w_1 + w_2}, \tag{25}$$

where w_1 is the weight of the high-priority queue and w_2 is the weight of the low-priority queue. From the above equation, it can be seen that the value of the system's overall blocking probability WL is related to the weights of the high- and low-priority queues. Additionally, the system's overall blocking probability WL may also be related to the input traffic. We aim to optimize the overall performance of the system, which requires minimizing the system's overall blocking probability WL. In this situation, the value of threshold T is the best priority threshold value.

3. Numerical Results

Based on the analysis and derivations in Section 2, this chapter will conduct numerical analysis on the main system performance indicators. Subsequently, by adjusting the relevant parameters that influence these performance indicators, the aim is to optimize system performance. On this foundation, the optimal priority threshold value is determined in order to attain optimal system performance. Finally, a comparative analysis is carried out between the proposed priority threshold management mechanism and other shared buffer management mechanisms to provide valuable references for network performance optimization in practical applications.

3.1. Accuracy Evaluation of the Truncated Chain Method

To systematically validate the accuracy and reliability of the truncated chain method adopted in this paper for approximating the steady-state distribution of the two-dimensional Markov chain, we specifically designed and conducted a series of baseline tests for scenarios with small buffer sizes. Buffer sizes of $B = 5$, $B = 6$, $B = 7$, and $B = 8$

were selected. Under the traffic scenario with a high-priority traffic load of $\rho_1 = 0.3$ and a low-priority traffic load of $\rho_2 = 0.6$, the system's steady-state probability distribution was solved via two distinct approaches. On one hand, we employed the classical Gaussian elimination method to solve the system's global balance equations exactly, thereby obtaining the precise steady-state distribution $P_exact(n_1, n_2)$ as the benchmark reference. On the other hand, the truncated chain method proposed in this paper was applied for computation, yielding the corresponding approximate steady-state distribution $P_approx(n_1, n_2)$. Figure 9 visually presents and compares the complete steady-state distribution results obtained from both methods across all state points, showing a high degree of consistency in both overall shape and local details.

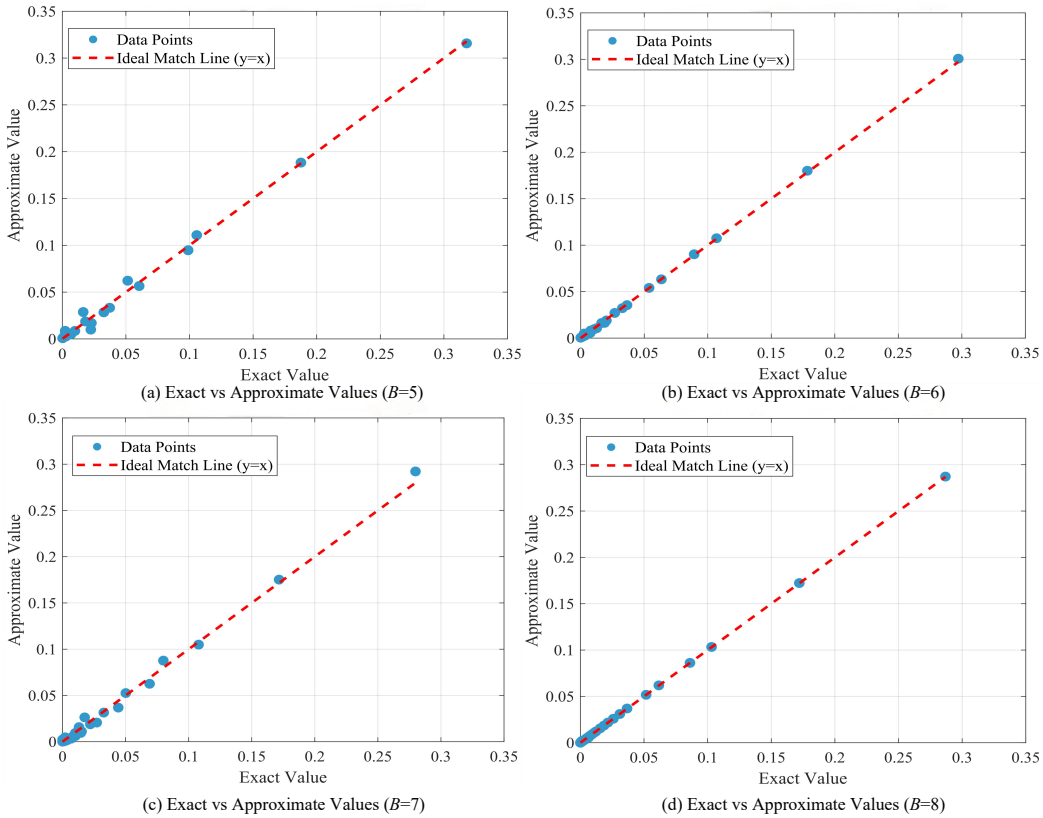


Figure 9. Comparison of exact and approximate steady-state distributions.

