AEM JOURNAL.ORG

WAPI-Integrated Congestion Control and Adaptive Transmission for Power IoT

Shan Lu[1], **Shiqi Cong**[2], **Qingrong He**[2], **Chunpeng Li**[2], **Yun Ju**[2]
[1]Electric Power Research Institute of State Grid Shanxi Electric Power Company, Taiyuan 030001, China
[2]North China Electric Power University, Beijing, China

Corresponding author: Yun Ju (e-mail: 50501482@ncepu.edu.cn).

ABSTRACT Modern power IoT systems feature massive terminal deployment and increasingly diverse service scenarios. Such conditions frequently cause network congestion, packet disorder and unstable transmission, severely undermining overall network performance. Traditional optimization methods cannot adapt to dynamic heterogeneous network environments and lack native compatibility with the widely adopted WAPI security protocol. Addressing these core defects in power IoT transmission, this study builds an integrated communication optimization framework with embedded WAPI security modules. It further proposes SymDistill-CC, a new algorithm serving as the core congestion control component of the framework. It leverages the teacher-student distillation architecture to transform deep reinforcement learning policies into readable symbolic decision trees, offering transparent decision logic, comparable transmission performance and significantly reduced computation overhead. We then put forward the ECN-FPS packet scheduling mechanism. It detects shared bottlenecks via explicit congestion signaling and use forward prediction to mitigate misordered packets, substantially easing buffer pressure on receiving terminals. Finally, we construct a security-aware dynamic retransmission mechanism for power IoT that perceives network states in real-time under the WAPI authentication framework and adaptively selects unicast or multicast retransmission modes to enhance transmission efficiency and resource utilization. Experimental results demonstrate that in simulated WAPI network environments, the proposed methods achieve a 15%–20% throughput improvement, a 42.5% reduction in out-of-order packets and a link utilization up to 88%–90%, with significant advantages also shown in retransmission efficiency and other key metrics, providing an effective solution for constructing secure, efficient, and reliable power IoT communication systems.

INDEX TERMS Power Internet of Things, WAPI, Congestion Control, Packet Scheduling, Adaptive Retransmission

I. INTRODUCTION

In recent years, the in-depth integration of wireless networks and the Internet of Things (IoT) has driven the digital transformation of key fields such as smart grids. As China's independently developed wireless security standard, Wireless Local Area Network Authentication and Privacy Infrastructure (WAPI) has been increasingly widely applied in power IoT scenarios with strict requirements for security and stability, thanks to its reliable authentication mechanism. However, the transmission environment of power IoT is complex and changeable: the network is characterized by large-scale terminal deployment (10^3–10^5 nodes in a regional smart grid), diverse business types and a typical many-to-one star communication mode; wireless links are prone to random packet loss and bandwidth fluctuations; shared bottleneck issues may occur during multipath transmission; moreover, most terminal devices have limited computing power and are sensitive to power consumption. Traditional transmission optimization solutions struggle to simultaneously meet the requirements of security, efficiency, and adaptability.

In terms of congestion control, traditional algorithms such as TCP CUBIC and TCP BBR have low computational overhead but lack flexibility when adapting to complex scenarios of WAPI networks (e.g., multi-user access and OFDM subcarrier allocation). Deep reinforcement learning (DRL)-based methods like Aurora can improve throughput, but their opaque decision-making process makes fault diagnosis difficult. Additionally, their high computational overhead prevents real-time operation on edge terminals in power IoT. Although symbolic regression (SR) technology can make decision rules more interpretable, static rules fail to cover diverse network conditions and become ineffective when facing sudden bandwidth changes or packet loss type switches.

In multipath transmission, existing scheduling strategies such as LowRTT and FPS only focus on path delay and ignore the shared bottleneck of subflows under WAPI security association. When multiple subflows compete for the same link, the blind growth of their respective congestion windows exacerbates congestion, leading to the accumulation of a large number of out-of-order packets at the receiver, exhaustion of buffers, and impairment of the real-time performance of services like power monitoring data.

The problems of retransmission mechanisms are also prominent: static unicast retransmission causes a sharp increase in network load under high packet loss rates, while static multicast retransmission results in non-packet-loss terminals receiving redundant data. More importantly, there is a strong coupling relationship among congestion control, multipath packet scheduling and retransmission mechanisms: unreasonable rate decisions of congestion control will aggravate the shared bottleneck conflict of multipath subflows, further trigger packet out-of-order and a sharp increase in retransmission times; blind retransmission will conversely increase network load and exacerbate congestion, and the out-of-order problem of packet scheduling will directly reduce retransmission efficiency. Their negative effects superimpose on each other and form a vicious circle. Existing studies optimize a single mechanism in isolation and lack an integrated collaborative design approach, making it impossible to fundamentally solve the transmission optimization problems in the WAPI scenario of power IoT. This is exactly the core necessity for constructing the integrated communication optimization framework in this paper.

To address these issues, this paper aims to optimize transmission in WAPI-based power IoT and proposes a multi-level solution. First, the SymDistill-CC congestion control framework is designed. Through a two-stage distillation process, the strategy of the DRL model is converted into transparent symbolic rules. A dynamic branching mechanism is also added to enable the algorithm to adapt to different network scenarios, balancing performance, interpretability, and lightweight design. Second, the ECN-FPS (Explicit Congestion Notification-Forward Prediction Scheduling) algorithm is proposed, which uses ECN congestion signals to detect shared bottlenecks of subflows and adjusts the packet transmission order based on forward prediction to reduce out-of-order issues at the source. Finally, a WAPI security-aware adaptive retransmission mechanism is constructed: an intelligent decision core is embedded in the Access Controller (AC) to monitor network load and terminal status in real time, flexibly selecting between unicast and multicast retransmission. This ensures the security of WAPI authentication while saving terminal resources.

To verify the effectiveness of these solutions, this paper builds a WAPI-based power IoT environment on the NS-2 and NS-3 simulation platforms, designing scenarios such as static packet loss, dynamic bandwidth, and multi-flow competition. Performance comparisons with traditional algorithms and mainstream solutions are conducted based on indicators including throughput, delay jitter, and link utilization. The research of this paper not only provides a new idea for the collaborative design of "security-efficiency-adaptability" in wireless network transmission optimization but also offers technical support for the reliable operation of WAPI-based power IoT, facilitating the stable digital upgrade of smart grids.

This paper's structure is shown below. Chapter 2 looks at past research about congestion control, multipath scheduling and retransmission methods used in power IoT networks. It also lists clear weaknesses of all existing optimization plans. Chapter 3 explains the three key methods this paper puts forward. It talks about model building, math formulas and detailed steps to run these methods. Chapter 4 tests how well our plans work in different usage cases and compares all test data. Chapter 5 wraps up all research results and names areas for further study later.

II. RELATED WORK

Wireless networks and the Internet of Things have grown fast in recent years. Many researchers and industry workers want to find better ways to make data transmission efficient, safe and stable in changing and complex network environments. Most related research works focus on three main parts. The first part is congestion control algorithms. Researchers keep improving these algorithms. They start from basic methods that judge network conditions by packet loss or delay. They also use deep reinforcement learning and symbolic technology for smarter control. Their main goal is to make the algorithm work well and easy to explain. The second part is multipath scheduling strategies. These strategies use multiple network interfaces and links to make better use of bandwidth and speed up data transmission. They still have problems in handling disordered data packets and finding shared network bottlenecks. The third part is retransmission mechanism optimization. Power IoT devices often have limited resources. For these devices, researchers struggle to keep a good balance of communication safety, network link usage and device energy saving. Current research has made good progress in these three fields, but it still lacks an all-in-one framework. This framework needs light-weight design, dynamic adaptation and reliable security functions.

