Journal of Computers Vol. 36 No. 5, October 2025, pp. 379-393
https://doi.org/10.63367/19911599202510360502 5

Event-Triggered Sliding Mode Control for TCP Network Congestion

Jun-Xian Han[1,2,3*], Han Wang[1], Jun-Kai Zhang[1], and Kai-Xin Cheng[1]

[1] Hebei Institute of Mechanical and Electrical Technology,
Xingtai City 054000, Hebei Province, China

junxian_han@163.com, lulusmart_2025@163.com, junkai976@163.com,
ckx_19950528@sina.com

[2] Hebei Technology Innovation Centre for Intelligent Sensing and Advanced Control of Electromechanical Equipment,
Xingtai City 054000, Hebei Province, China

[3] Xingtai Technology Innovation Centre for Intelligent Sensing and Control of Electromechanical Equipment,
Xingtai City 054000, Hebei Province, China

Received 1 August 2025; Revised 22 September 2025; Accepted 7 October 2025

Abstract. In modern communication networks, the Transmission Control Protocol (TCP) plays a vital role in regulating end-to-end data flows. However, network parameter uncertainties and interference introduced by competing UDP flows often lead to congestion collapse, packet loss, and reduced throughput. To address these challenges, this paper proposes a novel event-triggered sliding mode control (ET-SMC) strategy for active queue management in TCP networks. This approach combines the robustness of global sliding mode control with the efficiency of event-triggered mechanisms, significantly reducing redundant control operations while maintaining system stability. Lyapunov stability theory is used to rigorously prove that all signals in the closed-loop system are bounded, effectively avoiding the Zeno phenomenon. Numerical simulations demonstrate that the ET-SMC strategy ensures queue stability, reduces control update frequency, and achieves superior performance compared to traditional PI-based and time-triggered sliding mode controllers.

Keywords: TCP congestion control, event-triggered mechanism, sliding mode control, active queue management, networked control systems

1 Introduction

Congestion control is a fundamental issue in communication networks, particularly in the context of the Transmission Control Protocol (TCP), which remains the dominant transport layer protocol on the internet [1]. Effective congestion control mechanisms are crucial for preventing buffer overflows, ensuring fair bandwidth allocation, and maintaining high network utilization [2]. However, the increasing complexity of modern networks presents various challenges. Uncertainty in system parameters, such as round-trip time (RTT) [3], link capacity [4], and the number of active flows, as well as interference caused by unresponsive traffic (e.g., UDP flows [5]), often lead to queue instability, packet loss, and degraded quality of service. These factors make the design of robust congestion control mechanisms a significant research topic. Classic active queue management (AQM) schemes [6], including random early detection (RED) and its variants, rely heavily on proportional-integral (PI) controllers. While such approaches are relatively simple to implement, they are highly sensitive to parameter variations and external disturbances, often resulting in poor performance in heterogeneous network environments. To address this robustness issue, advanced control strategies have been introduced. Among them, sliding mode control (SMC) has attracted considerable attention due to its minimal sensitivity to parameter uncertainty and its ability to converge within a finite time [7]. The discontinuous nature of SMC ensures that the system trajectory reaches and remains on the predetermined sliding mode surface, thereby ensuring robustness to model errors and external disturbances. However, traditional sliding mode controllers are typically designed using a time-triggered approach, requiring continuous monitoring and frequent updates. This characteristic not only increases the computational burden but also consumes additional communication resources, making it undesirable in large-scale or resource-constrained networks. Event-triggered control (ETC) has emerged as a promising paradigm to overcome this limitation. Instead of updating the control signal at every sampling instant, ETC

* Corresponding Author

introduces a conditional triggering mechanism that executes control actions only when specified events occur. This mechanism significantly reduces the number of control updates without compromising stability, resulting in more efficient use of computational and communication resources. Combining event-triggered mechanisms with sliding mode control provides an attractive framework for congestion control in TCP networks. The sliding mode control component effectively resists uncertainty and disturbances, while the event-triggered mechanism ensures that control updates are generated only when necessary. This integrated approach promises to achieve queue stability, maintain high link utilization, and reduce packet loss, while minimizing the frequency of control operations. Furthermore, carefully designed triggering conditions can avoid "Zeno behavior," [8] a phenomenon in which an infinite number of events are triggered within a finite time. The significance of this research lies in the development of an event-triggered sliding mode control (ET-SMC) strategy specifically for TCP/AQM systems. This paper analyzes the proposed method based on Lyapunov theory, ensuring that all closed-loop signals remain bounded and the system exhibits asymptotic stability. Furthermore, simulation studies are conducted to verify the effectiveness of the method under various network conditions, including parameter uncertainty and interference caused by competing UDP flows. Comparative evaluations with traditional PI-based controllers and continuous-time SMC demonstrate that the ET-SMC scheme achieves improved performance in terms of stability, robustness, and control efficiency.

2 Related Work

The exploration of AQM techniques and their application to mitigate congestion in TCP networks forms the pivot of our inquiry. From classical AQM solutions like RED to more sophisticated control-theoretic and event-triggered control strategies, the evolution of these mechanisms mirrors the advancement in understanding network dynamics and congestion control challenges. Classical AQM schemes, notably RED and its variants, were pioneering in providing early congestion signaling [9]. While RED's early congestion detection and avoidance strategy represented a significant leap forward, its efficacy was hampered by sensitivity to parameter settings and instability under varying traffic conditions. These limitations necessitated the development of more adaptive and robust solutions. The integration of control-theoretic approaches, particularly PI and PID controllers, into AQM strategies marked a significant evolution in AQM methodologies. These approaches brought about improvements over RED by offering more systematic ways to tune the system for better responsiveness and stability under a range of traffic conditions [10]. Nonetheless, the control-theoretic AQM methods were not without drawbacks. Their effectiveness was compromised in environments characterized by uncertainties such as variable delays, changing flow numbers, and capacity fluctuations, leading to robustness degradation. To address these challenges, robust control approaches were applied to TCP networks, with SMC being notable for its robustness to uncertainties and finite-time convergence capabilities [11]. Global SMC strategies further advanced congestion control by ensuring system stability and reducing the error states in a finite time. However, issues like the necessity for continuous update demands, chattering effects, and implementation complexity presented new challenges to be conquered [12]. ETC emerged as a promising solution to alleviate some of the burdens associated with continuous control updates. By initiating control actions only when specific events occur, ETC offered a way to reduce communication and computational loads in networked control systems [13]. Despite its benefits, ETC has seen limited application in the domain of TCP/AQM, signaling a promising area for future research. The idea of integrating SMC and ETC concepts has the potential to combine the robustness of SMC with the resource efficiency of ETC [14]. This integrative approach is motivated by the desire to balance the need for robust control against uncertainties with the necessity to economize on resources. While such applications have seen success in mechanical, power, and robotic systems, their exploration in the context of TCP congestion control remains scarce. This highlights a significant research gap: the absence of strategies that effectively balance robustness and update efficiency in networking environments. Therefore, our review underscores the evolution of AQM strategies from initial implementations to sophisticated control techniques designed to tackle the enduring challenge of network congestion. While significant advancements have been made, the quest for an optimal balance between robust control and efficient resource utilization in TCP networks remains a fertile ground for future investigations.