To further quantitatively evaluate the error level of the truncated chain approximation, Table 1 in this paper details three key performance metrics: root mean square error (RMSE), mean absolute error (MAE), and Pearson correlation coefficient (PCC). Analysis of these metrics reveals that for all tested buffer sizes, the RMSE and MAE values between the approximate and exact solutions remain stably controlled below 1%, while the PCC value consistently maintains an extremely high level above 99%. Therefore, the truncated chain method employed in this paper demonstrates high reliability, and the steady-state distribution of the two-dimensional Markov system obtained by this method can effectively substitute for the exact solution method, which involves extremely high computational complexity.

Table 1. Overall approximation metrics.

	$B = 5, T = 4$	$B = 6, T = 5$	$B = 7, T = 6$	$B = 8, T = 7$
RMSE	0.005535	0.001211	0.004508	0.000225
MAE	0.004029	0.000859	0.003174	0.000146
PCC	0.997394	0.999896	0.997396	0.999992

3.2. Analysis of Blocking Probability of Each Priority Queue

In Figure 10, the relationship between the blocking probabilities of different priority queues and the priority threshold factor β is illustrated under various proportions of low-priority load, where $\beta = T/B$. As can be seen from the figure, for any low-priority load ratio, as β rises, the blocking probability of the high-priority increases, while the blocking probability of the low-priority decreases. This is because when threshold T is small, low-priority packets will occupy less shared buffer space and leaving more space for high-priority packets. Consequently, the blocking probability for low-priority packets is higher, while the blocking probability for high-priority packets is lower at this point. As β continues to increase, so does T, causing low-priority packets to occupy more shared buffer space, leaving less space for high-priority packets. Therefore, at this stage, the blocking probability for low-priority packets decreases, while the blocking probability for high-priority packets increases.

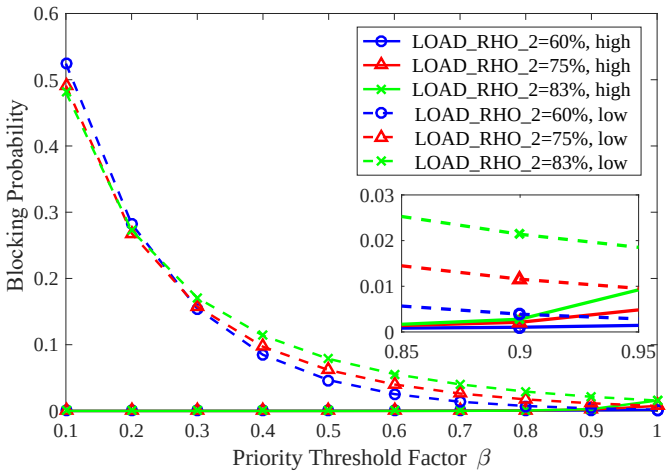


Figure 10. Relationship between blocking probability and priority threshold factor β for different priority queues.

We expect the blocking probability of the high-priority queue and the low-priority queue to be at a low level, but when T is small, it is beneficial to the transmission of the data frame of the high-priority queue. When T is large, it is beneficial for the transmission of low-priority queue data frames. Therefore, it is necessary to find an appropriate value of T, such that in this state, the high-priority and low-priority queue blocking probability are at a low level. It should be noticed that the blocking probability of the high-priority queue is always close to 0 until β is above 0.9. Therefore, it is intuitive to infer that the priority threshold should not be set too small.

3.3. Analysis of Average Queue Length and Delay

Figure 11 shows the relationship between the normalized low-priority queue length and the priority threshold factor β. The normalization of the low-priority queue length is defined as $\mathrm{E}[n_2]/(\mathrm{E}[n_1] + \mathrm{E}[n_2])$. For any low-priority load fraction, the fraction of low-priority packets rises as β rises. When β is increased to 1, that is, $T = B$, the low-priority group will occupy a large amount of shared cache space, especially for large traffic loads.

Figures 12 and 13 collectively illustrate the variation characteristics of the average delay of the low-priority queue with the priority threshold factor β. Specifically, Figure 12 adopts the parameter $B = 10$, while Figure 13 further expands the parameter range to $B = 50$ to improve parameter exploration. Consistent variation patterns can be observed from the statistical results of the two figures: first, as the priority threshold factor β increases, the average delay of the low-priority queue shows a significant and stable upward trend;

second, regardless of whether the parameter B is set to 10 or 50, the overall average delay of the low-priority queue increases accordingly when the proportion of low-priority traffic in the total network traffic rises.

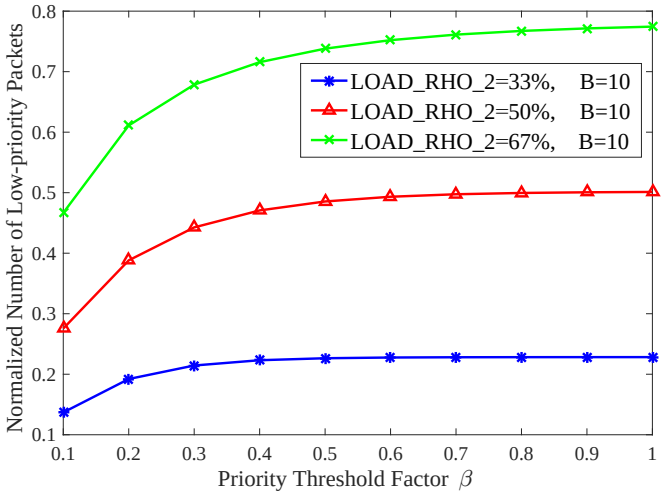


Figure 11. Relationship between normalized low-priority queue length and priority threshold factor β.