A. RESEARCH STATUS OF CONGESTION CONTROL ALGORITHMS

Early congestion control algorithms mostly work with fixed rules made by humans. TCP CUBIC uses packet loss to adjust network transmission [1]. TCP Vegas watches network delay to finish control work [1]. Traditional network optimization algorithms work well in stable network environments. They do not adapt to changing network states. Random packet loss, sudden bandwidth changes and mobile network nodes all lower the actual usage efficiency of network links. Researchers use data-driven learning methods to fix these weaknesses. Aurora is a typical learning-based network framework. It uses deep neural networks to adjust congestion control rules. It can raise network throughput by

75.3 percent in complex network scenes, compared with old algorithms [2]. These learning methods still have clear shortcomings. Neural networks make network decisions with unclear logic. Workers cannot easily find and fix network faults. The calculation process of these models also takes extra time. Such extra computing cost makes these methods hard to use on edge devices with limited hardware resources [3].

Many new optimization methods target low resource use in network infrastructure. The In-Network Resource Pooling method, also known as INRPP [4], changes data flow paths with router caching. This method reduces link congestion. It improves link utilization by 50 percent. This method needs special SRAM-DRAM hardware. It costs much money for real deployment. The Multipath Transmission Control Protocol (MPTCP) is able to aggregate bandwidth from multiple paths, yet it suffers from severe performance losses in asymmetric path scenarios, revealing its inherent limitations constrained by terminal device capabilities. In recent years, symbolic regression (SR) has provided a viable way to reconcile model performance and interpretability [5]. Existing research has successfully distilled reinforcement learning policies into symbolic decision trees, which greatly improves the interpretability of network decision rules and reduces computational overhead by 23 times. However, such static symbolic rules fail to cover heterogeneous network conditions (e.g., threshold conflicts in high/low bandwidth scenarios) [6], and branch switching relies on pre-clustering division, making real-time response to dynamic environmental changes impossible.

Therefore, there is an urgent need for a lightweight, dynamically aware congestion control framework that ensures cross-scenario generalization while meeting interpretability and real-time requirements. Existing congestion control studies for power IoT rarely integrate with WAPI security protocols, and lack targeted optimization for WAPI's OFDM subcarrier allocation and multi-user access control characteristics in power IoT scenarios. The SymDistill-CC proposed in this paper is a systematic solution addressing this gap: converting black-box neural network strategies into white-box symbolic decision trees through two-stage distillation, retaining the performance advantages of DRL while providing transparent decision paths; designing a dynamic branching mechanism that uses a lightweight KNN classifier to sense network context in real time [6] and activate optimal symbolic rules; achieving end-to-end lightweight design.

B. CURRENT STATE OF PACKET SCHEDULING ALGORITHM RESEARCH

Due to the heterogeneous and dynamic characteristics of networks, different paths often have varying bandwidth and latency, causing packets to arrive out of their original transmission order. Receivers need to buffer substantial out-of-order packets, which quickly depletes limited receiver buffer space and consumes available receiving window resources. To address this problem, Yu et al. [7] implemented an Admission-In First-Out (AIFO) queue at the data link layer. The core idea of AIFO is to maintain a sliding window to track the ranking of recent packets and calculate the relative ranking of arriving packets within the window for admission control. In literature [8], Wu et al. proposed a Deep Reinforcement Learning (DRL)-based scheduling strategy called DISA, which can identify competitive and reasonable channel and time slot combinations in industrial Internet of Things. The Multipath Transmission Control Protocol (MPTCP) is an extension of the standard TCP protocol that allows simultaneous data transmission through multiple network interfaces, thereby achieving link aggregation and higher network performance. In literature [9], Jiangping et al. proposed a coupled BBR and Adaptive Redundancy and Prediction (AR&P) scheduler to improve adaptability and maintain in-order packet transmission in highly dynamic network scenarios. Hayes et al. [10] proposed a novel dynamic clustering algorithm that introduces delay characteristics to realize shared bottleneck detection, and improved the offline wavelet-filter-based SBFG method to adapt to online operation. Xie et al. [11] designed a shared bottleneck detection based decoupled congestion control strategy, named SD-DCC, which combines packet loss and delay information to identify shared bottlenecks. Nevertheless, in WAPI-enabled multipath transmission scenarios, such methods suffer from inherent drawbacks including delayed observability and poor noise robustness.

Explicit Congestion Notification (ECN) proactively sends a clear signal to data senders when network devices detect impending congestion, and many congestion control and packet scheduling algorithms currently use ECN to determine network congestion [12]. Kaiping et al. [13] proposed a fine-grained forward prediction-based dynamic packet scheduling MPTCP scheme that simultaneously considers packet loss rate and time offset, significantly improving network throughput and reducing receiver buffer occupancy. However, the above methods do not consider shared bottlenecks and cannot maximize throughput on all available paths. In WAPI-integrated power IoT multipath transmission, existing scheduling algorithms fail to combine WAPI security association management with subflow competition control, leading to severe packet out-of-order under WAPI's secure multipath transmission constraints. Therefore, this study proposes a packet scheduling method based on ECN-FPS, which proactively avoids disorder and reduces receiver buffer requirements from the source.

C. RESEARCH STATUS OF ADAPTIVE RETRANSMISSION METHODS FOR POWER IOT BASED ON WAPI

Research on retransmission mechanisms for WAPI-integrated power IoT is still in the initial stage: existing studies either simply apply WAPI security to general wireless retransmission or optimize power IoT retransmission without considering WAPI's authentication and encryption overhead. Liu et al. [14] incorporated deep learning into WAPI systems to improve channel access performance, without conducting in-depth optimization for retransmission strategies. Guo et al. [15] proposed the HELDR algorithm for hyper-edge live streaming networks, which reduces latency via fast retransmission but is limited exclusively to media streaming scenarios. NetRT proposed by Wang et al. [16] and FaSR designed by Huang et al. [17] optimize RDMA performance in data centers through in-network retransmission offloading and NIC-level processing respectively. Nevertheless, both solutions depend on high-performance hardware, which limits their applicability to resource-constrained IoT scenarios. Lin et al. [18] and Yu et al. [19] investigated adaptive retransmission schemes for federated learning and vehicle-to-everything (V2X) communication systems, yet neither work accommodates the integration demands of security authentication frameworks. Current studies present several evident limitations. First, few existing methods achieve thorough integration with mainstream security specifications such as WAPI. Second, conventional retransmission strategies lack flexibility for power network scenarios, as static multicast causes resource waste while pure unicast easily induces network congestion. Third, existing solutions cannot effectively perceive real-time network and terminal resource states or achieve dynamic resource balancing. This paper presents a security-aware dynamic retransmission mechanism customized for power IoT systems to mitigate the above issues. The proposed mechanism tightly integrates dynamic retransmission algorithms with WAPI specifications. By embedding an intelligent decision core into the access controller, the system achieves real-time awareness of network load and terminal operating states, and adaptively selects optimal unicast or multicast retransmission policies. This design improves network throughput and extends terminal service life while guaranteeing overall network security and reliability.

III. METHOD

A. OVERVIEW OF THE METHOD

This paper build a multi-layer optimization framework. It fix congestion control and transmission problems for wireless networks and power IoT systems with WAPI support. We treat network optimization as a sequence decision task for congestion control. We mix reinforcement learning and symbolic distillation. This way the system can adjust transmission speed and schedule data packets with high efficiency and clear logic. We add a branch reinforcement learning module. The module fit different network environments and lifts overall system performance. We create the ECN-FPS forward prediction scheduling algorithm for multipath transmission. The algorithm find shared network bottlenecks. It cuts conflicts between different data streams and reduces disordered packets. It lets multiple data paths send data fairly and in order. We make an adaptive retransmission plan that matches WAPI safety verification rules. This plan meets real deployment needs of power IoT. It switch single-target transmission and multi-target transmission according to real-time link conditions. It keep a good balance between stable data transfer and device power use. The three layered plans form one complete transmission optimization framework. The framework have safety, adaptive ability and high efficiency. It offers practical technical support for new wireless and power IoT business.