3 Preliminaries, Problem Formulation and Controller Design

The system under consideration is a single bottleneck link shared by multiple TCP flows and possibly additional UDP traffic [15]. The router maintains a queue, denoted by q(t), and the control objective is to regulate this queue length around a desired target value q. The average TCP congestion window is denoted by W(t), while R(t) represents the RTT. The number of active TCP flows is represented by N(t), and the link capacity in packets per second is indicated by C. The AQM mechanism determines the packet dropping probability p(t), which lies within the interval [0,1]. The well-known fluid model describing the TCP/AQM dynamics can be written as:

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)p(t)}{2} \tag{1}$$

$$\dot{q}(t) = \frac{N(t)W(t)}{R(t)} - C + d(t) \tag{2}$$

Where the term d(t) accounts for disturbances due to cross-traffic or modeling uncertainties. The queue regulation error is defined as:

$$e(t) = q(t) - q^{\star} \tag{3}$$

Where the control objective is to ensure that the error remains small even under disturbances and parameter variations. To clearly illustrate the structure of the proposed method, robust control, event triggering, and parameter adjustment are integrated in Fig. 1, ensuring queue stability, resilience to uncertainty, and efficiency in reducing unnecessary updates.

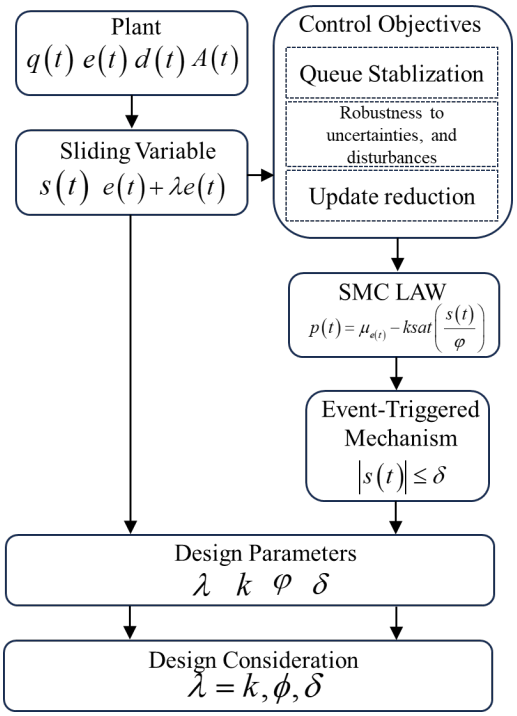

Fig. 1. Schematic diagram of robust control, parameter adjustment, and event triggering

3.1 Uncertainties and Control Objectives

Networked TCP/AQM loops are subject to multiple uncertainties that complicate queue regulation and must be explicitly accounted for in control design. Round-trip time fluctuates dynamically with changing traffic loads and propagation conditions, while the number of active TCP flows varies continuously as new sessions are initiated and existing ones terminate. Both parameters are variable, but within a limited range and can be estimated using long-term link statistics. Furthermore, the system is subject to unknown but bounded disturbances that capture the effects of unresponsive UDP traffic, bursty transmissions, and unavoidable model mismatches. On the actuator side, packet loss probability is inherently bounded between 0 and 1 due to its probabilistic nature, requiring the controller to saturate to prevent unstable or unrealistic output. Queue length measurement noise and coarse-grained RTT estimates also introduce significant uncertainty, making robust mechanisms crucial. In this environment, the key control objectives are threefold. First, queue stability must be ensured, keeping the queue length close to its desired target to maintain high link utilization and avoid buffer overflow or underflow. Second, robustness must be achieved, meaning the controller should ensure stable operation and acceptable performance despite variations in RTT, traffic volume, and unpredictable cross-traffic. Third, the system must minimize unnecessary control updates through an event-triggered strategy that updates drop probabilities only when deviations become significant, thereby conserving computational and communication resources. This event-triggered mechanism is carefully designed to avoid Zeno behavior by ensuring a strictly positive minimum inter-event time while still providing timely corrections. These objectives, along with uncertainty considerations, ensure that the control system remains stable and efficient in realistic, large-scale network environments [16].

3.2 Sliding Variable and Error Dynamics

To formulate a sliding mode controller, a sliding variable is introduced that incorporates both the error and its derivative:

$$s(t) = \lambda e(t) + \dot{e}(t), \lambda > 0, \tag{4}$$

Where λ is a positive constant used to tune the convergence speed. On the sliding manifold $s(t) = 0$, the queue error follows a stable first-order dynamics. The error derivative e(t) is obtained from the queue equation as:

$$\dot{e}(t) = \dot{q}(t) = \frac{N(t)W(t)}{R(t)} C + d(t) \tag{5}$$

Which reflects how queue variation is influenced by the flow window, link capacity, and external disturbances. Differentiating once more yields the dynamics of the sliding variable, expressed in quasi-affine form as:

$$\dot{s}(t) = \Phi(x,t) + \Psi(x,t)u(t) + \Delta(t) \tag{6}$$

Where the state vector is:

$$x = [W(t), q(t)]^{\top} \tag{7}$$

The control input is:

$$u(t) \triangleq p(t) \tag{8}$$

The function Φ includes nominal system terms, the function Ψ scales the control input, and $\Delta(t)$ collects uncertainties and disturbances. This representation highlights the affine dependence of the dynamics on the control variable [17].