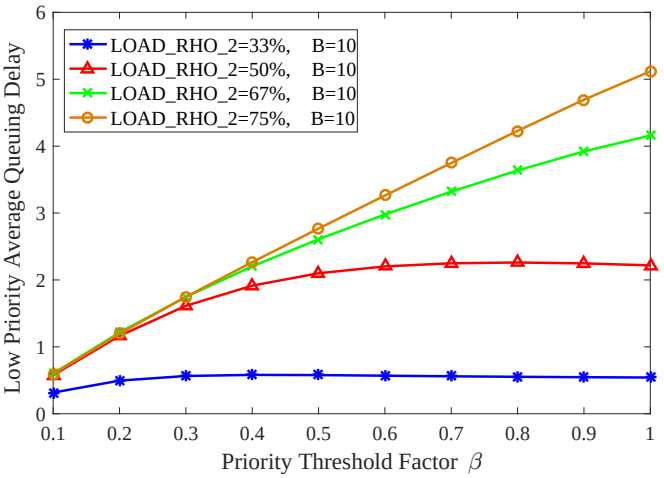


Figure 12. Relationship between average delay and priority threshold factor β for low-priority queues when $B = 10$.

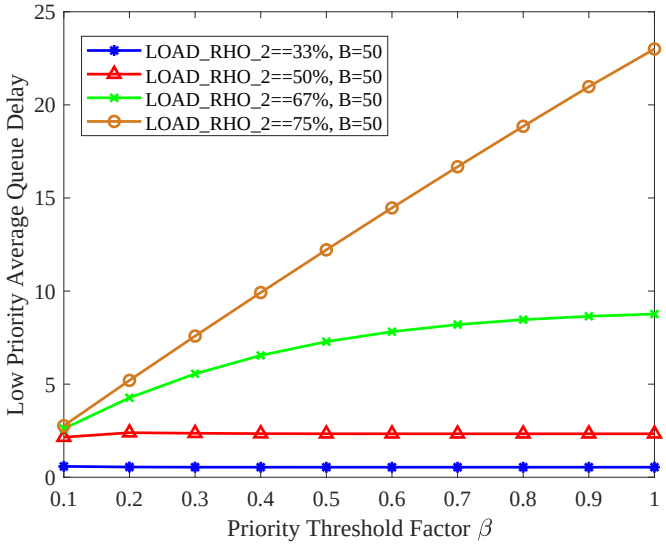


Figure 13. Relationship between average delay and priority threshold factor β for low-priority queues when $B = 50$.

By analyzing the above numerical analysis results, we have concluded the necessity of setting an appropriate threshold value for the low-priority queue. This setting not only ensures that high-priority data frames receive priority processing and efficient transmission, thereby guaranteeing the timely delivery of critical information, but also creates a relatively stable transmission environment for low-priority packets. Even when resource allocation is tight, through a reasonable cache allocation mechanism, low-priority packets can still obtain the necessary transmission opportunities. As a result, it effectively reduces the phenomena of long waiting times or packet losses caused by resource competition, thereby enhancing the transmission efficiency and user experience of the entire network system.

3.4. Optimal Priority Threshold Value

To evaluate the system's overall blocking probability and determine the optimal threshold value, we perform two numerical analyses. In the first numerical analysis, we will explore the impact of adjusting the weight allocation between high- and low-priority queues on the overall blocking probability of the system under the condition of constant input traffic. According to the actual system behavior, we know that the weight of the high-priority queue should always be greater than the weight of the low-priority queue, that is, $w_1 > w_2$.

Figure 14 shows the different priority weights, the system weighted average blocking probability varying with threshold factor β. Under the condition that the proportion of low-priority load is 75%, the experiment sets the low-priority queue weight $W_2 = 1$ and adjusts the high-priority weight W_1 to 2, 3, 4, and 5, respectively, and the overall blocking probability WL of the system varies with the threshold factor β. It can be seen from the figure that the average weighted packet loss rate of the system is smaller when the weight of the higher priority is larger under the same input traffic. As a whole, in the range of $0.7 \leq \beta \leq 0.9$, the average weighted blocking rate reaches the minimum, so the threshold $T = \beta \cdot B$ is obtained in this interval. Since the importance of high-priority services is to be higher, T should be as small as possible.