B. CONGESTION CONTROL ALGORITHM BASED ON WAPI AND REINFORCEMENT LEARNING

All experiments are carried out on WAPI-RL, an OpenAI Gym-based reinforcement learning testbed designed for WAPI WLAN congestion control. The platform is customized to reproduce WAPI-specific OFDM subcarrier allocation, multi-user access management and QoS classification mechanisms. We formalize WAPI congestion control as a sequential decision task. At the start of every WAPI Measurement Interval (W-MI), the agent picks transmission rates and frame scheduling policies based on observed WAPI network statistics, and then calculate a composite reward signal. The complete architecture as shown in Figure 1.

We establish a formal Markov Decision Process (MDP) model for WAPI congestion control to solidify our theoretical analysis, which is defined with the following tuple:

$$\mathbf{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma) \tag{3.1}$$

State $s_t \in \mathcal{S}$ is defined as a multi-dimensional vector used to capture network conditions at each time step t:

$$s_t = [\text{throughput}_t,\ \text{packet loss}_t,\ \text{queue length}_t,\ \text{delay}_t,\ \Delta\text{loss slope}_t,\ \text{buffer peak}_t, \ldots].$$

We further incorporate cross-W-MI statistical features to encode temporal network variations.

$$S_{\text{attr}} = [\text{SNR}_{\text{avg}}(k),\ \text{FLR}_{\text{trend}}(k),\ \text{QL}_{\max}(k)] \tag{3.2}$$

The aggregated features enable the model to track historical trends with a fixed-dimension state space. Let denote the state dimension d; it stays compact, generally under 20, so the learning task remains computationally feasible.

Action $a_t \in \mathcal{A}$ stands for the congestion control decision. We define the action space as a discrete set: $\mathcal{A} = \{\text{increase rate}, \text{decrease rate}, \text{maintain rate}\}$. Every available action adjusts the transmission rate together with the frame scheduling policy.

State transition dynamics follow the formulation $P(s_{t+1} \mid s_t, a_t)$. The transition function cannot be explicitly modeled due to random wireless channel fluctuations, user movement and signal interference. We instead learn this unknown transition law implicitly via continuous environment interaction.

We design the reward function to balance throughput efficiency and network stability as $r_t = \alpha \cdot \text{throughput}_t - \beta \cdot \text{delay}_t - \gamma \cdot \text{packet loss}_t$. Here α, β, γ serve as weighting coefficients to tune the trade-off between different performance targets.

Discount factor $\gamma \in (0, 1)$ weights the contribution of future rewards, which lets the agent optimize congestion control performance from a long-run perspective.

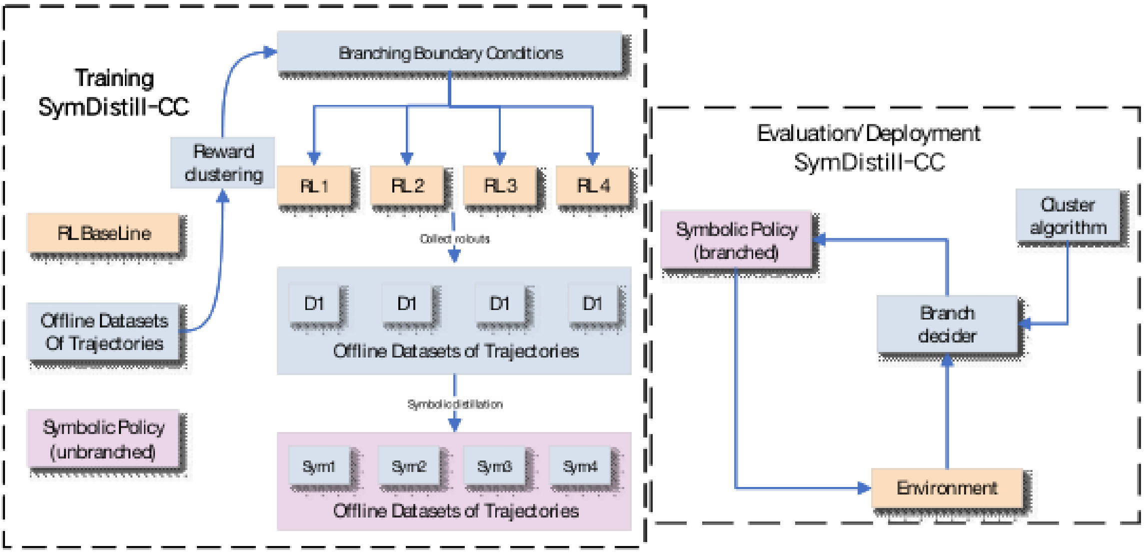

FIGURE 1. Structure Diagram of the SymDistill-CC Method

We train a WAPI-specific teacher agent using Proximal Policy Optimization (PPO) under the constructed MDP model. A teacher-student distillation framework is further applied to extract interpretable symbolic policies, resolving the black-box drawback of deep reinforcement learning and the decision delay present in wireless networks.

The process begins with trajectory collection, where deterministic decision trajectories of the trained teacher agent are gathered from multiple representative WAPI scenarios to construct a dataset:

$$\mathcal{D} = \{(o_t, a_t) \mid t = 1, 2, \ldots, N\} \tag{3.3}$$

Among them, $o_t = [\mathrm{SNR}_t, \mathrm{FLR}_t, \mathrm{QL}_t, \mathrm{QoS\_Lv}_t, \eta_t]$ denotes the WAPI vector statistical symbol, and a_t denotes the transmission rate decision; secondly, rule space initialization is carried out to define the search space for symbolic rules and add WAPI-specific operations; finally, iterative optimization is conducted through genetic programming, with cross-entropy loss as the feedback signal to realize rule mutation, crossover, and pruning, and select the optimal symbolic strategy.

The transmission rate decision for congestion control in this study is defined as a binary classification task, and the action space of the agent only contains two types of discrete actions: increasing the transmission rate and reducing/maintaining the transmission rate. The symbolic student policy is trained to approximate the teacher policy via a cross-entropy loss:

$$L_{\mathrm{CE}} = -\frac{1}{N}\sum_{t=1}^{N}\left[Y_t \log(P_t) + (1 - Y_t)\log(1 - P_t)\right] \tag{3.4}$$

Among them, Y_t is the teacher's decision label, $Y_t = 1$ indicates that the rate needs to be increased, $Y_t = 0$ indicates that the rate needs to be reduced/maintained, and P_t is the decision output probability of the student's symbolic strategy for o_t, which is calculated by combining symbolic rules, e.g. $P_t = \sigma(\mathrm{SNR\_TH}(o_t, 20\mathrm{dB}) \wedge (\mathrm{FLR}_t < 3\%))$, $\sigma(\cdot)$ is the Sigmoid function, and N is the total number of trajectory samples in the dataset.

To analyze the impact of distillation on policy performance, we define the expected distillation error as:

$$E_{\mathrm{distill}} = \mathbb{E}_{s\sim\mathcal{D}}[D_{\mathrm{KL}}(\pi_T(\cdot \mid s) \,\|\, \pi_S(\cdot \mid s))] \tag{3.5}$$

Where π_T and π_S denote the teacher and student policies, respectively.

Under standard policy approximation assumptions, the performance gap between the student and teacher policies can be bounded as:

$$|J(\pi_S) - J(\pi_T)| \le C\sqrt{E_{\mathrm{distill}}} \tag{3.6}$$

Where C is a constant related to the reward scale and horizon length. This indicates that minimizing the distillation loss leads to bounded degradation in control performance.

To improve generalization across diverse WAPI environments, the distillation dataset is constructed from multiple heterogeneous scenarios, ensuring broad state coverage. In addition, a branch reinforcement learning framework is introduced to partition the state space into multiple non-overlapping contexts, with each branch specializing in a subset of network conditions.

K-means clustering is applied to reward-feature representations to automatically determine context boundaries:

$$d(C_i, C_j) = \sqrt{\sum_{m=1}^{5} w_m \left(x_{i,m} - x_{j,m}\right)^2} \tag{3.7}$$

Among them, C_i, C_j represents the j-th, i, j types of contexts, $x_{i,m}$ represents the m feature of C_i, and w_m represents the feature weight.