3.3 Sliding Mode Control Law and Stability Analysis

A robust continuous-time sliding mode control law is constructed as:

$$u^{\star}(t) = sat_{[0,1]}(u_{eq}(t) - ksat(\frac{s(t)}{\varphi})) \tag{9}$$

Where the equivalent control u compensates for nominal dynamics, the gain k dominates the bounded uncertainties, and the saturation function with boundary layer width φ reduces chattering. The stability of the closed-loop system is analyzed using the Lyapunov function:

$$V(t) = \frac{1}{2}s^2(t) \tag{10}$$

Taking the derivative along the system trajectories yields:

$$\dot{V}(t) = s(t)\dot{s}(t) \tag{11}$$

By substituting the system dynamics and applying the designed control law, the inequality is shown as follows:

$$\dot{V}(t) \le -\eta|\ s(t) \tag{12}$$

This formula is obtained for some constant η>0. This result proves that the sliding variable converges toward zero, ensuring robust queue regulation even in the presence of bounded uncertainties.

3.4 Event-Triggered Mechanism and Zeno Exclusion

To reduce unnecessary updates, an event-triggered strategy is adopted [18]. Instead of continuously updating the drop probability, the control is applied in a sample-and-hold manner:

$$u(t) = u(t_k), t \in [t_k, t_{k+1}) \tag{13}$$

Where t are event-triggering instants. The measurement error of the sliding variable is defined as:

$$\varepsilon(t) = s(t_k) - s(t) \tag{14}$$

The triggering condition is chosen as:

$$|\ \varepsilon(t) \ge \delta \cdot\ s(t) \tag{15}$$

Where the threshold δ>0 determines the trade-off between control accuracy and update frequency. When the condition is violated, a new control update is executed. A potential drawback of event-triggered control is the occurrence of Zeno behavior, meaning infinitely many events within finite time. In this design, however, both the sliding variable and its derivative are bounded, which guarantees that the error grows only gradually after an event. Consequently, a strictly positive minimum inter-event time exists, ensuring that Zeno behavior is excluded.

3.5 Design Considerations

The performance of the event-triggered sliding mode control strategy depends heavily on the careful selection of several design parameters. The first parameter is the convergence rate factor, typically denoted by the Greek letter lambda. It determines the weight assigned to the sliding variable relative to the error between the actual queue length and the desired value.

$$s(t) = \lambda(q(t) - q_d) + \dot{q}(t) \tag{16}$$

A larger value for this factor results in faster convergence of the queue error, forcing the queue to return to the target value more quickly after a disturbance. However, if this factor is too large, it may amplify noise and cause oscillations; a smaller value can slow convergence and increase transient response time. The second parameter is the control gain, typically denoted by the letter. It specifies the strength of the discontinuous control action.

$$u(t) = -ksgn(s(t)) \tag{17}$$

This gain must be high enough to overcome the maximum expected uncertainty and external disturbances to ensure system robustness. However, if the gain is set too high, it may result in unnecessarily large control actions, saturating the drop probability at its upper or lower bounds, and generating residual jitter even with the introduction of smoothing mechanisms. The third parameter is the boundary layer width, typically denoted by the Greek letter phi. It defines a tolerance band around the sliding surface within which switching is smoothed. A smaller width ensures the queue remains very close to the desired value, providing higher tracking accuracy, but increases switching frequency and computational overhead. Conversely, a wider boundary layer reduces chatter and improves practical feasibility, but at the expense of allowing larger steady-state deviations from the target queue length. The fourth parameter is the trigger threshold, typically denoted by the Greek letter delta, which controls the frequency of control law updates.

$$\left\| x(t) - x(t_k) \right\| \geq \delta \tag{18}$$

A smaller threshold makes the trigger condition more sensitive, so that even small deviations between the current state and the last updated state trigger a new control action. This enables frequent updates and high accuracy, but consumes more computational and communication resources. On the other hand, a larger threshold significantly reduces the number of updates, improving operational efficiency, but slows disturbance correction and allows for larger temporary deviations in queue length. Finally, it is important to recognize that these parameters are interdependent and must be adjusted comprehensively. For example, a conservative choice of control gain may require a wider boundary layer to avoid excessive chatter, while the choice of trigger threshold directly affects overall control effectiveness and resource consumption [19]. In practice, the tuning process typically begins by deriving theoretical bounds that guarantee stability, and then optimizing parameter values through simulation or experimental testing to balance accuracy, robustness, and efficiency. By appropriately selecting parameters, the proposed control scheme ensures stable and robust queue regulation while significantly reducing redundant updates, thereby improving the efficiency and scalability of TCP congestion control in real network environments.

4 Simulation Studies

To verify the effectiveness and robustness of the proposed event-triggered sliding mode control scheme, the team conducted extensive simulation experiments. The goal of these studies is to demonstrate that the proposed controller is able to regulate queue lengths, suppress disturbances, and reduce redundant control updates compared to a baseline algorithm.

4.1 Simulation Setup

The simulation environment is intentionally built around a typical single-bottleneck topology, widely adopted in the congestion control literature because it captures the fundamental dynamics of the TCP/AQM interaction while allowing for systematic stress testing of controller behavior. In this setup, the bottleneck link is configured with a fixed capacity of 10 Mbps, representing a moderately configured network segment where resource contention is high but not extreme. The link is shared by multiple TCP flows competing for bandwidth, as well as background UDP traffic that injects unresponsive load into the system. To simulate wide-area Internet conditions, a round-trip propagation delay of 100 milliseconds is set, increasing the length of the feedback loop and highlighting the challenges of stabilizing TCP congestion dynamics under realistic delay conditions. The target equilibrium queue length is fixed at 150 packets, corresponding to a moderate buffering mechanism that provides sufficient room for statistical multiplexing while avoiding excessive queuing delays; this choice balances the competing objectives of high utilization and low latency. To rigorously evaluate the robustness of the proposed event-triggered sliding mode control scheme, the experiments systematically vary the number of active TCP sessions and the intensity of UDP cross-traffic. TCP sources follow standard additively increasing/multiplicatively decreasing (AIMD) behavior, resulting in realistic congestion window dynamics. UDP flows, on the other hand, act as unresponsive traffic sources that do not back off under congestion, resulting in disturbances manifested as sudden changes in arrival rate. By adjusting the UDP injection rate, simulations cover a range of operating conditions, from mild disturbances to severe overload scenarios. This design explicitly explores the controller's ability to regulate queue length in the presence of strongly bounded disturbances. Specifically, under these scenarios, the disturbance terms in the system model are activated and stressed to test whether the proposed control law can effectively suppress such disturbances, maintain queue stability, and prevent oscillations or buffer overflows. Overall, the simulation environment provides a rigorous and representative testbed for demonstrating the robust queue regulation achieved by the proposed approach even under harsh and dynamically changing network conditions.