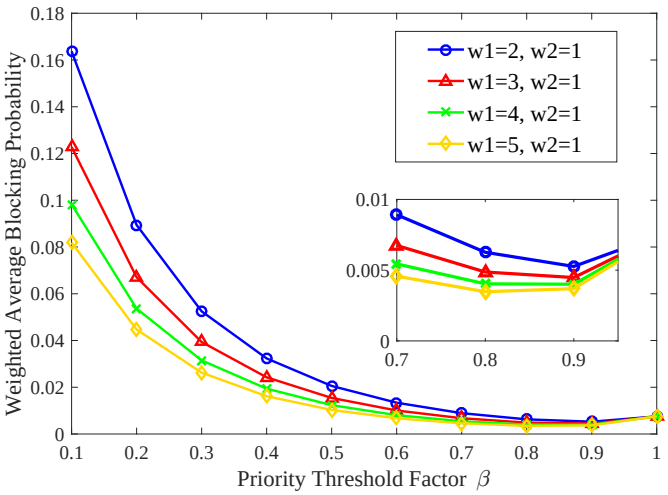


Figure 14. Relationship between the weighted average blocking probability WL and the threshold factor β for different weights.

In the second numerical analysis, under the condition of constant high- and low-priority weights, the input traffic of the system is changed by adjusting the proportion of low-priority load to explore how the input traffic would affect the blocking probability of the system.

As shown in Figure 15, two queues are set with weights $w_1 = 5$ and $w_2 = 1$, respectively. This figure illustrates the relationship between the overall system blocking probability WL and the threshold factor β under different traffic loads. From the figure, it can be observed that for any proportion of low-priority load, the overall system blocking probability decreases as the threshold factor β increases, reaching a minimum value before subsequently increasing with further increases in β. Moreover, within the range of approximately $0.7 \leq \beta \leq 0.9$, the average weighted packet loss rate reaches its minimum, and thus the threshold $T = \beta \cdot B$ is obtained within this interval.

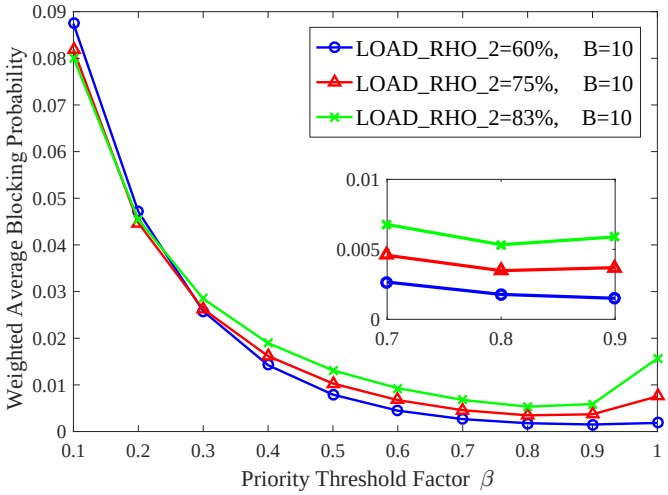


Figure 15. Relationship between the overall blocking probability of the system and the threshold factor under different traffic loads.

From the above two numerical analyses, it can be seen that the optimal value of the priority queue threshold T is approximately obtained at $0.7 \leq \beta \leq 0.9$ for various network parameters. Changing the weight settings or the proportion of network traffic does not affect the range of the optimal threshold. It shows that the priority threshold mechanism has good stability for all kinds of network traffic conditions.

3.5. Comparisons for Different Buffer Management Mechanisms

To verify the performance of the priority queue threshold control (PQTC) mechanism in time-sensitive networking (TSN) scenarios, this section includes five types of queue management mechanisms for comparative analysis, namely complete sharing weighted fair queuing (CS-WFQ), complete sharing with strict priority (CS-SP), static threshold (ST), shared-private queue management (SPQM), and the proposed PQTC mechanism.

The experimental parameter configuration matches the characteristics of TSN mixed-critical traffic: the high-priority load is set to $\rho_1 = 0.3$, and the low-priority load is set to $\rho_2 = 0.6$; considering TSN's QoS guarantee requirements for high-priority services, the high-priority weight $w_1 = 5$, and the low-priority weight $w_2 = 1$, the shared buffer space is uniformly set to $B = 10$ to ensure the comparison of various mechanisms under the same resource constraints. Among them, the blocking probability and delay data of CS-WFQ, CS-SP, ST, and PQTC mechanisms are obtained through simulation; the SPQM mechanism adopts the blocking probability of each priority and the overall blocking probability under the optimal priority threshold value from Ref. [53].

The comparison results of the blocking probability of the five mechanisms are shown in Figure 16. It can be seen from the figure that the two CS mechanisms (CS-WFQ and CS-SP) have no differentiated buffer control, so the blocking probabilities of high-priority, low-priority, and overall traffic are equal and relatively high, making it difficult to meet TSN's

demand for differentiated guarantee of high-priority services. Although the ST mechanism reduces the high-priority blocking probability to an extremely low level through static allocation, the low-priority blocking probability is the highest, resulting in the highest overall system blocking probability. This is because the ST mechanism is unable to fully utilize the shared memory space. In the optimal state of the SPQM mechanism, the high-priority blocking probability is higher than that of the low-priority, and the overall system blocking probability is still higher than that of the PQTC mechanism. Within the optimal threshold range, the PQTC mechanism effectively balances the buffer resource allocation between high-priority and low-priority, and its overall blocking probability is significantly lower than that of the other four mechanisms. Specifically, compared with the SPQM mechanism, it decreases by 40%; compared with the two CS mechanisms, it decreases by 50%; and compared with the ST mechanism, it decreases by 60%.