The number of branches $K = 3$ is chosen as a trade-off between expressiveness and complexity. Empirical results show performance saturates at $K \geq 3$, and sensitivity analysis over $K = 2, 3, 4$ indicates low sensitivity. Reward-guided clustering is adopted instead of raw state clustering, as the reward implicitly encodes multiple performance objectives (throughput, delay, loss), enabling a task-oriented and low-dimensional partition. Feature weights in Eq. (3.4) are aligned with reward coefficients to ensure consistency. To reduce boundary oscillation, a hysteresis mechanism is introduced for stable context switching. The computational overhead of clustering is negligible due to its low dimensionality. Finally, misclassification has limited impact since adjacent clusters exhibit similar characteristics, and branch policies trained on overlapping data improve robustness.

To determine the optimal number of contexts, we conduct a sensitivity analysis with $K \in \{2, 3, 4, 5\}$. When $K = 2$, the contexts are too coarse, leading to large intra-cluster variance and performance loss. When $K \geq 4$, marginal performance gain diminishes while computational overhead increases significantly. $K = 3$ achieves the best trade-off between expressiveness and overhead, corresponding to typical power IoT scenarios: low-bandwidth, medium-bandwidth, and high-bandwidth.

To prevent overfitting and ensure interpretability, explicit structural constraints are imposed on the symbolic policy during genetic programming, including the maximum tree depth $d_{\max}$ and the maximum number of nodes $N_{\max}$. The optimization objective is further augmented with a complexity regularization term, leading to the objective function $L = L_{\mathrm{CE}} + \lambda \cdot \mathrm{Complexity}(T)$, where T denotes the symbolic tree and the weight coefficient λ controls the trade-off between model accuracy and simplicity. This constraint mechanism effectively inhibits the excessive growth of symbolic expressions and significantly enhances the generalization capability and stability of the policy.

We stop policy performance drops in the distillation process. We add two supporting mechanisms. The first one is Teacher Consistency Regularization. It checks the student policy against the teacher policy at fixed time points. It keeps the two sets of decision rules consistent. The second one is Performance-Guided Selection. It picks usable symbolic policies in genetic programming. It looks at two metrics. One is small loss value. The other is total reward the policy gets inside the network environment. These mechanisms collectively ensure that the distilled symbolic policy preserves the key performance characteristics of the PPO-trained teacher policy and avoids collapsing into suboptimal strategies.

The above strategy mainly addresses the adaptive issue of WAPI's macro transmission rate. However, in wireless scenarios, the QoS (Quality of Service) requirements of multiple services vary significantly, so further optimization of packet-level scheduling is needed.

C. ECN-FPS-BASED PACKET SCHEDULING METHOD

In WAPI-based multipath transmission systems, a secure connection can simultaneously use multiple network paths to transmit data, such as using both Wi-Fi and 5G managed by WAPI. These paths may share the same network bottleneck or be independent of each other. If paths share a bottleneck, their congestion window growth should not be independent but should be coupled under the unified management of WAPI security associations to avoid exacerbating link congestion due to blind competition among multiple subflows. This study uses the ECN mechanism to detect shared bottlenecks. First, a queue length threshold is set in ECN-capable routers. When the queue length exceeds this threshold, the router marks the ECN field of passing packets as “11”. Then, after receiving marked packets, the receiver sets the ECE value to “1” in the ACK packet returned through the WAPI secure channel. The sender determines that the corresponding subflow path is congested based on this ACK packet and can evaluate the congestion situation of each subflow based on the number of marked ACK packets.

The shared bottleneck detection method is divided into two phases.

The first phase determines whether subflows share a bottleneck by checking the “observation window”, forming a “preliminary shared bottleneck set”. The observation window consists of past and future observation windows corresponding to packets marked with ECN “11”, with both window sizes being half of the congestion window. The second phase performs correlation verification on subflows in the preliminary set to confirm whether they truly share the same bottleneck, thereby obtaining the inal shared bottleneck set.

Assume a WAPI-based multipath connection contains several subflows, ordered by RTT from smallest to largest, i.e., $RTT_1 \leq \cdots \leq RTT_i \leq \cdots \leq RTT_n$. When subflow i is ready to send new data, the ECN-FPS method estimates, within subflow i's half round-trip time, the number of packets that all faster subflows can send, denoted as M. To ensure all packets arrive at the receiver in order, slower subflow i cannot send from the beginning of the buffer (denoted as S_1), but should send from position S_i. Their relationship is shown in Formula (3.8):

$$S_i = S_1 + M \tag{3.8}$$

If M_j is the number of packets that subflow j can transmit within $RTT_i/2$, then $M = \sum_{j=1}^{i-1} M_j$. If $i = 1$, then $M = 0$, indicating that i is the fastest subflow. The key lies in predicting M_j for subflow j. This is divided into two cases: When subflow j is in the shared bottleneck set, its window growth is constrained by other subflows. Within $RTT_i/2$ subflow j transmits $\lambda = (RTT_i/2)/RTT_j$ rounds of data.

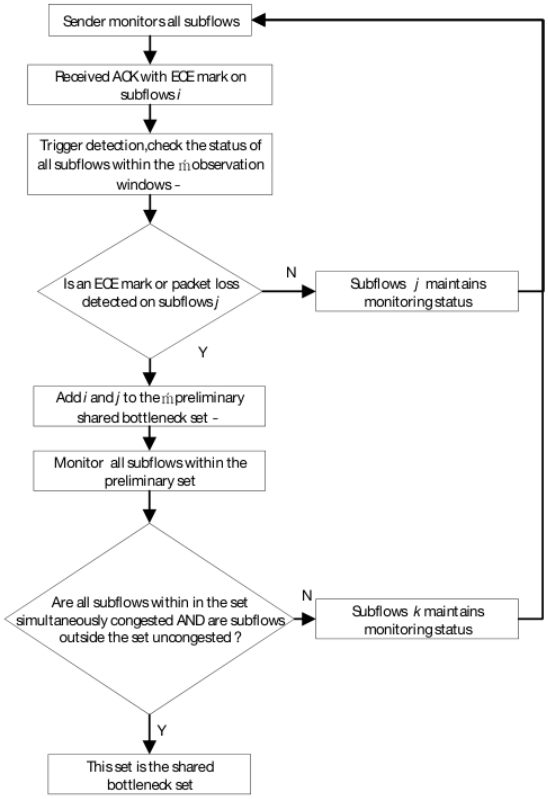

FIGURE 2. Shared Bottleneck Detection Method

Therefore, it needs m_j RTTs to increase the congestion window by 1. Thus:

$$M_j = \frac{\lambda/m_j}{2}\left(2w_j + \lambda/m_j - 1\right)m_j \tag{3.9}$$

Where w_j is the congestion window of subflow j.

When subflow j does not belong to any bottleneck set, its behavior is similar to traditional TCP, where the window increases by 1 per RTT, i.e., $m_j = 1$. At this time:

$$M_j = \frac{\lambda}{2}\left(2w_j + \lambda - 1\right) \tag{3.10}$$

D. ADAPTIVE RETRANSMISSION SCHEME FOR POWER-IOT BASED ON WAPI

This paper proposes a WAPI-compliant adaptive re-transmission control method tailored to the many-to-one traffic pattern typical of smart-grid IoT, which carries diverse critical businesses including power grid state sensing, equipment monitoring and real-time dispatch command transmission with distinct QoS requirements. The scheme integrates WAPI's mandatory security handshake with a real-time, state-aware re-transmission engine that dynamically chooses either unicast or multicast repair on a per-packet basis, so that reliability, bandwidth efficiency and node energy are jointly optimised. The protocol operates in three stages: network-state sensing, re-transmission decision, and threshold self-tuning. Figure 3 gives the control flow.