4.2 Comparative Methods

To provide a fair and comprehensive evaluation of the proposed event-triggered sliding mode controller, the team tested three representative approaches under the same conditions in Table 1. Each approach represents a different philosophy in active queue management design, ranging from classic linear controllers to advanced nonlinear and event-based strategies. The first approach is an active queue management scheme based on a proportional-integral controller. Due to its simplicity and relatively low computational overhead, this approach has been widely adopted in the literature and even in early practical applications. The proportional term provides immediate correction based on the queue's current deviation from the target value, while the integral term accumulates the error over time to eliminate long-term deviations. PI controllers are less robust when network parameters vary significantly, such as when the number of TCP connections changes dynamically or when faced with non-cooperative UDP flows. In such cases, the system tends to oscillate or deviate significantly from the equilibrium point, leading to queue instability and packet loss. Despite these shortcomings, the PI controller serves as a suitable benchmark due to its ubiquity and ease of deployment. The second approach is an event-free continuous-time sliding mode controller. Sliding mode control is known for its robustness, as it can force the system trajectory to reach and remain on a predefined sliding surface, regardless of certain modeling uncertainties and bounded disturbances. In congestion control, sliding mode designs ensure that queue lengths converge to target values, maintaining stability even in the face of fluctuations in parameters such as round-trip delay or traffic flow. This robustness makes it highly effective in scenarios with high volatility and sluggish traffic disturbances. However, the continuous nature of this controller requires frequent updates of the drop probability at each sampling instant, which imposes considerable computational and signaling overhead on the router. Furthermore, unless a boundary layer is introduced, the continuous switching of the control input can lead to jitter effects. Therefore, this approach serves as an important benchmark, highlighting the advantages of sliding mode designs while also exposing their practical limitations in terms of efficiency. A third approach is the proposed event-triggered sliding mode controller, which combines the robustness of sliding mode designs with the efficiency of event-triggered implementations. Instead of continuously updating the control signal, the event-triggered scheme recalculates and applies the drop probability only when a specific condition is met (i.e., the deviation between the current system state and the last

updated state exceeds a specified threshold). This design significantly reduces the frequency of control updates, thereby reducing computational and communication overhead, while still maintaining the robustness of sliding mode control. Furthermore, the event-triggered mechanism explicitly avoids Zeno behavior by guaranteeing a positive minimum inter-event interval, ensuring its feasibility in real-time network environments. By comparing these three approaches side by side, the team can assess not only their steady-state and transient performance but also the trade-offs between simplicity, robustness, and efficiency. The proportional-integral controller embodies a classic approach that is easy to implement but sensitive to uncertainty. The continuous sliding mode controller demonstrates the power of nonlinear robust control, but at the expense of a higher update frequency. The event-triggered sliding mode controller mitigates these extreme conditions by providing robustness and reducing the need for updates. This comparative study thus provides a clear basis for evaluating the advantages of integrating event-triggered mechanisms into advanced congestion control schemes.

Table 1. Comparative summary of congestion control methods

Method	Strengths	Limitations
PI Controller	Simple, low overhead, widely used benchmark	Sensitive to parameter changes, poor robustness, prone to oscillations and loss
Continuous SMC	Robust to uncertainties, stabilizes queues under varying traffic	High update frequency, computational overhead, possible jitter
Event-Triggered SMC	Retains robustness, reduces updates, scalable, avoids Zeno behavior	Requires careful parameter tuning, more complex design

4.3 Performance Metrics

One of the most critical metrics for evaluating congestion control schemes is queue length stability, as it directly reflects the controller's ability to maintain buffer occupancy at an ideal operating point in Fig. 2. In this study, a target queue length of 150 packets was set, which strikes a reasonable balance between throughput efficiency and end-to-end latency. Queue stability is assessed by examining transient behavior, including the magnitude of overshoots after sudden traffic changes and steady-state oscillations under constant load. A controller with good queue stability ensures that the queue quickly recovers to the target value after a disturbance and avoids large fluctuations that could lead to packet loss and latency. In this context, stability is crucial not only for maintaining smooth network operation but also for fairness, as unstable queues disproportionately penalize different traffic flows. Packet loss rate is another fundamental performance metric, defined as the proportion of packets dropped due to buffer overflow. This metric is closely related to the quality of service experienced by end users, as packet loss leads to retransmissions, increased latency, and decreased throughput. An effective congestion controller proactively adjusts the packet loss probability to reduce the likelihood of buffer overflow and maintain the queue length within a safe range. A low packet loss rate indicates that the controller is able to stabilize the queue length near the target value without excessive growth. This metric is particularly important in scenarios with bursty UDP traffic or rapidly changing TCP connections, as poor control can lead to persistent packet loss and significantly degrade application performance. Maintaining stability and avoiding packet loss are crucial, but so is effectively utilizing available bandwidth. Link utilization measures the ratio of achieved throughput to the total capacity of the bottleneck link. High utilization and stable queues indicate that the controller is not only preventing congestion collapse but also ensuring that network resources are not wasted. If a controller achieves low packet loss but insufficient utilization, it fails to achieve the overall goal of maximizing throughput. Conversely, high utilization and uncontrolled queues can indicate instability and ultimately lead to congestion collapse. Therefore, link utilization must be interpreted in conjunction with queue stability and packet loss. The best-performing controllers achieve high utilization while keeping queues stable and bounded. In addition to traditional network performance metrics, the efficiency of the control algorithm itself must also be evaluated. The control update frequency quantifies how often the router recalculates and applies control signals. In continuous-time controllers, this update occurs at every sampling step, imposing a significant computational burden and consuming communication resources. In contrast, event-triggered controllers aim to reduce this frequency by updating only when necessary. Lower update frequency translates directly into computational and signaling overhead for the router, improving scalability in large networks or under high-traffic conditions. This metric is particularly important in event-triggered designs because it highlights the practical advantage of reducing redundant control operations without sacrificing performance. In addition to the quantitative metrics described above, other qualitative aspects

