Semantic-Aware Multipath TCP: Enhancing Industrial IoT Flexible Manufacturing Networks with GAN-Based Packet Recovery

Deajin Han, Nhu-Ngoc Dao, Sungrae Cho, and Woongsoo Na

***Abstract*—Flexible manufacturing in Industrial Internet of Things (IIoT) environments leverages these capabilities to adapt dynamically to changing demands. However, IIoT applications face significant challenges in ensuring reliable data transmission due to high packet loss, resource constraints, and strict latency requirements. Multipath TCP (MPTCP) has emerged as a critical protocol for IIoT flexible manufacturing, leveraging multiple network paths to enhance throughput and reliability. Despite its advantages, MPTCP suffers from Head-of-Line (HOL) blocking. This paper addresses the challenge of reliable data transmission in IIoT flexible manufacturing networks, focusing on the latency and throughput degradation caused by Head-of-Line (HoL) blocking in MPTCP. We propose Semantic-Aware MPTCP (SA-MPTCP), integrating semantic communication with a GAN-based packet recovery mechanism that mitigates re-ordering delays and minimize retransmission overhead. SA-MPTCP prioritizes semantically significant data recovery, minimizes retransmission delays, and stabilizes congestion control. Evaluations show SA-MPTCP achieves 96% recovery accuracy in low-loss and over 85% in high-loss conditions, reducing delays by up to 90% compared to retransmission. SA-MPTCP enhances throughput and reliability, particularly for IIoT streaming and real-time control applications.**

***Index Terms*—Multipath-TCP, Generative Adversarial Network, Industrial IoT, Semantic Communication, and Packet Recovery**

I. Introduction

INDUSTRIAL Industrial Internet of Things (IIoT) is rapidly transforming the landscape of manufacturing and industrial processes by enabling seamless connectivity, advanced automation, and data-driven decision-making [1]. A critical component of this transformation is flexible manufacturing, which allows production systems to dynamically adapt to changing demands, product specifications, and market conditions. This adaptability relies on IIoT systems' ability to integrate distributed and heterogeneous devices, ensuring efficient collaboration between the service, application, and device domains [2].

Advanced AI tools, such as digital twins, machine learning (ML), and natural language processing (NLP), are key to this evolution [3]. These technologies enhance the capabilities of IIoT systems by enabling real-time data analysis, predictive maintenance, and adaptive workflows. In particular, human-machine interaction (HMI) in IIoT has evolved from simple information relay to context-aware, user-centric, and intelligent interfaces that improve decision-making and operational efficiency.

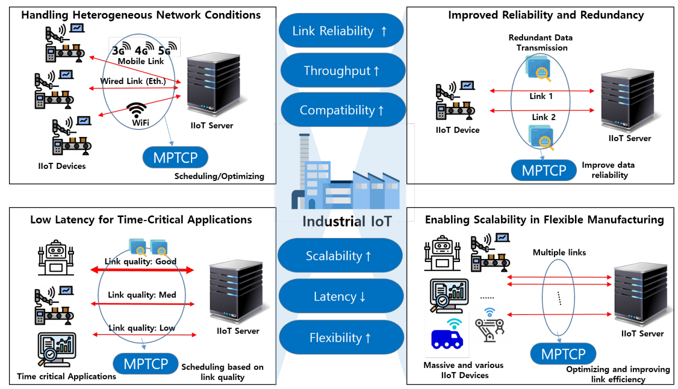

Fig. 1: The importance of MPTCP in IIoT flexible manufacturing networks

Multipath TCP (MPTCP) emerges as a vital communication protocol in this context, addressing the challenges of IIoT environments characterized by high mobility, variable bandwidth, and frequent disruptions. Fig. 1 shows the importance of MPTCP in IIoT flexible manufacturing networks. Recent studies have demonstrated that MPTCP can effectively enhance throughput and reliability in IIoT networks by utilizing multiple network interfaces simultaneously [4]. Furthermore, machine learning-based approaches to MPTCP path management have shown significant promise in optimizing network performance and reducing latency in dynamic IIoT environments [5]. By leveraging multiple network paths, MPTCP ensures reliable data transmission for time-critical applications like smart manufacturing and industrial automation, minimizing delays and operational inefficiencies. These applications often involve streaming data, such as remote vision and robotic control, where MPTCP's capabilities are particularly beneficial.

Despite its advantages, MPTCP encounters a critical bottleneck: Head-of-Line (HOL) blocking, which occurs when delays in one subflow impede the reassembly of subsequent packets. Studies have shown that the choice of congestion control algorithms can significantly influence the severity of HOL blocking in MPTCP-based communications [6]. HOL blocking critically impacts flexible manufacturing, where low-latency and high-reliability are essential. Traditional retransmission-based solutions exacerbate these delays, making them unsuitable for time-sensitive IIoT applications with high packet loss and asymmetric network conditions. To mitigate such issues, recent studies have proposed advanced scheduling algorithms

D. Han and W. Na are with the Department of Software, Kongju National University, Cheonan, Republic of Korea. (Email: djhan@bitlab.re.kr; wsna@kongju.ac.kr)

N.-N. Dao is with the Department of Computer Science and Engineering, Sejong University, Seoul, South Korea, email: nndao@sejong.ac.kr

S. Cho is with the School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, South Korea (e-mail: srcho@cau.ac.kr

This research was supported by the research grant of Kongju National University in 2025, Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2019-NR040074) and by the IITP (Institute of Information & Communications Technology Planning & Evaluation)-ITRC (Information Technology Research Center) grant funded by the Korea government (Ministry of Science and ICT) (IITP-2026-RS-2022-00156353)

that monitor transmission states to detect bufferbloat and prevent HoL blocking, thereby improving MPTCP performance in heterogeneous networks [7]. However, scheduling algorithms alone fail to provide a fundamental solution to HOL blocking in scenarios where certain links experience high traffic congestion. These methods struggle to adapt dynamically to severe traffic imbalances, which are common in IIoT flexible manufacturing networks and can lead to bufferbloat, further exacerbating HOL blocking.

To address these challenges, this study proposes Semantic-Aware Multipath TCP (SA-MPTCP), a framework tailored for IIoT flexible manufacturing networks. SA-MPTCP integrates semantic communication principles with a Generative Adversarial Network (GAN)-based packet recovery mechanism to enable real-time recovery of lost packets, minimize retransmission delays, and stabilize congestion control across subflows. By prioritizing semantically significant data recovery, SA-MPTCP aligns with the goals of flexible manufacturing, ensuring robust and efficient communication even under dynamic network conditions. Additionally, the integration of GANs with advanced HMI approaches introduces a new dimension of intelligence and adaptability, further enhancing IIoT flexible manufacturing networks. This study contributes to the advancement of next-generation IIoT communication protocols, laying the groundwork for smarter, more adaptive, and resource-efficient industrial networks. It should be emphasized that SA-MPTCP is not intended to redefine the semantics of the TCP/MPTCP protocol itself. Rather, it operates as an application-assisted communication framework built on top of MPTCP, where semantic relevance metadata generated at the application or edge layer is used to support transport-layer scheduling and recovery decisions while preserving existing industrial data formats and protocol structures.