All radios have already completed WAPI certificate authentication and pairwise key negotiation; hence every subsequent control or data frame is integrity-protected and encrypted. Consider a base-station that serves N IoT terminals. Let p be the wireless loss-rate, PS the packet size, B_i the link bandwidth and S the pre-configured multicast stream rate. At time slot t we compute the average network load β_t and the average link utilisation α_t as

$$\begin{aligned}\beta_t &= \frac{S}{B_i} + \frac{PS\cdot(H_t^m + H_t^u)}{t\cdot N\cdot B_i},\\ \alpha_t &= \frac{S}{B_i} + \frac{PS\cdot(H_t^m\cdot N + H_t^u)}{t\cdot N\cdot B_i}.\end{aligned} \tag{3.11}$$

where H_t^m and H_t^u are the numbers of unicast and multicast retransmissions that have occurred up to slot t, and t is the observation window.

During the retransmission decision process, if the current network load does not exceed the set threshold:

$$\beta_t \le \frac{S}{B_i}(1+c) \tag{3.12}$$

Then, unicast retransmission is prioritized to prevent non-packet-loss terminals from receiving redundant data, thereby reducing their link utilization and energy consumption. Otherwise, based on the current number of packet-loss terminals

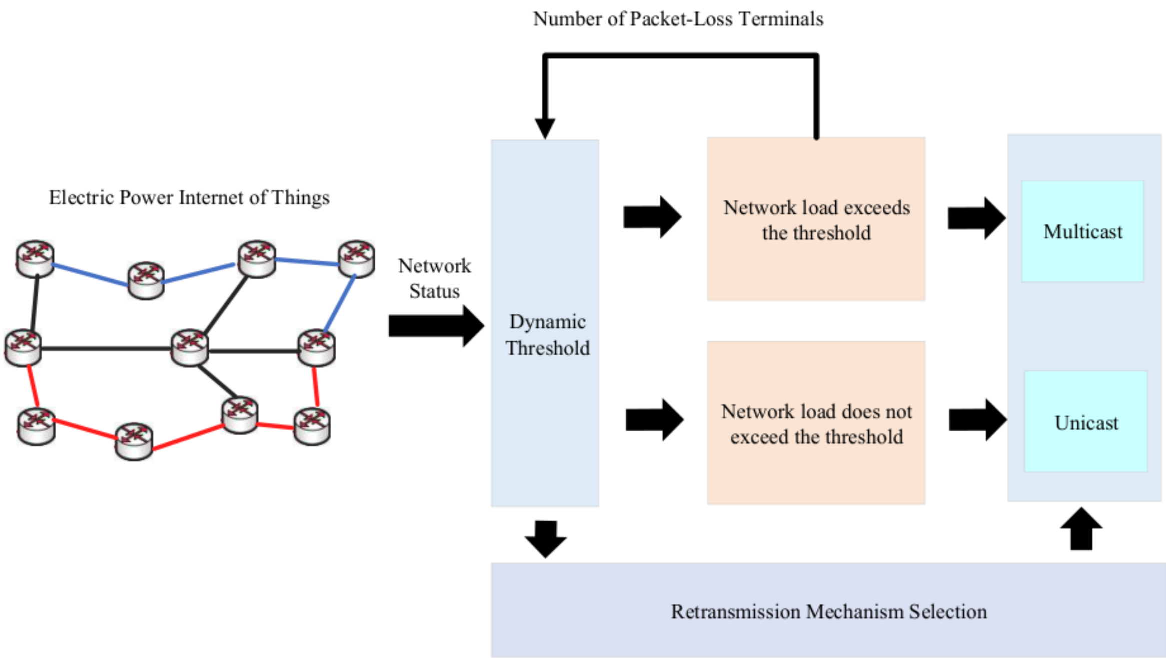

FIGURE 3. Schematic Diagram of Adaptive Retransmission in the WAPI-Based Power IoT

K, the expected network load under unicast and multicast modes is calculated separately:

$$\begin{aligned} \mathbb{E}[\beta_u \mid K] &= \frac{S \cdot K \cdot \mathbb{E}[R_u] + S}{N \cdot B_i}, \\ \mathbb{E}[R_u] &= \frac{1}{1-p}, \\ \mathbb{E}[\beta_m \mid K] &= \frac{S \cdot \mathbb{E}[R_m \mid K] + S}{N \cdot B_i}, \\ \mathbb{E}[R_m \mid K] &= 1 + \sum_{n=1}^{\infty} \left[1 - (1-p^n)^K\right]. \end{aligned} \tag{3.13}$$

By comparing the sizes of the two, the retransmission mode with the smaller expected load is selected. Throughout this process, the security context provided by WAPI ensures the authenticity and integrity of retransmission requests and data, preventing malicious exploitation of the retransmission mechanism. We further design a dynamic tuning mechanism for parameter c to cope with time-varying network conditions in power IoT systems. The mechanism flexibly modulates the aggressiveness of retransmission policies according to real-time network states and service demands, which delivers better overall system performance. This design balances network load and limited terminal resources while fully complying with WAPI security specifications.

It fits power IoT communication scenarios that demand high reliability and low power consumption, and support stable data transmission for distributed monitoring and scheduling tasks in smart grids.

IV. EXPERIMENTS AND ANALYSIS

A. EXPERIMENT DESCRIPTION

We build a complete test environment with common network scenes. The setup can fully check the performance and practical value of our new methods. We use two simulation tools NS-2.35 and NS-3.36. One tool builds wireless communication scenes. The other builds power IoT communication scenes. Users can test our algorithms under different real running states. All tests split into three main parts. We first test the reinforcement learning congestion control method. We run the test on three different WAPI network settings. The settings include steady packet loss, changing bandwidth and multiple data streams running at the same time. We collect key data like throughput, delay jitter, link usage and computing cost. These data show how the method actually works. We then carry out comparison tests for the ECN-FPS multipath scheduling algorithm. The main test target is its ability to cut disordered data packets. Third, we test the WAPI-aware adaptive retransmission scheme under heavy packet loss to check its adaptability for power IoT deployments. These multi-dimensional, multi-metric experiments clearly demonstrate the strengths and practical value of our methods. We select critical parameters for congestion window tuning and packet scheduling based on real-world WAPI deployment data and run sensitivity tests to guarantee reliable results. We set values for the congestion window growth threshold, ECN marking threshold and forward prediction window size using standard buffer capacities and WAPI wireless link RTT ranging from 10 ms to 100 ms. Auxiliary tests with parameters adjusted by $\pm 30\%$ show that all compared methods maintain consistent performance rankings, which proves our designs have strong robustness and do not rely on precise parameter tuning. We target two common power IoT traffic types: monitoring data and control signaling. We track several key values to judge algorithm performance. These values are throughput, link usage, number of disordered packets and network load. We also record end-to-end delay, real-time running status and QoS difference as extra reference data.

B. EXPERIMENTS ON WAPI AND REINFORCEMENT LEARNING-BASED CONGESTION CONTROL METHOD

We construct a WAPI wireless network testbed on the NS-3 simulator to replicate standard indoor and outdoor operating conditions. The simulation covers bandwidth from 10 to 50 Mbps, one-way propagation delay between 10 and 100 ms, and packet loss rates spanning 0.1% to 10%, with both random loss and congestion-triggered loss included. For comparative algorithms, traditional congestion control algorithms (TCP CUBIC, TCP BBR) and RL-based methods (Aurora) are selected to cover performance baselines of different technical routes.