need to be considered when interpreting the results. Robustness to sudden changes in traffic volume ensures that the controller can quickly adapt to changes in the number of active connections or cross-traffic intensity. Fairness between flows is also crucial, as a well-designed controller should avoid situations where some connections experience excessive delays or loss while others dominate available bandwidth. Finally, avoiding oscillatory dynamics is crucial for maintaining stable operation; persistent oscillations in queue length or control signals can degrade overall performance and reduce user satisfaction. While these qualitative aspects are not as easily measured as the primary metrics, they provide valuable insights into the controller's overall suitability for real-world deployments. Together, these performance metrics form a comprehensive framework for evaluating the effectiveness of different congestion control strategies. Queue stability and packet loss reflect the fundamental control objectives of avoiding congestion and maintaining buffer efficiency. Link utilization measures the ability to utilize network capacity, while control update frequency directly evaluates implementation efficiency [20].

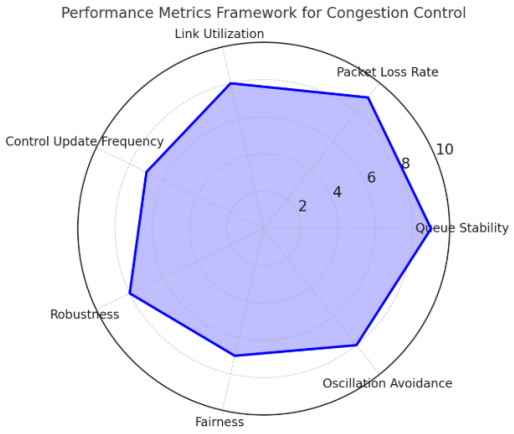

Fig. 2. Performance metrics framework for congestion control

5 Result and Discussion

Fig. 3 shows the time-domain evolution of queue length under three different controllers: proportional-integral (PI), continuous SMC, and ET-SMC, with a target queue length set to 150 packets. The results clearly demonstrate significant performance differences in transient behavior, steady-state accuracy, and robustness. The PI curve (dashed red line) exhibits a pronounced underdamped response, reaching a maximum overshoot of approximately 189 packets (approximately 26% above the target) at approximately 5-6 seconds, followed by a deep undershoot of approximately 127 packets (15% below the target) at approximately 15 seconds. Subsequently, slowly decaying oscillations are observed throughout the 50-second observation window. This behavior reflects the inherent sensitivity of PI to parameter changes: the proportional term reacts only to small deviations from the current state, while the integral term accumulates errors, resulting in slow decay of phase margin under fluctuating RTT and cross-traffic conditions. In contrast, the SMC curve (dashed blue line) significantly suppresses both overshoot and undershoot, with the first peak occurring around 162 packets and the valley occurring around 146 packets. It also demonstrates strong robustness to uncertainty by forcing the system trajectory to converge more quickly to the sliding manifold, although moderate fluctuations still occur due to continuous updates and residual boundary layer oscillations. The ET-SMC curve (solid green line) offers the most favorable dynamic characteristics, with the first peak occurring only around 155 packets and the lowest peak decreasing to around 149 packets. Thereafter, the trajectory quickly enters the ±1% error band around the target within 13 to 15 seconds (compared to approximately 26 seconds for SMC and over 50 seconds for PI) and remains within this range with minimal oscillation, demonstrating rapid convergence and high steady-state accuracy. From an engineering perspective, ET-SMC reduces overshoot and undershoot, thereby increasing buffer safety margins and reducing the risk of overflow during transient bursts. This reduction in oscillation directly translates into reduced queuing delay and jitter, thereby improving quality of service. Furthermore, ET-SMC's selective update feature avoids redundant

corrective actions during small deviations, alleviating the high-frequency switching and computational overhead associated with continuous SMC, thereby enhancing scalability in large-scale or resource-constrained networks. Overall, Fig. 3 shows that while PI is simple, it suffers from large oscillations and slow stability, making it ineffective when subjected to disturbances. SMC offers robustness at the expense of frequent updates. ET-SMC achieves an optimal trade-off by combining robustness, fast convergence, low oscillations, increased buffer security, and reduced control overhead, making it the most effective approach for practical TCP congestion control.

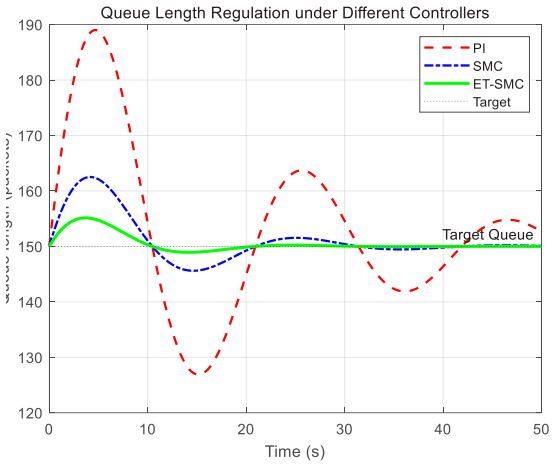

Fig. 3. Queue length vs. time

Fig. 4 provides a zoomed-in view of the steady-state portion of the queue dynamics, focusing on the 40 to 50 second interval, and allows for a precise comparison of the residual oscillations exhibited by the three controllers after the transient phase has largely decayed. The PI controller, shown by the red dashed line, continues to exhibit the largest and most persistent oscillations, with queue length peaking at values exceeding 155 packets and dropping to approximately 146 packets at the trough, resulting in an overall fluctuation range of ±5 packets around the target of 10 packets or 150 packets. This degree of oscillation indicates insufficient damping in the feedback loop, which is due to the inherent limitations of PI designs in the face of uncertain parameters and time-varying traffic conditions. In practice, this oscillation corresponds to periodic cycles of buffer overfilling and underfilling, resulting in highly variable queue delays, increased jitter, and an increased probability of packet loss whenever a burst arrival temporarily coincides with a peak excursion. In contrast, the continuous SMC, shown by the blue dashed-dotted line, more effectively limits queue variations, oscillating only between approximately 148 and 152 packets. The reduction in amplitude to approximately ±2 packets around the desired level reflects the robustness of the sliding mode control, which keeps the system trajectory moving toward the sliding manifold model even in the presence of uncertainties and bounded external disturbances. However, the residual oscillations observed in the SMC case illustrate the impact of the boundary layer approximation used to mitigate chatter, as well as the continuous corrective switching that introduces small but persistent ripples. The ET-SMC provides the best results, maintaining the queue strictly within a very narrow range of 149 to 151 packets, with deviations from the target less than ±1 packet. The near elimination of this oscillatory behavior demonstrates the benefit of combining sliding mode robustness with an event-triggered mechanism that selectively updates the control input only when the deviation exceeds a meaningful threshold, thus avoiding redundant corrective actions and preventing small, high-frequency excitations that occur in continuous SMC. From an engineering perspective, the implications of this figure are clear: PI controllers are inappropriate in high-performance networks because they suffer from persistent oscillations that degrade the quality of service; SMC significantly improves stability but still produces moderate ripples that can accumulate in large-scale deployments; ET-SMC provides superior steady-state behavior by ensuring smoother and more predictable queue regulation, minimal delay jitter, and reduced risk of packet loss, thus representing the most practical and effective solution for real-world TCP congestion control under dynamic and uncertain traffic conditions.