The primary contributions of the proposed study are:

- **Reconceptualization of Transport-Layer Reliability through Semantic Awareness**: This work redefines transport-layer reliability by introducing semantic relevance as a new design dimension for multipath communication. Unlike conventional MPTCP or FEC-based protocols that focus solely on bit-level delivery accuracy, the proposed SA-MPTCP incorporates task-level meaning and contextual importance into its recovery and scheduling decisions. This semantic-aware perspective bridges the gap between data transmission and task performance, aligning the transport mechanism with the emerging 6G paradigm of meaning-oriented communication.
- **Introduction of a GAN-Driven Packet Recovery Mechanism for MPTCP**: This study proposes a novel GAN-based packet recovery mechanism specifically designed to mitigate the HOL blocking issue in MPTCP environments. By leveraging adversarial learning, this mechanism enables real-time recovery of lost packets, thereby reducing retransmission overhead and stabilizing the congestion window (CWND) across subflows. This approach addresses the unique challenges of IIoT environments, such as high packet loss, limited bandwidth, and stringent latency requirements.
- **Holistic Design for Seamless Communication in IIoT Flexible Manufacturing**: The proposed method employs a lightweight GAN architecture tailored for resource-constrained IIoT devices, ensuring efficient operation under diverse and heterogeneous network conditions. By adapting to dynamic IIoT environments, the mechanism supports real-time applications like smart manufacturing, industrial automation, and healthcare systems. Additionally, the use of semantic communication principles in the recovery process prioritizes critical data delivery, enhancing overall communication reliability.
- **Scalable Solution for Next-Generation IIoT Flexible Manufacturing**: This study introduces a scalable and modular framework tailored for IIoT applications requiring low latency and high reliability. The proposed mechanism seamlessly integrates with existing MPTCP protocols while maintaining flexibility for large-scale deployments. By addressing the limitations of traditional retransmission-based methods, the framework serves as a foundational technology for next-generation IIoT systems, enabling more flexible and adaptive manufacturing processes.
- **Dynamic Adaptation with Semantic Awareness**: SA-MPTCP incorporates semantic communication principles to dynamically prioritize and recover critical data based on its contextual importance. This ensures that high-priority packets are delivered reliably, optimizing communication performance even in resource-constrained environments.

II. Related Work

Addressing the HOL blocking problem in MPTCP has been extensively studied, with various methods proposed to mitigate its effects on data transmission. However, most existing approaches fall short when applied to IIoT networks due to the unique challenges these environments present, such as limited bandwidth, high latency, and the demand for real-time communication.

One of the traditional solutions involves congestion control mechanisms that adjust the CWND to balance transmission speeds across subflows. Morawski and Ignaciuk [6] proposed a dynamic CWND adjustment algorithm, which successfully mitigated speed imbalances in stable networks. However, in IIoT environments with frequent packet loss and high variability, this approach often leads to significant transmission delays, reducing its effectiveness. Similarly, retransmission-based strategies address packet loss by resending missing data. While straightforward, these methods introduce latency incompatible with real-time IIoT applications such as remote patient monitoring or automated industrial processes. Furthermore, state-of-the-art schedulers like BLEST [8] specifically target HOL blocking in MPTCP for heterogeneous networks, but they rely on estimation-based mechanisms and may not be as effective in highly dynamic IIoT environments as a proactive recovery approach.

Recent advancements in artificial intelligence (AI) have offered new opportunities for optimizing communication systems. AI models have been applied to tasks such as traffic prediction and error correction, demonstrating potential for improving data transmission efficiency. For instance, a GAN-based hybrid deep learning approach has been developed to enhance intrusion detection in IIoT networks, addressing the challenges of malicious attacks in dynamic environments [9]. GFlow, a Graph Neural Network (GNN)-based Deep Reinforcement Learning (DRL) solution, represents a significant advancement in this area. By leveraging fine-grained network states collected through In-band Network Telemetry (INT), GFlow optimizes flow scheduling and reduces congestion, achieving higher throughput and lower round-trip time (RTT) compared to traditional method [10]. However, the computational complexity of GFlow limits its direct application in resource-constrained IIoT environments. Also, as this method is inherently scheduling-based, it faces inherent limitations in resource-constrained environments.

GANs have emerged as a promising solution for addressing packet loss issues. GANs leverage adversarial learning, where a generator creates data and a discriminator evaluates its quality. By iteratively improving both components, GANs can

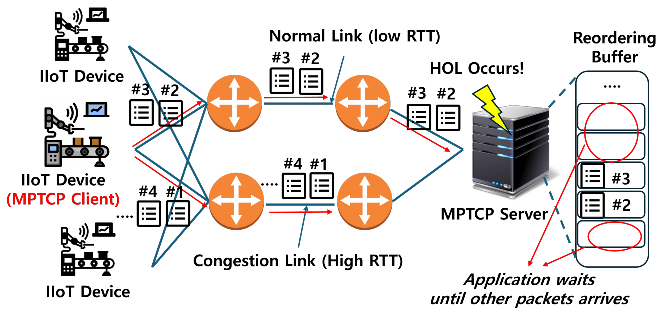

Fig. 2: Illustration of HoL blocking in multipath transmission.

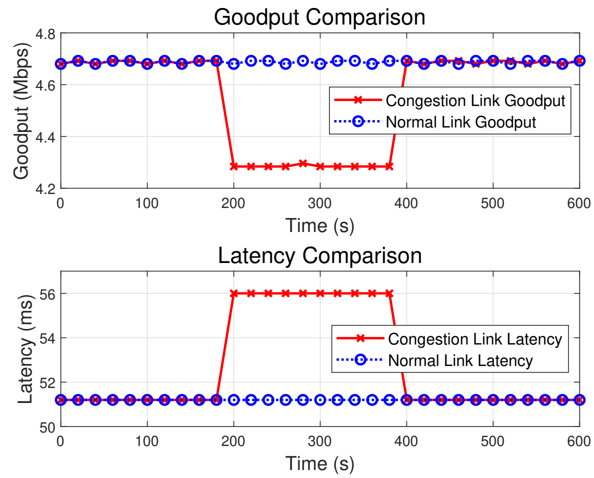

Fig. 3: Performance degradation due to asymmetric link

produce high-quality replacements for lost packets without relying on retransmissions [11], [12], [13], [14]. For example, Karmakar *et al.* proposed SmartCon, a GAN-based deep learning approach for auto link-configuration in Narrowband IoT (NB-IoT) networks. This method dynamically adapts transmission parameters to reduce packet loss and enhance transmission reliability in NB-IoT systems [15]. For instance, Aironi *et al.* proposed a time-frequency GAN-based method for audio packet loss concealment, demonstrating significant improvements in speech quality and intelligibility by effectively reconstructing lost audio packets [16]. Similarly, Cheng *et al.* introduced Attack-GAN, a GAN-based model designed to generate adversarial network traffic at the packet level, demonstrating the potential of GANs in crafting network traffic sequences and optimizing performance in dynamic conditions [17].

Despite these advances, existing GAN-based methods have not been specifically optimized for IIoT environments, where limited computational resources and the need for energy-efficient solutions pose significant challenges. Moreover, these GAN-based packet recovery methods have not been explored in the context of MPTCP to address HOL blocking issues.

IIoT networks introduce distinct constraints, including limited power and computational capacity, as well as frequent disruptions caused by mobility and interference. These challenges demand robust and efficient mechanisms capable of maintaining reliable communication under dynamic conditions. Traditional methods, while effective in general-purpose networks, often fail to address the specific requirements of IIoT communication systems.

This study builds upon prior research by proposing a GAN-based packet recovery mechanism explicitly tailored to IIoT networks. The proposed approach directly addresses HOL blocking by enabling real-time recovery of lost packets, reducing transmission delays and ensuring efficient utilization of network resources. adapting GAN architecture to IIoT-specific constraints, this work bridges the gap between existing HOL blocking solutions and the demands of modern IIoT systems.

III. Theoretical Background

A. Multipath TCP

MPTCP extends the traditional TCP protocol to support simultaneous data transmission over multiple network paths [18]. This capability significantly enhances network robustness and resource utilization, particularly in heterogeneous IoT environments where devices connect via diverse networks such as Wi-Fi, cellular, or LPWAN. By utilizing multiple paths, MPTCP ensures continuity in data transmission, even when individual paths experience congestion or failure.