The experiment sets up three types of core test scenarios, and each scenario is tested repeatedly 10 times, with each test lasting 600 seconds—the first 100 seconds are the warm-up period, and the subsequent 500 seconds are the data collection period. First, a static bandwidth and packet loss scenario is set, with the bandwidth fixed at 30 Mbps, and mixed packet loss is set in gradients of 2%, 4%, 6%, and 8%, where congestion-induced packet loss accounts for 60% and random packet loss accounts for 40%, so as to verify the algorithm's ability to distinguish between packet loss types; second, a dynamic bandwidth scenario is set, where the bandwidth switches periodically every 10 seconds in the order of 10 Mbps→30 Mbps→50 Mbps→30 Mbps→10 Mbps with no packet loss, which is used to test the algorithm's response speed to bandwidth fluctuations; finally, a multi-flow competition scenario is constructed, where 3 TCP flows for file transmission and 2 UDP flows for real-time video run concurrently with a fixed bandwidth of 30 Mbps, so as to evaluate the algorithm's fairness in resource allocation. The evaluation indicators and their calculation methods include throughput, delay jitter, link utilization, and computational overhead—among them, throughput refers to the number of valid bytes successfully transmitted per unit time, excluding retransmitted data; delay jitter is the standard deviation of transmission delays of adjacent data packets; link utilization is the ratio of actual throughput to the maximum bandwidth of the link; and computational overhead is calculated by using the built-in Profiler of NS-3 to count the FLOPs (Floating-Point Operations Per Second) of the algorithm's congestion judgment and decision-making modules, while excluding the overhead of data encapsulation and decapsulation.

The comprehensive performance of SymDistill-CC in WAPI scenarios is significantly superior to that of the comparative algorithms. In the static packet loss scenario, when the packet loss rate is 6%, its throughput reaches 28.2 Mbps, which is 17% higher than that of TCP CUBIC and 7.2% higher than that of TCP BBR. At the same time, the accuracy of packet loss type identification reaches 92%, which can avoid unnecessary rate reduction caused by misjudging random packet loss. In the dynamic bandwidth scenario, when the bandwidth switches from 10 Mbps to 50 Mbps, its response delay is only 0.8 seconds, 27% lower than that of TCP BBR, and the delay jitter is stable at 15 ms, with performance superior to TCP CUBIC and Aurora. In the multi-flow competition scenario, its link utilization is stably maintained at 89.3%, the fairness index (Jain's Index) reaches 0.9, and the throughput difference among various flows is controlled within $\pm 10\%$ (with a maximum of 3.2 Mbps and a minimum of 2.8 Mbps). In contrast, Aurora has a fairness index of only 0.78, and its throughput difference reaches $\pm 30\%$ (with a maximum of 3.8 Mbps and a minimum of 2.2 Mbps). In terms of computational overhead, the FLOPs of SymDistill-CC are 2.3×10^5 per second, 30% lower than that of Aurora and close to that of TCP CUBIC, which can meet the lightweight requirements of WAPI terminals such as IoT devices.

In addition, average end-to-end delay and delay jitter are evaluated. Under dynamic bandwidth scenarios, SymDistill-CC maintains a stable end-to-end delay of 25–35 ms with jitter within 15 ms, satisfying the real-time requirements of monitoring data. For high-priority control data, it outperforms baseline algorithms in latency, meeting the low-delay demands of control services. Beyond reducing FLOPs, we further assess lightweight performance from a deployment perspective via average decision delay. Results show that Aurora requires over 12 ms per control decision due to neural network inference, which is impractical for real-time power IoT terminals. In contrast, SymDistill-CC reduces the decision delay to approximately 2 ms, verifying that symbolic distillation substantially improves real-time deployability beyond theoretical computational savings.

The computational complexity of the proposed SymDistill-CC algorithm is analyzed from the training phase and online inference phase, and the asymptotic complexity expressions of the core modules (PPO, genetic programming, branch decision) are given, where N is the number of training samples, D is the dimension of state features, K is the number of branches ($K = 3$), T is the number of genetic programming iterations, and M is the maximum number of nodes in the symbolic tree. PPO uses a deep neural network to fit the policy, and its training complexity is determined by the number of samples, feature dimension and network layers, with an asymptotic complexity of $O(N \cdot D \cdot \log D)$, which is a typical polynomial complexity. The complexity of genetic programming is determined by the number of iterations, symbolic tree scale and sample number. The symbolic tree scale is controlled by node and depth constraint, in this paper, with an asymptotic complexity of $O(T \cdot M \cdot N)$. Since M is a fixed constant it is actually a low-order polynomial complexity. The symbolic policy is a lightweight

TABLE I. Performance Comparison of Different Congestion Control Algorithms in WAPI Networks

	Static Bandwidth Loss Scenario		Dynamic Bandwidth Scenario	Multi-flow Competition Scenario	Computational Overhead (FLOPs Reduction Rate)	Avg. Decision Delay (ms)
Comparison Algorithm	Throughput Improvement Rate (%)	Packet Loss Recognition Accuracy (%)	Response Delay Reduction Rate (%)	Link Utilization (%)		
TCP CUBIC	0 (Benchmark)	65	−10 (Higher than TCP BBR)	82–84		1.8
TCP BBR	8–12	70	0 (Benchmark)	84–86		2.1
Aurora (RL-based)	12–15	80	15	80–82		12.6
SymDistill-CC	15–20	92	25	88–90		2.4

decision tree structure without neural network forward computation, only requiring feature matching and simple logical judgment, with an asymptotic complexity of $O(D)$. Lightweight K-means clustering and distance calculation are adopted for branching, and K is a fixed constant, with an asymptotic complexity of $O(K \cdot D) = O(D)$, which is negligible.

C. EXPERIMENTAL ANALYSIS OF ECN-FPS-BASED PACKET SCHEDULING METHOD

We conduct comparative experiments under identical settings to assess the proposed ECN-marking-based two-stage shared-bottleneck detection against Enhanced Multipath TCP (EMPTCP) and Dynamic Window Coupling (DWC). We further compare ECN-FPS with Lowest RTT First (LowRTT) and Forward Prediction Scheduling (FPS) to validate its scheduling performance. This experiment adopts the method proposed in Section 4.2 as the congestion control mechanism. The network topology of the simulation environment is shown in Figure 4,,where S represents the server and C represents the client. The key setup involves three subflows: subflow SF0 transmits independently, while subflows SF1 and SF2 share the same network bottleneck link. The bottleneck link has a bandwidth of 5 Mbps, a propagation delay of 30 ms, and a packet loss rate ranging from 0.0% to 1.0%, while the remaining paths are configured with a bandwidth of 100 Mbps and a delay of 10 ms.Additionally, other competing traffic exists in the network to simulate real network congestion scenarios.

Figure 5 compares the shared-bottleneck detection precision of ECN-FPS (the proposed ECN-marking-based two-stage detector), EMPTCP, and DWC as background traffic increases from 0 to 20 Mbps. At 0 Mbps, the three schemes start close to saturation (98%, 96%, and 95%, respectively), indicating reliable detection under light contention. As the traffic rises to 20 Mbps, precision decreases for all methods, but ECN-FPS remains the highest at 84%, outperforming EMPTCP (81%) and DWC (74%). The cumulative degradation from 0→20 Mbps is 14 percentage points for ECN-FPS, 15 for EMPTCP, and 21 for DWC, showing that DWC is the most sensitive to cross-traffic perturbations while the proposed detector maintains a clearer margin at heavy contention.

Figure 6 presents the corresponding recall values. All three methods start with recall rates of 98%, 96% and 95% under 0 Mbps bandwidth. When bandwidth rises to 20 Mbps, ECN-FPS reaches 70%, outperforming EMPTCP (65%) and DWC (61%). ECN-FPS only loses 28 percentage points from 0 Mbps to 20 Mbps, compared with 31 points for EMPTCP and 34 points for DWC. This comparison shows our proposed method fails to detect fewer shared bottlenecks under heavy congestion. Taken together, Figures 5 and 6 demonstrate that ECN-FPS achieves a more favorable precision-recall balance than EMPTCP and DWC in the shared bottleneck simulation environment.