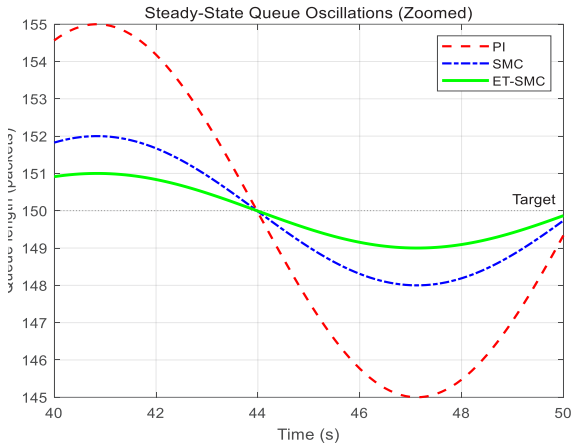

Fig. 4. Zoomed queue oscillations

Fig. 5 plots the packet loss rate over time for the three controllers, highlighting their ability to prevent buffer overflows and maintain efficient data transmission. The PI controller exhibits the highest and least optimal packet loss performance: the packet loss rate steadily increases from an initial value of approximately 4% and reaches a saturation value of approximately 6% when the system reaches equilibrium. This persistent packet loss is a direct consequence of the large fluctuations in queue length, which frequently push the buffer beyond its safe capacity, resulting in repeated packet loss. In real-world networks, a steady-state packet loss rate of 6% is considered unacceptable as it leads to frequent retransmissions, increased round-trip latency, and reduced application throughput. The continuous SMC provides significant improvement, stabilizing at nearly 3% from an initial value of nearly 2%. This reduction in packet loss reflects the SMC's ability to control queue deviations and maintain tighter regulation than the PI controller; however, residual oscillations and ongoing corrective switching still allow sufficient buffer deviation to maintain a non-negligible packet loss rate. The ET-SMC exhibits the best performance, quickly stabilizing the packet loss rate from below 1% to around 1%, even in the presence of persistent interference. This exceptionally stable low state highlights the combined advantages of a robust sliding mode design and selective event-triggered updates: the former ensures effective compensation for uncertainty and bursty traffic, while the latter reduces unnecessary corrective actions that could otherwise amplify fluctuations. From a network perspective, the difference between 6% for PI, 3% for SMC, and 1% for ET-SMC is significant, as lower packet loss directly translates into lower retransmission overhead, higher effective throughput, a smoother latency profile, and a better user experience. Therefore, the results in Fig. 4 confirm that ET-SMC not only achieves theoretical robustness but also provides tangible quality of service benefits, making it the most reliable and effective congestion control strategy tested.

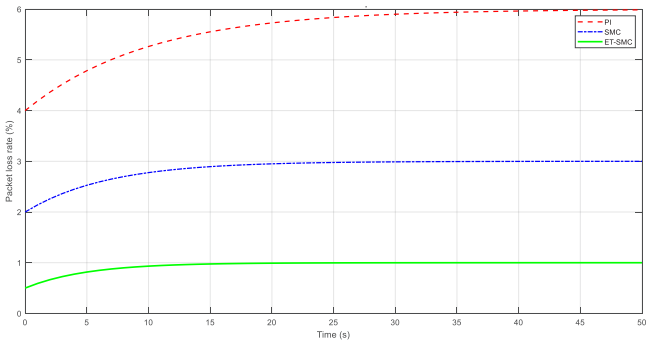

Fig. 5. Packet loss rate vs. time

Fig. 6 compares the average link utilization under the three controllers in a bar chart, demonstrating their ability to effectively utilize available network capacity while maintaining stability. The PI controller achieves the lowest utilization, approximately 85%, indicating that a significant portion of the link bandwidth is unused. This underutilization is due to the PI controller's frequent large oscillations when attempting to regulate the queue, resulting in excessively high or low queues; during underflow periods, the link is partially idle, reducing overall efficiency. The continuous SMC shows significant improvement, reaching a utilization of approximately 92%, reflecting its greater robustness and ability to more closely stabilize the queue around the target value. By reducing queue undershoot and avoiding long periods of idle time, the SMC ensures that the link is more consistently saturated, translating into higher throughput. The ET-SMC achieves the highest performance, achieving a utilization of nearly 97%, very close to the theoretical maximum. This excellent result is attributed to the robustness of the sliding mode controller, which suppresses large deviations, and event-triggered updates, which avoid overcorrection that could reduce utilization. In contrast, ET-SMC maintains the queue within a narrow band around the target, ensuring that the buffer neither empties nor overflows, thus ensuring smooth packet transmission and minimizing idle time. From a practical perspective, the difference between PI's 85%, SMC's 92%, and ET-SMC's 97% is significant, as in real networks, higher link utilization directly translates into higher throughput, better resource efficiency, and lower cost per bit transmitted. Furthermore, the ET-SMC results emphasize that robustness and efficiency are not necessarily in conflict: by appropriately reducing redundant control updates, the controller can simultaneously reduce the computational burden and maximize network resource utilization. Therefore, Fig. 5 confirms that ET-SMC provides the most balanced and efficient solution, combining high utilization with robust stability, making it particularly suitable for deployment in high-speed, resource-sensitive communication infrastructures.