A key feature of MPTCP is its sophisticated congestion control mechanism, which dynamically adjusts CWND across different paths. The primary goal of this mechanism is to maximize overall throughput while maintaining fairness to other network flows. For this purpose, MPTCP employs coupled congestion control algorithms such as the Linked Increases Algorithm (LIA). These algorithms ensure that the total congestion window size across all subflows is distributed efficiently, as expressed by the equation:

$$W_{total} = \sum_{i=1}^{N} W_i, \tag{1}$$

where (W_{total}) represents the total congestion window size across N subflows, and W_i denotes the congestion window of the i-th subflow. To ensure fairness with single-path TCP flows, MPTCP incorporates a fairness adjustment factor, α, which modulates the rate of change in the congestion window based on the RTT of each path. This relationship is captured by the equation:

$$\Delta W_i \propto \frac{1}{\alpha + \text{RTT}_i}, \tag{2}$$

where RTT_i is the RTT of the i-th path. Additionally, MPTCP balances traffic across paths by allocating more data to paths with higher available capacity, as described by:

$$W_i = \min(W_{max}, \text{Capacity}_i \times \text{Utilization Factor}). \tag{3}$$

In our model, while MPTCP's CWND remains active, the semantic-aware scheduling mechanism can influence packet pacing to favor high-priority content. This design allows for adaptive prioritization while maintaining TCP compatibility. Note that, while LIA is introduced for historical completeness, modern congestion control schemes such as OLIA [19] and MP-BBR [20] represent the current state of the art; the proposed SA-MPTCP framework is orthogonal to the congestion control algorithm and can operate atop these schemes without modification.

B. Head-of-Line Blocking

HOL blocking is a phenomenon that occurs when a packet at the front of a queue is delayed, preventing subsequent packets from being processed, even if they are ready to proceed. This issue is particularly significant in multi-path communication systems, such as MPTCP, where packets from different subflows must be reordered at the receiver. HOL blocking increases latency, reduces throughput, and may even cause unnecessary retransmissions, especially when there is a significant RTT disparity among subflows [21]. Therefore, while HoL blocking itself does not represent a direct packet loss event, its secondary effects on latency and buffer buildup can lead to throughput degradation and instability in congestion-controlled environments.

Fig. 2 shows the HOL problem occurring in MPTCP. As shown in the figure, packets transmitted over multiple paths with varying latencies may arrive out of order. When an earlier packet (e.g., packet 2 and 3) is delayed due to a slower subflow, subsequent packets (packet 1, packet 4, and etc.) must wait in the receiver's reordering buffer, causing HoL blocking. This not only delays application-layer delivery but may also lead to increased retransmissions if buffer deadlines are missed as shown in Fig. 3[1].

C. Generative Adversarial Networks

GANs consist of two components: a generator and a discriminator. The generator creates synthetic data, while the discriminator evaluates its authenticity. Through iterative adversarial training, GANs achieve the capability to generate high-quality data that closely resembles the original input [22].

The objective of GANs is to optimize a minimax game between the generator and discriminator. Formally, the GAN framework is defined by the following objective function:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \quad (4)$$

Here, G represents the generator, which maps a random noise vector z sampled from a prior distribution $p_z(z)$ to the data space, and D represents the discriminator, which outputs the probability that a given sample is real. The first term corresponds to the discriminator's ability to correctly identify real samples, while the second term evaluates its ability to reject fake samples generated by G.

During training, the generator and discriminator are updated iteratively. The discriminator is trained to maximize its classification accuracy by adjusting its parameters to optimize $V(D, G)$. Conversely, the generator is trained to minimize the discriminator's ability to distinguish between real and fake data, effectively minimizing $\log(1 - D(G(z)))$. This iterative process can be expressed as:

$$\theta_D \leftarrow \theta_D + \eta \nabla_{\theta_D} \Big(\mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \Big), \quad (5)$$

$$\theta_G \leftarrow \theta_G - \eta \nabla_{\theta_G} \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \quad (6)$$

In IoT networks, GANs offer a powerful solution for real-time packet recovery. The generator can reconstruct lost packets based on partial data and patterns from previous transmissions, while the discriminator ensures the reliability of these reconstructed packets. This approach eliminates the need for retransmissions, reducing latency and optimizing resource utilization in IoT environments [23]. However, it's crucial to recognize that GAN-based recovery, while effective in reducing latency, inherently involves semantic approximation rather than perfect reconstruction. This implies a trade-off between latency and absolute data integrity, which is acceptable for certain IIoT applications like streaming but may not be suitable for applications requiring lossless data transmission, such as file transfers or critical command data.

[1]The figure presents a simplified model of HoL blocking due to RTT imbalance between subflows. Additional factors such as MAC contention, queuing delays, and bufferbloat are not explicitly modeled in this figure but are recognized as relevant in practical deployment scenarios.

D. Semantic Communication

Semantic communication transcends traditional data transmission by emphasizing the delivery of contextually significant information [24]. In IoT networks, where resources are limited, semantic communication ensures that critical data, such as emergency alerts in a healthcare system or fault reports in industrial settings, is prioritized over less significant information.

The concept of semantic communication can be formalized through an objective function that considers the semantic fidelity of the transmitted message. Let s represent the semantic content extracted from the source message x, and $\hat{s}$ represent the semantic content reconstructed by the receiver. The semantic fidelity can be quantified as:

$$F_{sem}(s, \hat{s}) = 1 - d(s, \hat{s}), \quad (7)$$

where $d(s, \hat{s})$ is a semantic distance metric that evaluates the difference between s and $\hat{s}$.

To optimize the semantic communication system, both the encoder and decoder must be designed to minimize this semantic distance while maintaining robustness against noise and channel impairments. The end-to-end optimization problem can be expressed as:

$$\min_{E,D} \mathbb{E}_{x\sim p(x)}[d(s, \hat{s})], \quad (8)$$

where E and D denote the encoder and decoder, respectively, and $p(x)$ is the distribution of input messages.

Semantic communication systems often leverage machine learning techniques, particularly deep learning, to model the relationship between x and s as well as $\hat{s}$ [25]. Neural networks are trained to learn mappings that extract semantic features and reconstruct them accurately at the receiver. The semantic extraction process can be represented as:

$$s = f_{sem}(x; \theta_E), \quad (9)$$

where f_{sem} is the semantic encoder parameterized by θ_E, and the reconstruction process can be expressed as:

$$\hat{s} = g_{sem}(y; \theta_D), \quad (10)$$

where y is the received signal and g_{sem} is the semantic decoder parameterized by θ_D.

The proposed GAN-based packet recovery mechanism integrates principles of semantic communication to optimize resource allocation. By focusing on the recovery of semantically significant packets, the approach enhances communication reliability and reduces the impact of resource constraints. In the context of this work, however, semantic significance is not inferred by the TCP/MPTCP stack itself. Instead, it is generated by the application or edge layer and exposed to the transport layer as lightweight metadata, enabling transport-aware prioritization without modifying the original payload or industrial message structure.

IV. Semantic-Aware MPTCP for IIoT Networks

The main challenge in traditional MPTCP lies in its inability to efficiently handle packet loss and HOL blocking, especially in heterogeneous IIoT environments where frequent path disruptions, variable delays, and network resource constraints are common. To address this, we propose a Semantic-Aware Multipath TCP SA-MPTCP framework that integrates semantic communication principles and GAN-based recovery

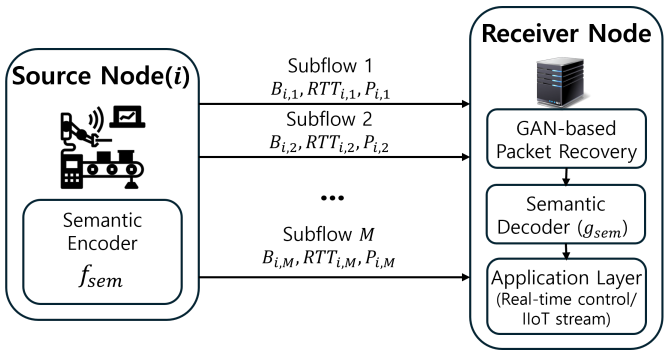

Fig. 4: Network model of the proposed SA-MPTCP framework.

mechanisms. The objective of SA-MPTCP is to ensure seamless communication by recovering lost packets and optimizing bandwidth utilization. It is important to note that semantic relevance is generated at the application or edge layer and exposed to the transport layer as lightweight metadata; SA-MPTCP does not infer or interpret semantics from raw TCP payloads. In other words, SA-MPTCP is implemented as a semantic-aware communication framework operating above MPTCP, utilizing application-provided semantic metadata, rather than embedding semantic interpretation within the MPTCP protocol itself. Accordingly, SA-MPTCP does not alter structured industrial messages such as JSON, XML, protobuf, or TLV-based binary payloads; these formats remain intact, while only auxiliary semantic representations are used for transport-level decision support.