Table 2 lists the total out-of-order packet counts of three scheduling algorithms, LowRTT, FPS and ECN-FPS, under various receiver buffer capacities. The metric recorded in the table counts all misordered packets accumulated at the receiver throughout the 120-second test with a steady application sending rate. All competing schemes run with identical traffic loads, path settings and test durations, so the measured values can be directly compared without extra normalization steps. This indicator isolates packet disorder generated only by scheduling decisions and eliminates interference from inconsistent traffic volumes. Table 2 shows that the count of out-of-order packets rises for all three tested approaches as the receiver buffer size grows. Larger buffers hold more pending packets in queues, which creates more opportunities for misordered delivery. Among all schemes, ECN-FPS have the slowest growth rate, meaning its performance suffers the least from buffer size adjustments. It also maintains better results than LowRTT in all cases, which suggests that forward-looking prediction strategies works better than algorithms that only react to instantaneous network states. ECN-FPS outperforms FPS because ECN marks detect early queue buildup in shared bottlenecks, much earlier than loss- or delay-based signals. This enables the sender to maintain coupled subflow groups stably. With accurate shared bottleneck identification, ECN-FPS restricts parallel transmission of coupled subflows and uses forward-prediction-based sending constraints to suppress traffic bursts, effectively reducing packet reordering at the receiver. In terms of end-to-end delay, ECN-FPS lowers average latency by 10–18% versus LowRTT by mitigating packet disorder and reducing receiver buffer queuing, significantly improving real-time performance for control data. The proposed scheduling mechanism meets differentiated QoS requirements for both monitoring and control data.

In addition to reducing packet reordering, ECN-FPS maintains good subflow fairness without penalizing high-RTT paths. Mechanistically, unlike LowRTT or delay-

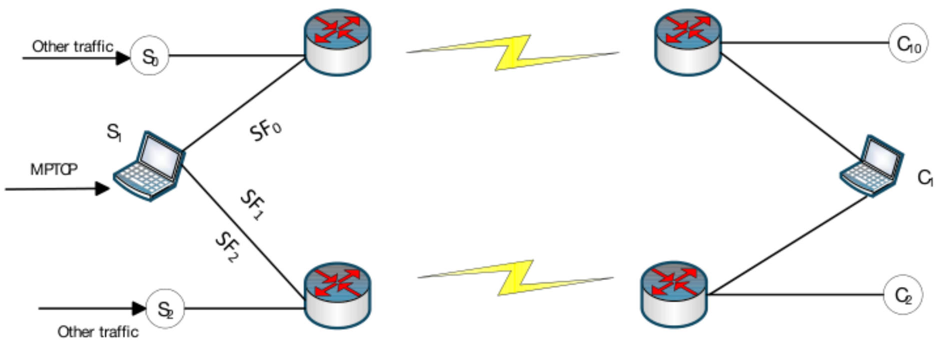

FIGURE 4. Simulation Topology Diagram

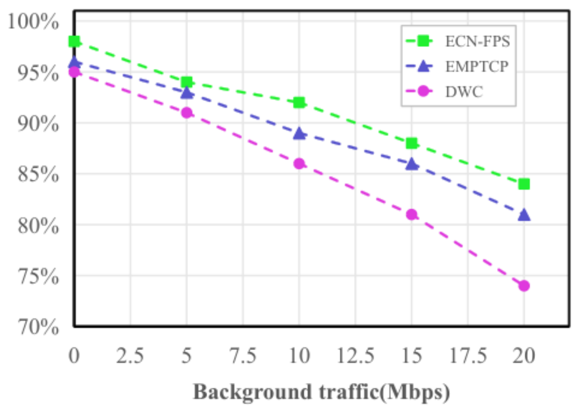

FIGURE 5. Shared Bottleneck Detection Precision Comparison

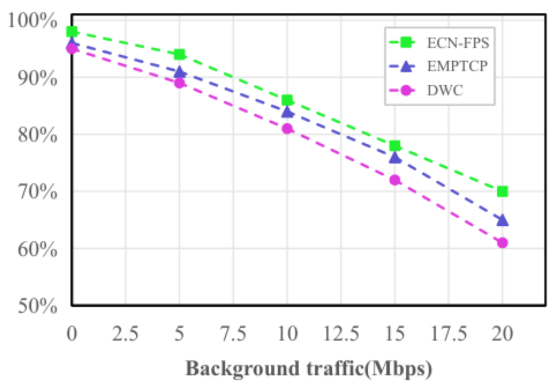

FIGURE 6. Shared Bottleneck Detection Recall Comparison

priority schedulers that rely on RTT ranking, ECN-FPS activates coupling constraints only when ECN marks signal shared queue buildup. Outside congestion events, each subflow operates independently under its native congestion window. This event-driven scheduling avoids systematic bias toward low-delay paths.

ECN-FPS uses forward prediction and ECN signals together. It can catch changing network states with higher precision. It fits tasks that need low delay and ordered data,

TABLE II. Comparison of Out-of-Order Packet Counts Across Different Methods

Receive Buffer Size ($\times$65536 byte)	LowRTT	FPS	ECN-FPS
0.25	6	3	3
0.5	15	9	7
1	25	20	14
2	31	25	16
3	34	28	19
4	37	31	21
5	40	33	23

like HD video transmission and remote device control.We test SymDistill-CC for long-term stability on dynamic power IoT networks. We watch changes of congestion window and compare this algorithm with TCP CUBIC and Aurora.The test lasts 600 seconds. Bandwidth changes from 10 Mbps to 50 Mbps, and random packet loss sits between 0 and 2 percent.Figure X shows SymDistill-CC has steady congestion window curves. It converges faster with less fluctuation and keeps high throughput all the time.The algorithm does not expand window too fast or roll back window values often during long testing. It keeps stable running on unstable power IoT networks with WAPI..

D. EXPERIMENTS ON ADAPTIVE RETRANSMISSION METHODS FOR THE POWER IOT BASED ON WAPI

We build a multi-terminal communication testbed on the NS-2 simulator that matches the real scale of regional smart grid power IoT deployments, to verify the performance of our WAPI-compatible adaptive retransmission scheme. We configure the simulation parameters as listed below: each wireless link carries 14 Kbps bandwidth with 50 ms propagation delay, the multicast stream transmits at 2.8 Kbps, each packet holds 180 bytes of payload, the number of terminals scales from 5 to 40, and the packet loss rate adjusts from 0.01 to 0.50. We also add a constant 10 ms delay to all WAPI security authentication steps to replicate real authentication overhead.

We evaluate three distinct retransmission policies in our tests. The static unicast policy relies solely on unicast for all retransmission packets, while the static multicast scheme transmits every retransmission via multicast chan-

FIGURE 7. Comparison of Average Link Utilization

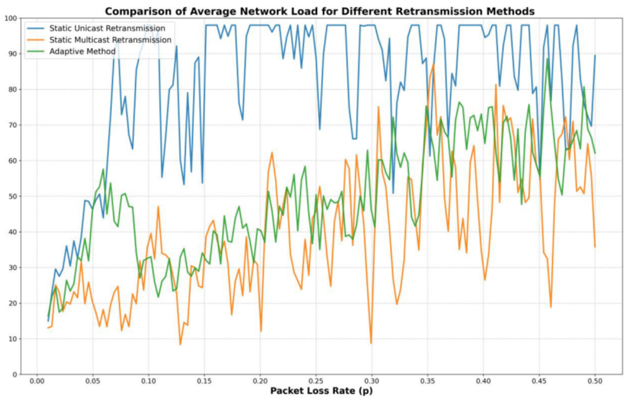

FIGURE 8. Average Network Load Comparison

nels. Our proposed adaptive strategy switches between unicast and multicast retransmission modes dynamically under the WAPI security framework. We measure average link utilization, average network load and total retransmission times as key performance metrics. Figure 7 plots average link utilization and network load across the three schemes under varying packet loss rates with the terminal number fixed at 20. Figure 8 shows that static multicast retransmission pushes link utilization sharply upward as packet loss grows, which surpasses 90% when p reaches 0.3 and may trigger congestion on terminal wireless interfaces. Static unicast retransmission yields moderate link utilization, yet its network load rises far above the other two methods under severe packet loss. Our adaptive scheme defaults to unicast retransmission for p below 0.1 and keeps link utilization under 20%. Once p hits or exceeds 0.1, the mechanism switches to multicast retransmission automatically, which limits overall network load and achieves lower link utilization than the static multicast baseline.