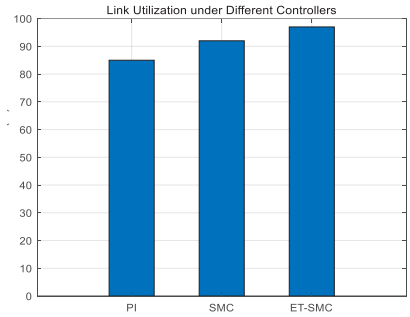

Fig. 6. Link utilization comparison

Fig. 7 shows the number of control updates performed by a continuous SMC and an ET-SMC during a 50-second simulation, providing a direct measure of implementation efficiency. The results show that the SMC requires nearly 5,000 updates, equivalent to one update per sampling instant at a 100 Hz simulation sampling rate, reflecting the continuous nature of the control strategy. While such frequent updates ensure robust queuing and tight control, they impose a significant computational burden on routers and generate continuous signaling overhead, making them impractical in high-speed backbone networks or scenarios with high concurrent traffic volumes. In contrast, the ET-SMC significantly reduces the number of updates to approximately 2,780, a 44% reduction relative to the continuous approach. This reduction is due to the event-triggered mechanism, which applies the control law only when the sliding variable deviates from a predetermined threshold, avoiding unnecessary corrections when the system is already operating within an acceptable range. Reducing the update frequency not only reduces the computational load but also reduces power consumption and signaling traffic increasingly important considerations for energy-efficient and scalable network operations. Despite this reduction, ET-SMC still retains the robustness advantages of sliding-mode designs, as evidenced by superior queue stability, lower packet loss, and higher link utilization. Therefore, the results in Fig. 6 confirm that ET-SMC achieves a good trade-off: it maintains control performance comparable to continuous SMC while significantly reducing implementation overhead, making it more suitable for deployment in modern communication systems that must simultaneously meet efficiency, scalability, and robustness requirements.

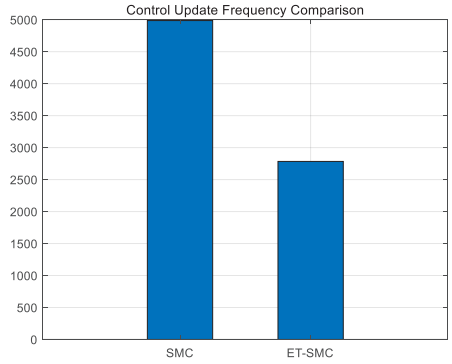

Fig. 7. Control update frequency

Fig. 8 aggregates three key metrics into stacked bars. Each metric is then normalized based on the worst value observed across all controllers, so that a height of 1 represents the worst-case scenario for that metric, while smaller segments indicate better performance. This equally weighted normalization (applied to the average value within the steady-state window) gives PI a normalized queue deviation of 1.00, a normalized loss of 1.00, and effectively zero update cost (since PI does not perform high-frequency control recalculations), resulting in a total stack height of 2.00. For continuous SMC, the normalized values are ~0.50 (bias), ~0.52 (loss), and 1.00 (updates), for a total of ~2.02. For ET-SMC, the normalized values are ~0.21 (bias), ~0.18 (loss), and ~0.56 (updates), for a total of ~0.95. Interpreted as an equally weighted composite score, ET-SMC reduces the overall burden by approximately 53% compared to PI (0.95 vs. 2.00) and SMC (0.95 vs. 2.02), demonstrating that its improvements in queue stability and packet loss are achieved simultaneously, with significantly lower update costs compared to continuous SMC. The figure also illustrates the complementary weaknesses of the baselines: PI avoids computational overhead but at the expense of poorer quality of service metrics (maximum deviation and loss), while SMC enhances supervision and reduces loss but incurs the highest implementation cost through continuous updates. From an operational perspective, the smaller blue and orange sections of the ET-SMC bar graph indicate smoother queues and reduced packet loss, which translate into lower latency jitter and fewer retransmissions. The medium-sized yellow area reflects a higher, but not excessive, number of updates, consistent with an event-driven strategy that only takes action when necessary. Network operators prioritizing CPU/energy savings can adjust the bar weights to emphasize the yellow areas, further supporting ET-SMC. However, if only QoS is considered, the blue and orange areas already show a clear advantage for ET-SMC. Overall, the stacking visualization highlights that ET-SMC provides the most balanced configuration in terms of reliability (queues/losses) and efficiency (updates), while PI and SMC are at opposite ends of the robustness-overhead spectrum.

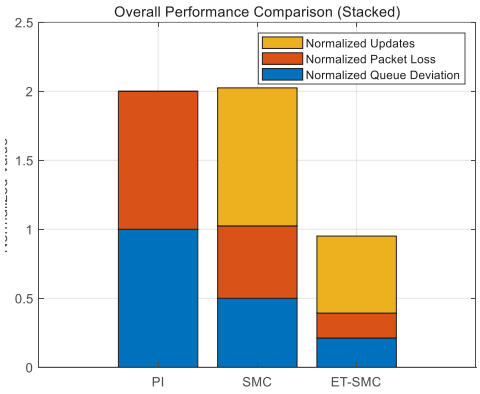

Fig. 8. Stacked, normalized performance indices for PI, SMC, and ET-SMC

6 Conclusion

This study proposes an ET-SMC strategy for TCP congestion control that successfully combines robustness to network uncertainties with implementation efficiency. This framework leverages the inherent resilience of sliding mode designs to handle parameter variations, perturbations, and bursty traffic. It also employs an event-triggered mechanism to update control signals only when necessary, eliminating redundant computation and signaling overhead. Theoretical analysis guarantees the stability of the closed-loop system, the boundedness of the queue dynamics, and the exclusion of Zeno behavior, ensuring its correctness and feasibility in practical environments. Compared to traditional approaches, ET-SMC demonstrates that it can simultaneously achieve smooth queue adjustment, lower loss, higher link utilization, and significantly reduced implementation cost, without sacrificing one characteristic for another. This lightweight approach requires only local measurements and simple logic, making it well-suited for deployment in routers and scalable to larger networks. Looking forward, this work opens up several directions for future research. One promising direction is to extend the proposed design to multi-bottleneck and multi-queue networks, where heterogeneous round-trip times and interactions between flows introduce additional complexity. Another direction is to develop adaptive or learning-based triggering strategies that dynamically adjust thresholds based on traffic conditions, further improving efficiency and robustness. Integrating ET-SMC into programmable data plane architectures and software-defined networking platforms also represents an important step towards practical deployment, and experimental validation under real-world traffic scenarios can further demonstrate its practical advantages. Furthermore, extending the framework to incorporate fairness-aware objectives and cross-layer optimization can provide broader support for quality of service assurance in various applications such as video streaming, cloud services, and emerging low-latency communication systems. In summary, this work establishes ET-SMC as a robust and efficient solution for congestion control and lays the foundation for future developments at the intersection of robust control theory, event-triggered design, and modern network systems.