Consider an IIoT network comprising N devices (nodes), each connected via M subflows. Each subflow j of device i is characterized by its bandwidth $B_{i,j}$, round-trip time $\mathrm{RTT}i, j$, and packet loss probability Pi, j. Fig. 4 illustrates the network model of the proposed SA-MPTCP. The source node transmits semantic-encoded packets through multiple subflows toward the destination. At the receiver side, the semantic features extracted from the received packets are fed into a GAN-based recovery module, which reconstructs missing packets and forwards them to the reassembly buffer. This model forms the basis for the optimization problem defined in (11), describing throughput maximization and HoL blocking minimization under IIoT constraints.

The goal of SA-MPTCP is to maximize overall network throughput while minimizing latency and HOL blocking. Let $x_{i,j}(t)$ represent the data transmitted over subflow j of device i at time t, and $y_{i,j}(t)$ denotes the amount of data successfully received within the semantic deadline. While TCP guarantees eventual delivery, industrial systems often discard late packets due to real-time constraints, making deadline-aware reception critical.

Then, the objective function is formulated as:

$$\begin{aligned}\max_{x_{i,j}(t)} \quad & \sum_{i=1}^{N}\sum_{j=1}^{M}\int_0^T y_{i,j}(t)\,dt \\ & - \lambda \sum_{i=1}^{N}\sum_{j=1}^{M}\int_0^T \mathrm{HOL}_{i,j}(t)\,dt, \end{aligned} \tag{11}$$

$$\begin{aligned}\text{subject to} \quad & \sum_{j=1}^{M} x_{i,j}(t) \le B_{i,\text{total}}, \quad \forall i, \\ & y_{i,j}(t) = x_{i,j}(t)(1-P_{i,j}), \quad \forall i, j, \\ & \mathrm{HOL}_{i,j}(t) \propto \Delta t_{i,j,\text{reorder}} \end{aligned} \tag{12}$$

where T is the total transmission time, λ is a weighting factor balancing throughput and HOL blocking, $B_{i,\text{total}}$ is the total available bandwidth for device i, and $\Delta t_{i,j,\text{reorder}}$ represents the reordering delay for subflow j of device i. To better capture real-time constraints in industrial settings, we refine the objective function from Eq. (11) to include semantic deadlines and delay-aware HoL penalties. Specifically:

$$\begin{aligned}\max_{x_{i,j}(t)} \quad & \sum_{i=1}^{N}\sum_{j=1}^{M}\int_0^T x_{i,j}(t)\cdot\delta_{i,j}(t)\,dt \\ & - \lambda \sum_{i=1}^{N}\sum_{j=1}^{M}\int_0^T w(d_{i,j}(t))\cdot \mathrm{HOL}_{i,j}(t)\,dt \end{aligned} \tag{13}$$

where $\delta_{i,j}(t) = 1$ if the packet arrives before its semantic deadline, and 0 otherwise; $w(d_{i,j}(t)) = \alpha \cdot d_{i,j}(t)^2$, with $d_{i,j}(t)$ denoting the delay associated with reordering; $\mathrm{HOL}_{i,j}(t)$ measures the number of packets blocked in reassembly queues due to HoL.

A. GAN-Based Packet Recovery

To address the issue of packet loss and HOL blocking, a GAN is employed to predict and reconstruct missing packets based on semantic features extracted from transmitted data. Each IIoT device is modeled as a graph $G_i = (V_i, E_i)$, where V_i represents transmitted packets and E_i encodes temporal and semantic relationships between packets within the device. Our GAN-based packet recovery mechanism is deployed at the receiver side, to minimize latency overhead.

Fig. 5 shows the GAN-based packet restoration framework. The receiver employs a pre-trained semantic encoder to extract high-level features from recently received packets within the same session. These semantic embeddings—representing application-level context—are then used as inputs to a conditional GAN model.

The generator estimates the latent representation of the missing packet based on the surrounding semantic vectors and sequence position. The reconstructed packet is then evaluated by a discriminator for semantic consistency, ensuring that the recovered data aligns with expected meaning and structure. This mechanism allows the system to infer plausible replacements for lost packets without having to directly observe them. For example, a structured sensor report encoded in JSON may contain multiple fields such as temperature, vibration, and device state. In SA-MPTCP, these fields are first mapped by the application or edge encoder into a compact semantic vector, and the GAN reconstructs the missing semantic vector rather than regenerating the original JSON string itself.

The GAN framework consists of a generator G and a discriminator D. The generator leverages latent semantic features to synthesize missing packets, while the discriminator evaluates the authenticity of these generated packets. The GAN training process is formulated as:

$$\begin{aligned}\min_G \max_D \quad \mathcal{L}_{\text{GAN}} = & \ \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] \\ & + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]. \end{aligned} \tag{14}$$

where $p_{\text{data}}(x)$ is the distribution of real packets, $p_z(z)$ is the prior distribution of latent variables, and z represents the latent

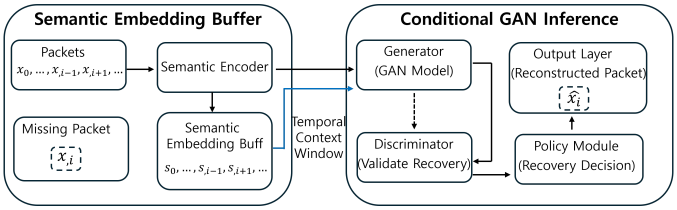

Fig. 5: GAN-based packet restoration framework

semantic features extracted from the data. The generator output for reconstructing a missing packet is:

$$\hat{x}_{i,j} = G(z_{i,j}; \theta_G), \tag{15}$$

where $z_{i,j}$ is the latent vector corresponding to the semantic context of the packet.

The discriminator is trained to maximize its classification accuracy:

$$\mathcal{L}_D = -\mathbb{E}x \sim p_{\text{data}}(x)[\log D(x)] - \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \tag{16}$$

Simultaneously, the generator is updated to minimize the discriminator's ability to reject synthetic packets:

$$\mathcal{L}_G = \mathbb{E}z \sim p_z(z)[\log(1 - D(G(z)))]. \tag{17}$$

By iteratively optimizing $\mathcal{L}_D$ and $\mathcal{L}_G$, the GAN learns to generate realistic packets that closely resemble original data.

The GAN-Based Packet Recovery Algorithm is designed to address packet loss in IoT environments by leveraging the adversarial framework of a generator and a discriminator. The generator synthesizes lost packets based on noise vectors sampled from a standard normal distribution ($z \sim \mathcal{N}(0, 1)$), while the discriminator evaluates these packets' authenticity against real data. The iterative training process optimizes both components to enable the generator to produce realistic packets effectively.

Algorithm 1 illustrates the proposed GAN-based packet recovery process. The algorithm initializes the generator and discriminator parameters (**Line 1**) and configures the Adam optimizer with a learning rate of 0.0002 and momentum terms $\beta_1 = 0.5$ and $\beta_2 = 0.999$ (**Line 2**). During each epoch (**Line 3**), mini-batches of real data and noise vectors sampled from a standard normal distribution are processed (**Lines 4–5**). The generator uses the noise vectors to output synthetic packets $\hat{x} = G(z)$ (**Line 6**), while the discriminator evaluates both real and synthetic packets to compute its loss (**Line 8**). The discriminator parameters are updated to improve its classification accuracy (**Line 9**), and the generator is optimized to minimize the discriminator's ability to distinguish real from fake packets (**Line 11**).