Table 3 lists full performance indicators for all three schemes under the setting $p = 0.2$ and $M_{mk} = 20$. Our approach generates network load comparable to static multicast, keeps link utilization at a low level, and records the smallest number of retransmissions, which proves its

TABLE III. Performance Comparison of Three Retransmission Methods

Retransmission Method	Average Link Utilization (%)	Average Network Load (%)	Total Number of Retransmissions
Static Unicast	36.5	72.9	128
Static Multicast	68.3	43.6	24
The Proposed Adaptive Method	40.9	45.8	19

strong adaptability on devices with limited resources. We also test our method on two typical power IoT traffic types, monitoring data and control signaling, to verify its capacity to satisfy differentiated QoS constraints. When handling monitoring traffic, SymDistill-CC stabilizes end-to-end latency between 25 and 35 ms with jitter no higher than 15 ms, meeting the real-time requirements of periodic data collection. For high-priority control packets, our design cuts latency by 10–18% relative to LowRTT and reduces out-of-order packets, delivering low-latency, highly reliable transmission for critical control tasks. All test results verify that the overall framework can accommodate distinct QoS demands from various power IoT services.

Our experimental results show that the WAPI-integrated adaptive retransmission scheme balances network load and terminal resource usage without compromising communication security, which fits power IoT scenarios requiring high reliability and low power draw.

V. CONCLUSION

This work targets three key challenges existing in power IoT wireless networks: uninterpretable congestion control models, severe packet disorder over multipath links, and static retransmission strategies that cannot adapt to varying network conditions. We first put forward SymDistill-CC, a congestion control algorithm built on symbolic distillation. The method converts opaque neural network policies into readable symbolic decision trees, cutting computation costs while retaining high throughput. Tests show its throughput exceeds TCP CUBIC by 17% at maximum, and it lowers computational overhead by 30% relative to Aurora. We then design an ECN-based Forward Predictive Scheduling (ECN-FPS) mechanism. Combining ECN marking with shared bottleneck detection, the scheme arranges packet sending sequences in advance. Simulations prove it can decrease out-of-order packets by as much as 42.5% versus the classic LowRTT scheduler and keep data packets delivered in order. Lastly, we develop a WAPI-native adaptive retransmission approach that picks proper retransmission modes according to real-time network measurements. Experiments demonstrate that at a packet loss rate of 0.2%, the network load is maintained at 45.8% while reducing the total retransmission count to 19, achieving a balance between security and efficiency.

In summary, this paper systematically enhances the communication performance of the power Internet of Things across three dimensions: algorithmic interpretability, transmission ordering, and adaptive retransmission. It provides effective technical support for building highly reliable, low-overhead smart grid communication systems. Future work will focus on cross-layer joint optimization to address more complex heterogeneous network environments. Future work will focus on cross-layer joint optimization to address more complex heterogeneous network environments, and will further conduct systematic evaluations of end-to-end delay, real-time performance, and differentiated QoS for multi-service scenarios including monitoring data and control data, to complete full-scenario performance assessment for WAPI-enabled power IoT. Additionally, future efforts will explore large-scale network validation and integration with emerging network optimization mechanisms, so as to further expand the depth and breadth of this research.

ACKNOWLEDGMENT

Financial support from the State Grid Corporation of China Science and Technology Project (No. SGSXDK00HLJS2500142) is gratefully acknowledged.

REFERENCES

[1] T. Zhang and S. Mao, “Machine learning for end-to-end congestion control,” *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 52–57, 2020.

[2] B. Gupta, K. K. Gola, N. Jee, and P. Dimri, “Energy-efficient routing protocol for congestion control in wireless sensor network,” in *Proc. Int. Conf. Wireless Commun. Signal Process. Netw. (WiSPNET)*, Chennai, India, 2022, pp. 385–390.

[3] X. Xiao, M. Zhao, and Y. Zhu, “Multi-stage resource-aware congestion control algorithm in edge computing environment,” *Energy Rep.*, vol. 8, pp. 6321–6331, 2022.

[4] S. Rene, O. Ascigil, I. Psaras, and G. Pavlou, “A congestion control framework based on in-network resource pooling,” *IEEE/ACM Trans. Netw.*, vol. 30, no. 2, pp. 683–697, 2021.

[5] Z. Li, H. Xun, Y. Miao, W. Zhang, P. Luo, and K. Liu, “FleCom: A flexible congestion control protocol in named data networking,” in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2023, pp. 1–6.

[6] S. N. Mishra and M. Khatua, “Game theoretic congestion control to achieve hard reliability in mission-critical IoT,” *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, 2024.

[7] Z. Yu, C. Hu, J. Wu, X. Sun, V. Braverman, M. Chowdhury, Z. Liu, and X. Jin, “Programmable packet scheduling with a single queue,” in *Proc. ACM SIGCOMM Conf. (SIGCOMM ’21)*, 2021, pp. 179–193.

[8] J. Wu, G. Zhang, J. Nie, Y. Peng, and Y. Zhang, “Deep reinforcement learning for scheduling in an edge computing-based industrial Internet of Things,” *Wireless Commun. Mobile Comput.*, vol. 2021, no. 1, Art. no. 8017334, 2021.

[9] J. Han, K. Xue, Y. Xing, J. Li, W. Wei, D. S. L. Wei, *et al.*, “Leveraging coupled BBR and adaptive packet scheduling to boost MPTCP,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7555–7567, 2021.

[10] D. A. Hayes, M. Welzl, S. Ferlin, R. Ros, and S. Islam, “Online identification of groups of flows sharing a network bottleneck,” *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2229–2242, 2020.

[11] G. Xie, H. Chen, L. Xie, and K. Wang, “Shared bottleneck detection based decoupled congestion control for multipath TCP in wireless networks,” *Crossref*, 2024.

[12] D. Hu, Y. Bai, D. Dong, S. Huang, and X. Liao, “Converging credit-based and reactive datacenter transport using ECN and RTT,” in *Proc. IEEE Int. Conf. High Perform. Comput. Commun.*, 2020, pp. 433–440.

[13] K. Xue, J. Han, D. Ni, W. Wei, Y. Cai, Q. Xu, *et al.*, “DPSAF: Forward prediction based dynamic packet scheduling and adjusting with feedback for multipath TCP in lossy heterogeneous networks,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1521–1534, 2017.

[14] L. Liu, C. Liu, J. Hu, Z. Liu, J. Zhang, and T. Fang, “Deep learning-based channel state prediction and access control for WAPI systems in unlicensed bands,” in *Proc. SPIE—Int. Soc. Opt. Eng.*, Dalian, China, vol. 13805, Sep. 2025, pp. 78–82.

[15] P. Guo, J. Zhang, Y. Wang, H. Li, Z. Xue, Y. Peng, *et al.*, “HELDR: Packet loss detection and retransmission for live streaming hyper-edge network,” in *Proc. IEEE/ACM 33rd Int. Symp. Quality Service (IWQOS)*, 2025, pp. 1–10.

[16] W. Wang, J. Han, K. Xue, J. Li, K. Ding, and R. Li, “NetRT: Enhancing RDMA with retransmission offloading in data center networks,” in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2025, pp. 1–10.

[17] P. Huang, G. Chen, X. Zhang, C. Liu, H. Wang, H. Shen, *et al.*, “Fast and scalable selective retransmission for RDMA,” in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2025, pp. 1–10.

[18] S. J. Yu and C. Mun, “Adaptive DENM retransmission for enhancing reliability of mixed V2X traffic,” *IEEE Commun. Lett.*, vol. 29, no. 7, pp. 1724–1728, 2025.

[19] Z. Lin, X. Zhang, Z. Li, and Y. Liu, “Adaptive retransmission for efficient wireless federated learning,” in *Proc. 5th Int. Conf. Electron., Circuits Inf. Eng. (ECIE)*, 2025, pp. 33–36.