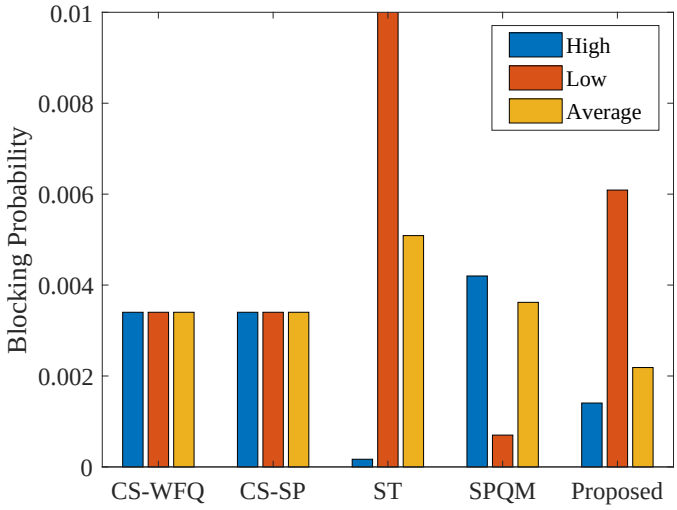


Figure 16. Comparison of blocking probabilities of different queue management mechanisms.

The comparison results of the queueing delays of different mechanisms are shown in Figure 17. The results show that CS-WFQ, CS-SP, and SPQM mechanisms exhibit relatively high levels of low-priority delay. This is the data backlog effect caused by the imbalanced traffic. Although the ST mechanism maintains a low level of average delay, it is limited by the static resource allocation logic, resulting in insufficient memory utilization and higher packet loss rate. Under optimal threshold configuration, PQTC balances high- and low-priority delays while maintaining acceptable overall latency. Although its high-priority delay is marginally higher than resource-aggressive schemes like CS-SP, this results from admitting more high-priority packets into the shared buffer. This design improves buffer utilization while ensuring lower loss rates for critical traffic. With controllable latency, low packet loss, and optimal blocking probability, PQTC collectively satisfies TSN's comprehensive QoS requirements for mixed-criticality traffic.

In summary, through dynamic threshold adjustment, the PQTC mechanism realizes the precise allocation of shared buffer resources. It demonstrates superior performance in blocking probability that far surpasses the other four mechanisms, while effectively balancing the delay performance of high-priority and low-priority queues. Consequently, it provides a buffer management solution that balances reliability and fairness for mixed-critical traffic scenarios in TSN.

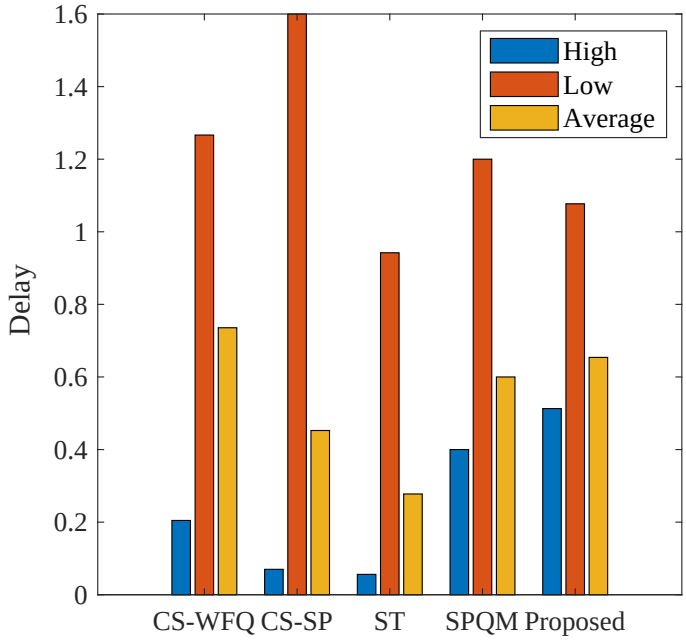


Figure 17. Comparison of delay of different queue management mechanisms.

4. Conclusions

This paper designs and analyzes a priority queue threshold control policy for mixed-critical traffic in time-sensitive networks. A two-dimensional Markov chain model is constructed to evaluate the performance of the proposed policy. The steady-state distribution of the two-dimensional Markov chain is derived and calculated by queuing theory, and the closed-form solutions for multiple parameters of each priority queue are obtained by theoretical analysis. Through adequate numerical analyses, the optimal value range of the priority threshold is obtained. In addition, by varying system parameters, it is verified that the optimal threshold value is quite stable for various traffic conditions. Finally, compared with others shared buffer queue management strategies, the proposed priority queue threshold control strategy can effectively reduce the blocking probability of mixed-critical traffic and optimize the transmission performance. The research results of this paper can provide guidance in the design of memory management modules of TSN switches.