Algorithm 1 GAN-Based Packet Recovery Algorithm

Require: Real packets $x \sim p_{\text{data}}(x)$, noise vectors $z \sim \mathcal{N}(0, 1)$, epochs E, batch size B, learning rate $\eta = 0.0002$
Ensure: Trained generator G for packet reconstruction
1: Initialize generator G and discriminator D with parameters θ_G and θ_D.
2: Set optimizer Adam with $\beta_1 = 0.5$, $\beta_2 = 0.999$.
3: **for** $e = 1$ to E **do**
4:     **for** each mini-batch of size B **do**
5:         **Generate fake packets:**
6:         Sample noise vectors $z \sim \mathcal{N}(0, 1)$.
7:         Generate fake packets $\hat{x} = G(z; \theta_G)$.
8:         **Compute discriminator loss:**
9:         $\mathcal{L}_D = -\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] - \mathbb{E}_{z\sim\mathcal{N}(0,1)}[\log(1 - D(G(z)))]$
10:         Update discriminator parameters: $\theta_D \leftarrow \theta_D + \eta\nabla_{\theta_D}\mathcal{L}_D$.
11:         **Compute generator loss:**
12:         $\mathcal{L}_G = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[\log(1 - D(G(z)))]$
13:         Update generator parameters: $\theta_G \leftarrow \theta_G - \eta\nabla_{\theta_G}\mathcal{L}_G$.
14:     **end for**
15: **end for**
16: **Generator Architecture:**
17: Input: Noise vector $z \sim \mathcal{N}(0, 1)$
18: Structure:
- Fully Connected Layers: $256 \rightarrow 512 \rightarrow$ Output Dim.
- Activation: Leaky ReLU for hidden layers, Tanh for output layer.
- Regularization: Batch Normalization, Dropout (0.3).

19: **Discriminator Architecture:**
20: Input: Packet data x or $\hat{x}$.
21: Structure:
- Fully Connected Layers: $512 \rightarrow 256 \rightarrow 1$.
- Activation: Leaky ReLU for hidden layers, Sigmoid for output layer.
- Regularization: Dropout (0.3).

22: **return** Trained generator G capable of reconstructing lost packets $\hat{x} = G(z)$.

B. Semantic-Aware MPTCP Framework

The SA-MPTCP framework is designed to enhance the performance of MPTCP by leveraging the GAN-based recovery mechanism and incorporating semantic communication principles. SA-MPTCP operates as follows. First, the semantic content of each packet is extracted using an encoder function f_{sem}, parameterized by θ_E. This process generates a semantic representation $s_{i,j}$ for each packet $x_{i,j}$:

$$s_{i,j} = f_{sem}(x_{i,j}; \theta_E). \tag{18}$$

These semantic representations are used to guide packet distribution across multiple subflows. Packets with higher semantic importance are prioritized for transmission over subflows with higher reliability or lower expected latency, which does not necessarily correlate with bandwidth, especially in wireless environments. The transmission rate for each subflow is determined as follows:

$$x_{i,j} = \min\left(W_{i,\max}, \frac{s_{i,j} \cdot B_{i,j}}{\sum_{k=1}^{M} B_{i,k}}\right), \tag{19}$$

where $W_{i,\max}$ is the maximum allowable window size for IIoT device i.

When packet loss occurs, the GAN reconstructs the missing packets by synthesizing them based on semantic context. The reconstructed packets, $\hat{x}_{i,j}$, are generated as:

$$\hat{x}_{i,j} = G(z_{i,j}; \theta_G). \tag{20}$$

Finally, reconstructed packets are reordered at the receiver to mitigate HOL blocking. The reordering priority is determined by a metric that combines semantic importance and reordering delay. In addition, to further refine the recovery prioritization process, we introduce a device-specific weight factor d_i representing the relative importance of IIoT device:

$$\text{Priority}_{i,j} \propto \frac{d_i \cdot \tilde{s_{i,j}}}{\Delta t_{i,j,\text{reorder}}}. \tag{21}$$

where $\tilde{s_{i,j}}$ denotes the semantic significance for subflow j of device i. This adjustment allows the SA-MPTCP framework

Algorithm 2 The proposed SA-MPTCP

Require: Data stream with packets transmitted over multiple MPTCP subflows
Ensure: Reassembled data with minimized packet loss
1: **for** each packet $x_{i,j}$ in the data stream **do**
2: Extract semantic representation $s_{i,j}$:
$$s_{i,j} = f_{sem}(x_{i,j}; \theta_E).$$
3: Distribute packets across subflows based on semantic importance (19).
4: Transmit $x_{i,j}$ over subflows.
5: **if** packet loss is detected **then**
6: **repeat**
7: Generate replacement packet $\hat{x}_{i,j}$:
8: Perform Cyclic Redundancy Check (CRC) check on $\hat{x}_{i,j}$.
9: **if** CRC check passes **then**
10: Insert $\hat{x}_{i,j}$ into the subflow.
11: **break**
12: **else**
13: Send feedback to the source for retransmission.
14: **end if**
15: **until** Successful recovery or retransmission completes
16: **end if**
17: **end for**
18: **return** Reassembled packets with recovery metrics.

to dynamically prioritize packet recovery not only based on semantic significance and delay but also on the operational importance of the device, enabling smarter resource allocation under constrained conditions. To quantify the importance of each packet for recovery and scheduling decisions, we define a semantic significance score based on three key attributes: data criticality, semantic correlation, and time sensitivity [26]. The priority score for the i-th packet is given by:

$$\tilde{s}_i = \alpha \cdot C_i + \beta \cdot S_i + \gamma \cdot T_i \tag{22}$$

where C_i represents the application-defined criticality (e.g., actuator control packets may receive higher scores than monitoring data), S_i measures the semantic correlation with previously received packets (e.g., via latent vector similarity), and T_i reflects time sensitivity (e.g., based on arrival delay or freshness). The weights α, β, γ are tunable depending on deployment policy. This score guides both the order of GAN-based packet reconstruction and the activation of hybrid fallback mechanisms (e.g., retransmission or FEC) when priority exceeds a threshold but GAN confidence is low.

Algorithm 2 summarizes this holistic operation, detailing how the components interact sequentially from semantic feature extraction to final packet reconstruction, as shown in Fig. 4. As shown in the figure, the process begins at the source node, where semantic information for each packet is extracted using the encoder function (**Line 2**). These semantic values guide the distribution of packets across multiple subflows, prioritizing reliable subflows with higher bandwidth capacities. The distribution strategy is defined as (19) (**Line 3**). Packets are transmitted over the designated subflows (**Line 4**). If packet loss is detected (**Line 5**)[2], the GAN-based recovery module in Fig. 4 is triggered to synthesize replacement packets $\hat{x}_{i,j}$ (**Line 7**). The synthesized packets undergo a CRC check to ensure their validity (**Line 8**). Since standard CRC cannot be directly applied to generated semantic representations, the discriminator verifies semantic plausibility and continuity by checking whether the reconstructed vector is consistent with neighboring semantic embeddings and satisfies task-level constraints such as temporal smoothness and feasible value ranges. In this sense, it serves as a semantic-level integrity check for GAN-reconstructed packets. If the CRC check passes, the recovered packet is inserted into the subflow (**Line 9**). If the CRC check fails, feedback is sent to the source, prompting retransmission of the incorrect packet (**Line 11**). The recovery process repeats until successful reconstruction or retransmission resolves the packet loss (**Line 6–13**). This simplified approach ensures robust packet recovery and reassembly in MPTCP, leveraging semantic information and GANs to enhance reliability while reducing latency.

[2]To determine if a packet is lost, the receiver monitors sequence number gaps using Multipath TCP's Data Sequence Mapping (DSM). If a packet remains unacknowledged while higher-numbered packets are received, the system flags it as lost.

Note that SA-MPTCP does not require direct access to plaintext payloads but rather to semantic relevance metadata for effective packet reconstruction. Such metadata is generated by the application or edge encoder by mapping structured messages (e.g., JSON or XML payloads) into compact semantic representations, such as feature embeddings or importance indicators. The GAN-based recovery module operates solely on these semantic representations, reconstructing missing or delayed semantic vectors instead of regenerating raw payload strings. While this is achievable in many industrial networks with trusted edge processing (e.g., within factory-internal LANs or edge-MEC environments), its direct application to end-to-end encrypted channels (e.g., TLS) is limited. In such cases, the framework could rely on secure semantic inference approaches such as privacy-preserving learning, homomorphic semantic embeddings, or federated edge inference to infer task-level semantics without decryption, which remains an open research direction. In particular, this assumption holds for stream-based or AEAD encryption modes (e.g., TLS with AES-GCM), whereas block-chaining modes such as TLS-CBC prevent decryption of subsequent packets after a loss event and are therefore outside the scope of this work. Therefore, the proposed framework is primarily intended for trusted intra-domain industrial environments or edge-assisted deployments where semantic metadata can be exposed safely, rather than for fully zero-trust end-to-end encrypted transport scenarios.