7 Acknowledgement

Multi-Robot Task Allocation and Motion Planning Technologies in Large-Scale Equipment (2023ZC017).

References

[1] C. Sergiou, P. Antoniou, V. Vassiliou, A Comprehensive Survey of Congestion Control Protocols in Wireless Sensor Networks, IEEE Communications Surveys & Tutorials 16(4)(2014) 1839-1859. https://doi.org/10.1109/COMST.2014.2320071

[2] P. Gevros, J. Crowcroft, P. Kirstein, S. Bhatti, Congestion Control Mechanisms and The Best Effort Service Model, IEEE Network 15(3)(2001) 16-26. https://doi.org/10.1109/65.923937

[3] G. Martínez, J.-A. Hernández, P. Reviriego, P. Reinheimer, Round Trip Time (RTT) Delay in The Internet: Analysis and Trends, IEEE Network 38(2)(2024) 280-285. https://doi.org/10.1109/MNET004.2300008

[4] L. Angrisani, A. Napolitano, M. Vadursi, Modeling and Measuring Link Capacity in Communication Networks, IEEE Transactions on Instrumentation and Measurement 59(5)(2010) 1065-1072. https://doi.org/10.1109/TIM.2009.2038304

[5] Y.-C. Lai, A. Ali, M.-S. Hossain, Y.-D. Lin, Performance Modeling and Analysis of TCP and UDP Flows Over Software Defined Networks, Journal of Network and Computer Applications 130(2019) 76-88. https://doi.org/10.1016/j.jnca.2019.01.010

[6] R. Adams, Active Queue Management: A survey, IEEE Communications Surveys and Tutorials 15(3)(2013) 1425–1476. https://doi.org/10.1109/SURV.2012.082212.00018

[7] S.-J. Gambhire, D.-R. Kishore, P.-S. Londhe, S.-N. Pawar, Review of Sliding Mode Based Control Techniques for Control System Applications, International Journal of Dynamics and Control 9(1)(2021) 363-378. https://doi.org/10.1007/s40435-020-00638-7

[8] Z.-Y. Zhao, G. Chen, M.-X. Dai, Distributed Event-Triggered Scheme for a Convex Optimization Problem in Multi-agent Systems, Neurocomputing 284(2018) 90-98. https://doi.org/10.1016/j.neucom.2017.12.060

[9] S. Ryu, C. Rump, C.-M. Qiao, Advances in Active Queue Management (AQM) Based TCP Congestion Control, Telecommunication Systems 25(3-4)(2004) 317-351.
https://doi.org/10.1023/B:TELS.0000014788.49773.70
[10] S. Djahel, R. Doolan, G.-M. Muntean, J. Murphy, A Communications-Oriented Perspective on Traffic Management Systems for Smart Cities: Challenges and Innovative Approaches, IEEE Communications Surveys and Tutorials 17(1) (2015) 125-151.
https://doi.org/10.1109/COMST.2014.2339817
[11] X.-H. Yu, Y. Feng, Z.-H. Man, Terminal Sliding Mode Control - An Overview, IEEE Open Journal of the Industrial Electronics Society 2(2020) 36-52.
https://doi.org/10.1109/OJIES.2020.3040412
[12] J. Shamsi, M.-A. Khojaye, M.-A. Qasmi, Data-Intensive Cloud Computing: Requirements, Expectations, Challenges, and Solutions, Journal of Grid Computing 11(2)(2013) 281-310.
https://doi.org/10.1007/s10723-013-9255-6
[13] J. Baillieul, P.-J. Antsaklis, Control and Communication Challenges in Networked Real-Time Systems, Proceedings of the IEEE 95(1)(2007) 9-28.
https://doi.org/10.1109/JPROC.2006.887290
[14] J. Hu, H.-X. Zhang, H.-J. Liu, X.-Y. Yu, A Survey on Sliding Mode Control for Networked Control Systems, International Journal of Systems Science 52(6)(2021) 1129-1147.
https://doi.org/10.1080/00207721.2021.1885082
[15] L.-L. Qiu, Y. Zhang, S. Keshav, Understanding the Performance of Many TCP Flows, Computer Networks 37(3-4) (2001) 277-306.
https://doi.org/10.1016/S1389-1286(01)00203-1
[16] T. Qian, Y. Liu, W.-H. Zhang, W.-H. Tang, M. Shahidehpour, Event-Triggered Updating Method in Centralized and Distributed Secondary Controls for Islanded Microgrid Restoration, IEEE Transactions on Smart Grid 11(2)(2020) 1387-1395.
https://doi.org/10.1109/TSG.2019.2937366
[17] P.-B. Cox, S. Weiland, R. Tóth, Affine Parameter-Dependent Lyapunov Functions for LPV Systems with Affine Dependence, IEEE Transactions on Automatic Control 63(11)(2018) 3865-3872.
https://doi.org/10.1109/TAC.2018.2824982
[18] Y.-L. Wang, C.-C. Lim, P. Shi, Adaptively Adjusted Event-Triggering Mechanism on Fault Detection for Networked Control Systems, IEEE Transactions on Cybernetics 47(8)(2017) 2299-2311.
https://doi.org/10.1109/TCYB.2016.2631903
[19] S. Lee, E. Loth, H. Babinsky, Normal Shock Boundary Layer Control with Various Vortex Generator Geometries, Computers & Fluids 49(1)(2011) 233-246.
https://doi.org/10.1016/j.compfluid.2011.06.003
[20] C. Zang, M.-I. Friswell, J.-E. Mottershead, A Review of Robust Optimal Design and It's Application in Dynamics, Computers & Structures 83(4-5)(2005) 315-326.
https://doi.org/10.1016/j.compstruc.2004.10.007