Author Contributions: Conceptualization, L.Z. and Q.M.; methodology, L.Z.; software, W.W. and Y.F.; validation, L.Z. and Y.F.; formal analysis, Y.F. and Q.M.; investigation, W.W.; resources, L.Z.; data curation, W.W.; writing—original draft preparation, L.Z. and Y.F.; writing—review and editing, L.Z. and Y.F.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62102314 and the Natural Science Basic Research Program of Shaanxi Province under Grants 2021JQ-708 and 2022JQ-635.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hegr, T.; Voznak, M.; Kozak, M.; Bohac, L. Measurement of switching latency in high data rate ethernet networks. *Elektron. Elektrotechnika* **2015**, *21*, 73–78. [CrossRef]
2. Trowbridge, S.J. Ethernet and OTN-400G and beyond. In Proceedings of the 2015 Optical Fiber Communications Conference and Exhibition (OFC), Los Angeles, CA, USA, 22–26 March 2015; pp. 1–18. [CrossRef]
3. Du, J.L.; Herlich, M. Software-defined networking for real-time ethernet. In Proceedings of the International Conference on Informatics in Control, Automation and Robotics, Lisbon, Portugal, 29–31 July 2016; Volume 3, pp. 584–589. [CrossRef]
4. Finn, N. Introduction to time-sensitive networking. *IEEE Commun. Stand. Mag.* **2018**, *2*, 22–28. [CrossRef]
5. Messenger, J.L. Time-sensitive networking: An introduction. *IEEE Commun. Stand. Mag.* **2018**, *2*, 29–33. [CrossRef]
6. Bello, L.L.; Steiner, W. A perspective on IEEE time-sensitive networking for industrial communication and automation systems. *Proc. IEEE* **2019**, *107*, 1094–1120. [CrossRef]
7. Farkas, J.; Bello, L.L.; Gunther, C. Time-sensitive networking standards. *IEEE Commun. Stand. Mag.* **2018**, *2*, 20–21. [CrossRef]
8. Zanbouri, K.; Noor-A-Rahim, M.; John, J.; Sreenan, C.J.; Poor, H.V.; Pesch, D. A comprehensive survey of wireless time-sensitive networking (tsn): Architecture, technologies, applications, and open issues. *IEEE Commun. Surv. Tutorials* **2025**, *27*, 2129–2155. [CrossRef]
9. Xue, C.; Zhang, T.; Zhou, Y.; Nixon, M.; Loveless, A.; Han, S. Real-time scheduling for 802.1 Qbv time-sensitive networking (TSN): A systematic review and experimental study. In Proceedings of the 2024 IEEE 30th Real-Time and Embedded Technology and Applications Symposium (RTAS), Hong Kong, China, 13–16 May 2024; pp. 108–121. [CrossRef]
10. Zhang, T.; Wang, G.; Xue, C.; Wang, J.; Nixon, M.; Han, S. Time-Sensitive Networking (TSN) for Industrial Automation: Current Advances and Future Directions. *ACM Comput. Surv.* **2024**, *57*, 1–38. [CrossRef]
11. Kopetz, H.; Bauer, G. The time-triggered architecture. *Proc. IEEE* **2003**, *91*, 112–126. [CrossRef]
12. Pahlevan, M.; Obermaisser, R. Genetic algorithm for scheduling time-triggered traffic in time-sensitive networks. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 4–7 September 2018; Volume 1, pp. 337–344. [CrossRef]
13. Atallah, A.A.; Hamad, G.B.; Mohamed, O.A. Routing and scheduling of time-triggered traffic in time-sensitive networks. *IEEE Trans. Ind. Inform.* **2019**, *16*, 4525–4534. [CrossRef]
14. Alderisi, G.; Patti, G.; Bello, L.L. Introducing support for scheduled traffic over IEEE audio video bridging networks. In Proceedings of the 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), Cagliari, Italy, 10–13 September 2013; pp. 1–9. [CrossRef]
15. Deng, L.; Xie, G.; Liu, H.; Han, Y.; Li, R.; Li, K. A survey of real-time ethernet modeling and design methodologies: From AVB to TSN. *ACM Comput. Surv. (CSUR)* **2022**, *55*, 1–36. [CrossRef]
16. Houtan, B.; Ashjaei, M.; Daneshtalab, M.; Sjödin, M.; Mubeen, S. Synthesising schedules to improve QoS of best-effort traffic in TSN networks. In Proceedings of the 29th International Conference on Real-Time Networks and Systems, Nantes, France, 7–9 April 2021; pp. 68–77. [CrossRef]
17. Nasrallah, A.; Thyagaturu, A.S.; Alharbi, Z.; Wang, C.; Shao, X.; Reisslein, M.; Elbakoury, H. Performance comparison of IEEE 802.1 TSN time aware shaper (TAS) and asynchronous traffic shaper (ATS). *IEEE Access* **2019**, *7*, 44165–44181. [CrossRef]
18. Hellmanns, D.; Falk, J.; Glavackij, A.; Hummen, R.; Kehrer, S.; Dürr, F. On the performance of stream-based, class-based time-aware shaping and frame preemption in TSN. In Proceedings of the 2020 IEEE International Conference on Industrial Technology (ICIT), Buenos Aires, Argentina, 26–28 February 2020; pp. 298–303. [CrossRef]
19. Reusch, N.; Zhao, L.; Craciunas, S.S.; Pop, P. Window-based schedule synthesis for industrial IEEE 802.1 Qbv TSN networks. In Proceedings of the 2020 16th IEEE International Conference on Factory Communication Systems (WFCS), Porto, Portugal, 27–29 April 2020; pp. 1–4. [CrossRef]
20. Nakayama, Y.; Yaegashi, R.; Nguyen, A.H.N.; Hara-Azumi, Y. Real-time reconfiguration of time-aware shaper for ull transmission in dynamic conditions. *IEEE Access* **2021**, *9*, 115246–115255. [CrossRef]
21. Srinivasan, S.; Nelissen, G.; Bril, R.J.; Meratnia, N. Analysis of tsn time-aware shapers using schedule abstraction graphs. In Proceedings of the 36th Euromicro Conference on Real-Time Systems (ECRTS 2024), Lille, France, 9–12 July 2024. [CrossRef]
22. Craciunas, S.S.; Oliver, R.S.; Chmelík, M.; Steiner, W. Scheduling real-time communication in IEEE 802.1 Qbv time sensitive networks. In Proceedings of the 24th International Conference on Real-Time Networks and Systems, Brest, France, 19–21 October 2016; pp. 183–192. [CrossRef]
23. Wang, B.; Luo, F.; Fang, Z. Performance analysis of IEEE 802.1 Qch for automotive networks: Compared with IEEE 802.1 Qbv. In Proceedings of the 2021 IEEE 4th International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China, 13–15 August 2021; pp. 355–359. [CrossRef]
24. Ansah, F.; Abid, M.A.; de Meer, H. Schedulability analysis and GCL computation for time-sensitive networks. In Proceedings of the 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, 22–25 July 2019; Volume 1, pp. 926–932. [CrossRef]