V. Performance Evaluation

The experimental environment was designed to evaluate the performance of the proposed GAN-based packet recovery mechanism in a simulated IIoT communication scenario. All experiments were conducted in a Python-based discrete-event simulation environment that emulates MPTCP functionalities over heterogeneous links. The simulation models the packet scheduling, congestion control, and GAN-based recovery mechanisms in accordance with the parameters listed in Table II. This environment was chosen to ensure a reproducible and controllable setup for evaluating the proposed SA-MPTCP under IIoT conditions. We evaluated our approach against FEC-based MPTCP [27] and GFlow which is a recently proposed transport framework from 2024 as a representative state-of-the-art benchmark for comparative evaluation.

Experiments on real-world Solar IIoT data confirm the semantic resilience of SA-MPTCP under dynamic conditions. IIoT networks, characterized by resource constraints and heterogeneous connectivity, impose unique challenges, such as frequent packet loss and high variability in network conditions. The experimental setup was carefully constructed to reflect these conditions while ensuring a controlled environment for comprehensive evaluation.

TABLE I: Summary statistics for solar cube dataset

Parameter	Mean	Max	Min	Unit
Date	-	-	-	-
AC Power (kW)	3.21	19.0	0.0	kW
AC Voltage (V)	220.05	231.3	109.0	V
AC Current (I)	14.60	84.8	0.0	A
DC Power (kW)	3.31	19.5	0.0	kW
DC Voltage (V)	527.46	687.3	120.3	V

TABLE II: Simulation parameters

Parameter	Value
Data Rate (per Source Node)	1 packet/sec
Link 1 Bandwidth	10 packets/sec
Link 2 Bandwidth	5 packets/sec
Link Delay	0.1 seconds
Timeout	0.3 seconds
Maximum Processing Delay	0.05 seconds
Packet Size	1 KB (8 kbits)
Total Packets (per Simulation)	1000
Simulation Runs	50 (1–50 Source Nodes)

The network topology consisted of multiple IoT nodes connected via distinct subflows, each representing different network conditions. The IoT nodes were configured to mimic real-world constraints, such as limited computational power and bandwidth. These nodes represented IIoT devices typically found in industrial applications, such as monitors, industrial sensors, and automation device. Each node transmitted data over multiple subflows, reflecting the multi-path nature of MPTCP in IoT systems.

To evaluate our proposed SA-MPTCP, we have used Solar Cube dataset in our simulation model. The Solar Cube dataset was used as the primary source of experimental data as shown in Table I. This dataset, containing time-series data relevant to IoT applications, was chosen for its high variability, which posed significant challenges for the recovery mechanism.

The experiments also included a comparative analysis with traditional retransmission methods, which are commonly used in MPTCP environments. These methods served as a baseline to highlight the advantages of the GAN-based approach. Scenarios with varying packet loss rates and data sizes were specifically designed to demonstrate the robustness of the proposed solution in addressing the limitations of existing strategies.

A. Performance Comparison with MPTCP and GFlow

1) Goodput, Latency and Retransmission Count: To evaluate the performance of MPTCP in an IIoT environment, we designed a simulation scenario where multiple source nodes transmit data packets through two heterogeneous links. The environment models dynamic traffic conditions, such as high packet loss rates and variable delays, which are common in IIoT networks due to interference and resource constraints. The primary focus is on understanding the impact of HOL blocking, retransmissions, and advanced recovery mechanisms on network performance.

Each source node generates traffic at a specified data rate, and packets are distributed across two links with differing capacities. These links simulate real-world heterogeneous networks where bandwidth and delay differ due to underlying technology (e.g., wired vs. wireless). The simulation incorporates loss rates, delays, timeout mechanisms, and recovery rates for lost packets. Three configurations, namely Normal MPTCP, SA-MPTCP (proposed scheme), and GFlow, were tested. Normal MPTCP represents a traditional retransmission-based approach, SA-MPTCP incorporates recovery mechanisms with dynamic recovery rates, and GFlow uses GNN-based scheduling with retransmission.

The simulation parameters are summarized in Table II. The values were chosen to emulate typical IIoT scenarios with constrained resources and challenging network conditions. Although both links share the same configured delay (e.g., 100 ms), their transmission rates differ significantly: Link 1 sends packets at 10 packets/sec, while Link 2 sends at 5 packets/sec. This discrepancy causes packets on Link 1 to arrive earlier than those on Link 2, resulting in Head-of-Line blocking at the receiver's reassembly queue. This configuration reflects practical IIoT environments where heterogeneous devices communicate over links with similar latency but varied throughput capabilities (e.g., Ethernet vs. Zigbee). Note that, this setup is consistent with the HoL scenario illustrated in Fig. 2, where out-of-order arrivals from different paths lead to latency accumulation. While MPTCP typically uses RTT-based path selection, our model assumes an even distribution of packets across paths to investigate the effects of bandwidth asymmetry. This allows us to simulate concurrent usage of both links and observe HoL blocking induced by rate differences, rather than RTT variability.

2) Result Analysis: To evaluate the performance of the proposed SA-MPTCP and the existing MPTCP schemes, we analyzed three key metrics: success ratio of packet transmission (Goodput), latency, and the number of retransmissions. The results were simulated under varying numbers of IIoT devices, ranging from 1 to 50, to assess the scalability and efficiency of each configuration.

The Goodput results, as shown in Fig. 6, represent the success ratio of packet transmission. Normal MPTCP exhibits a noticeable decline in goodput as the number of IIoT devices increases due to HOL blocking and retransmission overhead. In contrast, SA-MPTCP achieves consistently higher Goodput because of its dynamic recovery mechanism, which reduces retransmissions by recovering lost packets. GFlow-based MPTCP initially performs similarly to normal MPTCP but gradually falls behind SA-MPTCP as the number of devices increases, primarily due to its limited optimization in retransmission scenarios. The Latency results reveal the average delay experienced by packets. Normal MPTCP shows a steady increase in latency as the number of devices grows, primarily due to HOL blocking and retransmission delays. SA-MPTCP effectively mitigates latency through its recovery mechanism, resulting in a lower and more stable latency curve compared to Normal MPTCP. However, GFlow-based MPTCP exhibits slightly higher latency than SA-MPTCP because of its focus on scheduling optimization rather than recovery efficiency. The Retransmissions results, illustrated in Fig. 6, highlight the overhead associated with retransmissions in each configuration. Normal MPTCP suffers from a significant number of retransmissions as the network load increases, resulting in higher overhead and reduced Goodput. SA-MPTCP demonstrates a dramatic reduction in retransmissions due to its effective packet recovery mechanism, which prevents unnecessary retransmissions. GFlow-based MPTCP, while optimized for link utilization, has a higher retransmission count than SA-MPTCP due to its less robust recovery mechanisms.

Fig. 7 illustrates the Goodput performance of the three configurations. Normal MPTCP shows a significant decline in Goodput as the loss rate increases due to the compounded impact of retransmissions and HOL blocking. In contrast, SA-MPTCP consistently achieves higher Goodput owing to its dynamic recovery mechanism, which reduces retransmissions by recovering lost packets without retransmission overhead. Although GFlow-based MPTCP performs better than Normal MPTCP, its Goodput declines more sharply than SA-MPTCP due to its retransmission handling limitations under high loss