25. Zhao, L.; Pop, P.; Gong, Z.; Fang, B. Improving latency analysis for flexible window-based GCL scheduling in TSN networks by integration of consecutive nodes offsets. *IEEE Internet Things J.* **2020**, *8*, 5574–5584. [CrossRef]
26. Yoshimura, A.; Ito, Y. A Study on Determination of an Appropriate GCL of Time-Aware Shaper in Ethernet-Based Industrial Networks. In Proceedings of the 2024 Fifteenth International Conference on Ubiquitous and Future Networks (ICUFN), Budapest, Hungary, 2–5 July 2024; pp. 355–359. [CrossRef]
27. Shalghum, K.M.; Noordin, N.K.; Sali, A.; Hashim, F. Critical offset optimizations for overlapping-based time-triggered windows in time-sensitive network. *IEEE Access* **2021**, *9*, 130484–130501. [CrossRef]
28. Nie, H.; Li, S.; Liu, Y. An enhanced routing and scheduling mechanism for time-triggered traffic with large period differences in time-sensitive networking. *Appl. Sci.* **2022**, *12*, 4448. [CrossRef]
29. Atallah, A.A.; Bany Hamad, G.; Ait Mohamed, O. Multipath routing of mixed-critical traffic in time sensitive networks. In *Advances and Trends in Artificial Intelligence. From Theory to Practice, Proceedings of the 32nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2019, Graz, Austria, 9–11 July 2019*; Proceedings 32; Springer: Cham, Switzerland, 2019; pp. 504–515. [CrossRef]
30. Akram, B.O.; Noordin, N.K.; Hashim, F.; Rasid, M.A.F.; Salman, M.I.; Abdulghani, A.M. Enhancing reliability of time-triggered traffic in joint scheduling and routing optimization within time-sensitive networks. *IEEE Access* **2024**, *12*, 78379–78396. [CrossRef]
31. Akram, B.O.; Noordin, N.K.; Hashim, F.; Rasid, M.F.A.; Salman, M.I.; Abdulghani, A.M. Joint scheduling and routing optimization for deterministic hybrid traffic in time-sensitive networks using constraint programming. *IEEE Access* **2023**, *11*, 142764–142779. [CrossRef]
32. Li, Z.; Wan, H.; Deng, Y.; Zhao, X.; Gao, Y.; Song, X.; Gu, M. Time-triggered switch-memory-switch architecture for time-sensitive networking switches. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2018**, *39*, 185–198. [CrossRef]
33. Wu, W.; Zhang, T.; Li, Z.; Feng, X.; Zhang, L.; Ren, F. Dynamic Per-Flow Queues in Shared Buffer TSN Switches. *ACM Trans. Des. Autom. Electron. Syst.* **2025**, *30*, 1–21. [CrossRef]
34. Yun, Q.; Xu, Q.; Zhang, Y.; Chen, Y.; Sun, Y.; Chen, C. Flexible switching architecture with virtual-queue for time-sensitive networking switches. In Proceedings of the IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society, Toronto, ON, Canada, 13–16 October 2021; pp. 1–6. [CrossRef]
35. Heilmann, F.; Fohler, G. Size-based queuing: An approach to improve bandwidth utilization in TSN networks. *ACM SIGBED Rev.* **2019**, *16*, 9–14. [CrossRef]
36. Xue, J.; Shou, G.; Liu, Y.; Hu, Y.; Guo, Z. Time-aware traffic scheduling with virtual queues in time-sensitive networking. In Proceedings of the 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), Bordeaux, France, 17–21 May 2021; pp. 604–607. ISBN 978-3-903176-32-4.
37. Choudhury, A.K.; Hahne, E.L. Dynamic queue length thresholds for shared-memory packet switches. *IEEE/ACM Trans. Netw.* **1998**, *6*, 130–140. [CrossRef]
38. Yang, J.P.; Liang, M.C.; Chu, Y.S. Threshold-based filtering buffer management scheme in a shared buffer packet switch. *J. Commun. Netw.* **2003**, *5*, 82–89. [CrossRef]
39. Ascia, G.; Catania, V.; Panno, D. An evolutionary management scheme in high-performance packet switches. *IEEE/ACM Trans. Netw.* **2005**, *13*, 262–275. [CrossRef]
40. Wang, Y.; Zhan, Y.C.; Yu, S.H. An Effective-Traffic Based Dual-Threshold Queue Control Scheme in Shared Memory Switch. In Proceedings of the 2007 International Conference on Wireless Communications, Networking and Mobile Computing, Shanghai, China, 21–25 September 2007; pp. 1933–1936. [CrossRef]
41. Shan, D.; Jiang, W.; Ren, F. Absorbing micro-burst traffic by enhancing dynamic threshold policy of data center switches. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 118–126. [CrossRef]
42. Shan, D.; Jiang, W.; Ren, F. Analyzing and enhancing dynamic threshold policy of data center switches. *IEEE Trans. Parallel Distrib. Syst.* **2017**, *28*, 2454–2470. [CrossRef]
43. Apostolaki, M.; Vanbever, L.; Ghobadi, M. Fab: Toward flow-aware buffer sharing on programmable switches. In Proceedings of the 2019 Workshop on Buffer Sizing, Palo Alto, CA, USA, 2–3 December 2019; pp. 1–6. [CrossRef]
44. Huang, S.; Wang, M.; Cui, Y. Traffic-aware buffer management in shared memory switches. *IEEE/ACM Trans. Netw.* **2022**, *30*, 2559–2573. [CrossRef]
45. Wang, M.; Cui, Y.; Wang, X.; Xiao, S.; Jiang, J. Machine learning for networking: Workflow, advances and opportunities. *IEEE Netw.* **2017**, *32*, 92–99. [CrossRef]
46. Wang, M.; Huang, S.; Cui, Y.; Wang, W.; Liu, Z. Learning buffer management policies for shared memory switches. In Proceedings of the IEEE INFOCOM 2022-IEEE Conference on Computer Communications, London, UK, 2–5 May 2022; pp. 730–739. [CrossRef]
47. Gao, Y.; Pan, W.; Zheng, L. Improved analytical model for performance evaluation of crosspoint-queued switch under uniform traffic. *IET Netw.* **2017**, *6*, 81–86. [CrossRef]