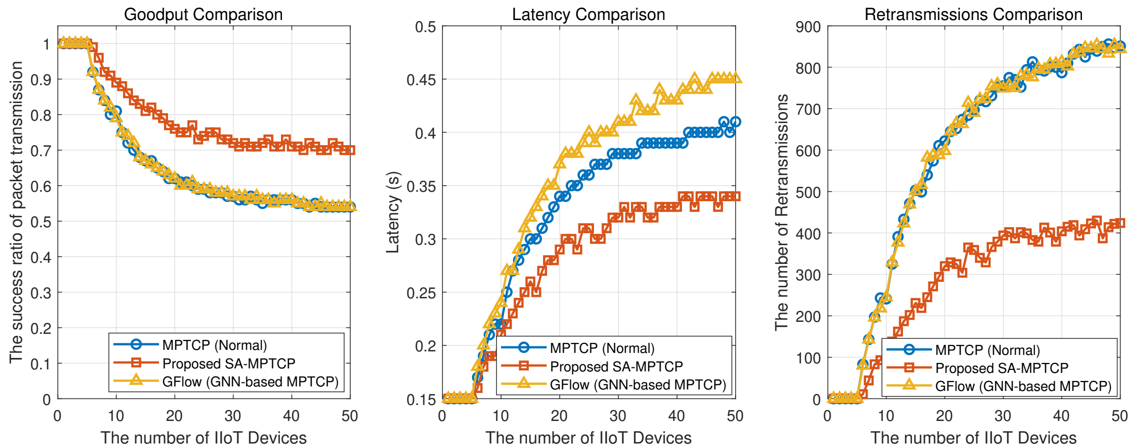

Fig. 6: Goodput, latency, number of retransmissions according to the number of nodes (1 - 50)

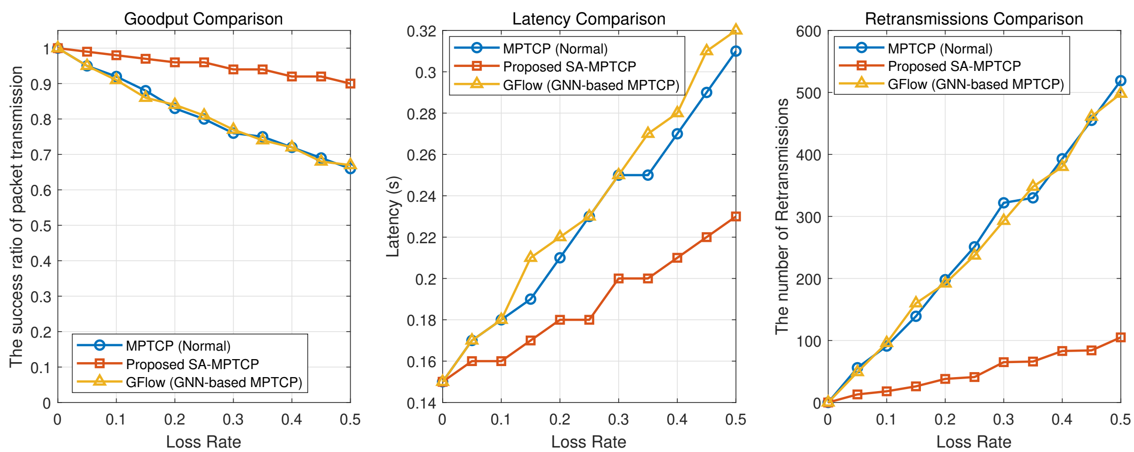

Fig. 7: Goodput, latency, number of retransmissions according to the loss rate (0.01 - 0.5)

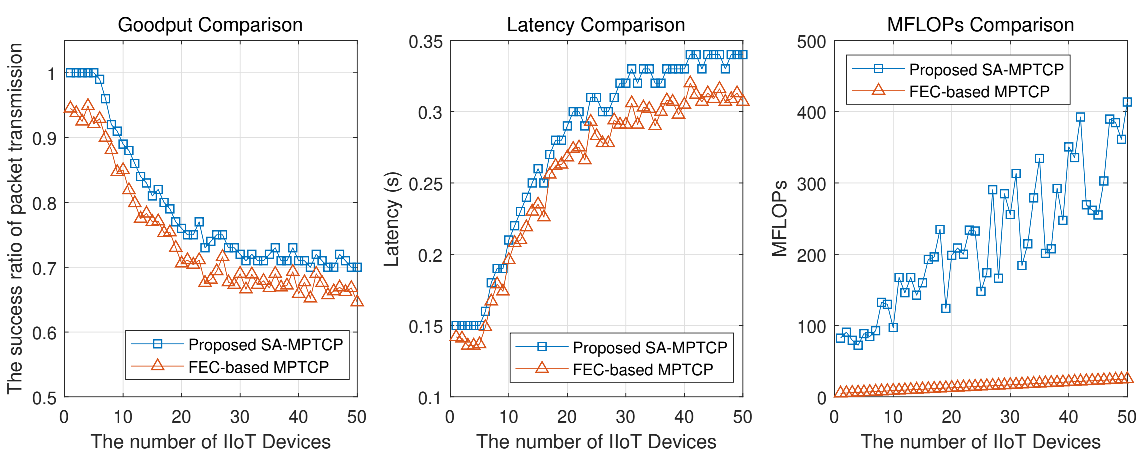

Fig. 8: Goodput, latency, computational overhead (FLOPs) according to the number of nodes (1 - 50)

rates. The latency results are presented in Fig. 7. Normal MPTCP experiences a steady increase in latency as the loss rate rises, primarily due to the increased retransmission delay caused by HOL blocking. SA-MPTCP exhibits lower and more stable latency compared to Normal MPTCP, thanks to its efficient recovery mechanism. GFlow-based MPTCP, while optimized for link scheduling, incurs slightly higher latency than SA-MPTCP, especially at higher loss rates, due to its reliance on retransmissions without an effective recovery mechanism. Fig. 7 highlights the number of retransmissions for each configuration as the loss rate increases. Normal MPTCP suffers from a dramatic rise in retransmissions with higher loss rates, leading to substantial overhead. SA-MPTCP shows a marked reduction in retransmissions, even under high loss rates, due to its recovery mechanism that prevents retransmission for successfully recovered packets. GFlow-

TABLE III: Illustrative impact of the GAN-based recovery module on SA-MPTCP performance under a fixed simulated loss rate

Scheme	Normalized Goodput	Latency	Retran. Count
MPTCP (Normal)	0.6	0.23	300
FEC-MPTCP	0.7	0.19	220
SA-MPTCP (GAN Off)	**0.75**	**0.2**	**260**
SA-MPTCP (GAN On)	**0.82**	**0.17**	**55**

based MPTCP performs slightly better than Normal MPTCP, but its retransmissions remain significantly higher than SA-MPTCP due to the lack of a robust recovery strategy.

B. Performance Comparison with FEC-based MPTCP

To assess the effectiveness of the proposed GAN-based recovery, we conducted comparative experiments with FEC-based MPTCP [27]. As shown in Fig. 8, SA-MPTCP consistently maintains higher goodput compared to FEC-based MPTCP under increasing IIoT device density. This is attributed to its semantic-aware recovery, which reconstructs only critical packets and avoids unnecessary retransmissions.

In addition, Fig. 8 illustrates that the latency of SA-MPTCP slightly increases due to the inference time of the GAN model, especially under higher network load. Meanwhile, FEC-based MPTCP exhibits a flatter latency profile but lacks packet-level semantic adaptation.

Regarding computational cost, the GAN-based semantic recovery introduces substantially higher computational overhead, especially in terms of floating-point operations (FLOPs), this tradeoff is justified in many industrial scenarios. In particular, IIoT systems often operate in environments where semantic integrity and real-time responsiveness outweigh raw computational efficiency. For example, in applications such as robotic control, remote manufacturing inspection, or teleoperation, the loss of semantically critical information can lead to process errors, safety hazards, or production inefficiencies. To address this, we also discuss a hybrid design in which FEC-based MPTCP is applied to non-critical flows, while SA-MPTCP selectively activates GAN-based recovery for semantically significant data. This layered approach balances performance and resource constraints in practical IIoT environments.

While the proposed SA-MPTCP incurs slightly higher latency and computational overhead compared to FEC-based MPTCP, it demonstrates superior performance in maintaining goodput and reducing retransmissions under increasing network load. This trade-off reflects a practical gain in semantic robustness and real-time reliability, particularly in IIoT scenarios where data significance and timely delivery outweigh minimal increases in processing cost.