48. Hajlaoui, N.; Jabri, I.; Ben Jemaa, M. An accurate two dimensional Markov chain model for IEEE 802.11 n DCF. *Wirel. Netw.* **2018**, *24*, 1019–1031. [CrossRef]
49. Karakus, M.; Durresi, A. Quality of service (QoS) in software defined networking (SDN): A survey. *J. Netw. Comput. Appl.* **2017**, *80*, 200–218. [CrossRef]
50. Keshari, S.K.; Kansal, V.; Kumar, S. A systematic review of quality of services (QoS) in software defined networking (SDN). *Wirel. Pers. Commun.* **2021**, *116*, 2593–2614. [CrossRef]
51. Giambene, G. Markov Chains and Queuing Theory. In *Queuing Theory and Telecommunications: Networks and Applications*; Springer: Cham, Switzerland, 2021; pp. 235–282. [CrossRef]
52. Little, J.D.; Graves, S.C. Little's law. In *Building Intuition: Insights from Basic Operations Management Models and Principles*; Springer: Boston, MA, USA, 2008; pp. 81–100. [CrossRef]
53. Zheng, L.; Pan, W.; Li, Y.; Gao, Y. Performance analysis and hardware implementation of a nearly optimal buffer management scheme for high-performance shared-memory switches. *Int. J. Commun. Syst.* **2020**, *33*, e4365. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.