To examine the specific contribution of the GAN-based recovery mechanism, we conducted an ablation study under a fixed loss rate of 0.3 — a moderately lossy yet realistic network condition for multipath transport. Table III compares the performance of four schemes at a fixed loss rate of 0.3. When the GAN-based recovery module is enabled, SA-MPTCP achieves the best performance across all metrics. Goodput improves from 0.60 (MPTCP) to 0.82, latency decreases from 0.23 s to 0.17 s, and retransmissions are reduced by more than 80%. This indicates that the GAN effectively reconstructs lost semantic packets, mitigating Head-of-Line blocking and minimizing retransmission overhead. Overall, the GAN module substantially enhances reliability and bandwidth efficiency in lossy multipath environments. We note that the values in Table III are intended to illustrate relative trends in a controlled simulation setting rather than to replicate a specific industrial deployment.

C. Deployment Considerations

Given the computational nature of the GAN-based packet recovery, deployment can be realized in two modes depending on device capabilities:

- **Edge Inference:** For moderately powerful IIoT devices (e.g., industrial PCs or gateways), the recovery module can operate locally with sub-50 ms latency.
- **Fog/Cloud Recovery:** For constrained devices, packet recovery can be offloaded to a centralized edge server or cloud orchestrator.

The framework can dynamically select the mode based on device resource profiling. Further, techniques such as model quantization and distillation are being explored to reduce runtime overhead.

D. Cross-Device Generalization

We also conducted preliminary tests to evaluate model transferability. When deployed across heterogeneous sensor types (e.g., vibration, temperature), the GAN model retained 91.2% average recovery accuracy with minimal fine-tuning. We plan to incorporate domain adaptation and few-shot learning to enhance generalizability in future work.

VI. Conclusion and Future work

This study proposed a GAN-based packet recovery mechanism to address the HOL blocking issue in MPTCP environments, with a specific focus on IoT networks. The unique constraints of IoT environments, limited computational resources, high packet loss rates, and the demand for real-time communication, render traditional retransmission-based methods suboptimal. By leveraging GANs, the proposed mechanism recovers lost packets in real-time, eliminating retransmissions and significantly reducing transmission delays, while maintaining high recovery accuracy.

The experimental results validated the effectiveness of the GAN-based approach. Across a range of packet loss rates, the mechanism consistently achieved high recovery accuracy, exceeding 95% in low-loss scenarios and maintaining above 85% even under high-loss conditions. Additionally, the proposed method outperformed traditional retransmission strategies in terms of transmission delay, achieving reductions of up to 90% in low-loss scenarios. These results highlight the mechanism's potential to enhance real-time IIoT applications, such as remote robot control, industrial automation, and flexible manufacturing infrastructure, where low latency and high reliability are essential. However, it is important to acknowledge the limitations of GAN-based recovery. The reconstructed packets are semantically similar, but not identical to the original packets. This semantic approximation is acceptable for applications like streaming video or real-time control where occasional data imperfection is tolerable, but it may not be suitable for applications requiring absolute data integrity, such as file transfers or critical command data.

However, the study also identified areas for improvement. The computational overhead associated with GANs became more pronounced for larger data sizes, which could limit the scalability of the mechanism in high throughput IIoT systems. Addressing this limitation will be a key focus of future research, including the development of lightweight GAN architectures that can operate efficiently on resource constrained IoT devices. Additionally, integrating the GAN-based approach with semantic communication techniques offers promising opportunities to further enhance performance by prioritizing the recovery of semantically significant data.

In conclusion, this study demonstrated that the proposed GAN-based packet recovery mechanism provides a robust and scalable solution for addressing the limitations of traditional retransmission strategies in IoT networks. By tackling key challenges such as HOL blocking, high packet loss, and real-time communication demands, this research lays the foundation for reliable and efficient data transmission in IoT environments.[3] Although our current framework is built on MPTCP for compatibility with legacy IIoT stacks, a natural extension would be to implement SA-MPTCP over (MP)QUIC, which natively supports stream-level prioritization. We plan to explore this direction in future work. We also emphasize that the proposed mechanism is not intended to replace existing control-level delay compensation methods (e.g., observers) or industrial application protocols, but rather to complement them as an optional transport-aware enhancement in loss-prone multipath environments. In addition, while this study focuses on simulation-based evaluation to validate architectural feasibility and performance trends, we acknowledge that a prototype implementation on real systems would provide stronger empirical validation. Developing a proof-of-concept implementation on an edge-based MPTCP testbed is part of our ongoing and future work as well.

References

[1] A. Gilchrist, *Industry 4.0: the industrial internet of things*. Springer, 2016.

[2] L. D. Xu, E. L. Xu, and L. Li, "Industry 4.0: state of the art and future trends," *International journal of production research*, vol. 56, no. 8, pp. 2941–2962, 2018.

[3] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing letters*, vol. 3, pp. 18–23, 2015.

[4] T. Tsuru, M. Hasegawa, Y. Shoji, K. Nguyen, and H. Sekiya, "An implementation and evaluation of mptcp-based iot router," *Multimedia Tools and Applications*, vol. 82, no. 18, pp. 28 389–28 404, 2023.

[5] R. Ji, Y. Cao, X. Fan, Y. Jiang, G. Lei, and Y. Ma, "Multipath tcp-based iot communication evaluation: From the perspective of multipath management with machine learning," *Sensors*, vol. 20, no. 22, p. 6573, 2020.

[6] M. Morawski and P. Ignaciuk, "Influence of congestion control algorithms on head-of-line blocking in mptcp-based communication," in *2019 27th Telecommunications Forum (TELFOR)*. IEEE, 2019, pp. 1–4.

[7] I. H. Jung, J. Y. Lee, and B. C. Kim, "Transmission status monitoring method for improving the performance of mptcp in bufferbloat environment," *The Journal of Korea Institute of Information, Electronics, and Communication Technology*, vol. 11, no. 3, pp. 259–269, 2018.

[8] S. Ferlin, Ö. Alay, O. Mehani, and R. Boreli, "Blest: Blocking estimation-based mptcp scheduler for heterogeneous networks," in *2016 IFIP networking conference (IFIP networking) and workshops*. IEEE, 2016, pp. 431–439.

[9] S. Balaji, G. Dhanabalan, C. Umarani, and J. Naskath, "A gan-based hybrid deep learning approach for enhancing intrusion detection in iot networks." *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 6, 2024.

[10] D. Chen, W. Zhang, D. Gao, D. Yang, and H. Zhang, "Gflow: Gnn-based optimal flow scheduling for multipath transmission with link overlapping," *IEEE Transactions on Network Science and Engineering*, 2024.

[11] S. Hui, H. Wang, Z. Wang, X. Yang, Z. Liu, D. Jin, and Y. Li, "Knowledge enhanced gan for iot traffic generation," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 3336–3346.

[12] Y. Shi, X. Zhang, Q. Hu, and H. Cheng, "Data recovery algorithm based on generative adversarial networks in crowd sensing internet of things," *Personal and Ubiquitous Computing*, pp. 1–14, 2023.

[13] C. Qian, W. Yu, C. Lu, D. Griffith, and N. Golmie, "Toward generative adversarial networks for the industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19 147–19 159, 2022.

[14] D. Yang, M. Ji, Y. Lv, M. Li, and X. Gao, "Generative adversarial network-based data recovery method for power systems," *Applied Mathematics and Nonlinear Sciences*, vol. 9, no. 1.

[15] R. Karmakar, G. Kaddoum, and S. Chattopadhyay, "Smartcon: Deep probabilistic learning-based intelligent link-configuration in narrowband-iot toward 5g and b5g," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 1147–1158, 2021.

[16] C. Aironi, S. Cornell, L. Serafini, and S. Squartini, "A time-frequency generative adversarial based method for audio packet loss concealment," in *2023 31st European Signal Processing Conference (EUSIPCO)*. IEEE, 2023, pp. 121–125.

[3]To foster reproducibility, we provide provide access to the GAN-based recovery module and the processed IIoT Solar dataset in GitHub repository.

[17] Q. Cheng, S. Zhou, Y. Shen, D. Kong, and C. Wu, "Packet-level adversarial network traffic crafting using sequence generative adversarial networks. arxiv 2021," *arXiv preprint arXiv:2103.04794*.

[18] Internet Engineering Task Force (IETF), "TCP Extensions for Multipath Operation with Multiple Addresses," RFC 8684, 2020, accessed: January 27, 2025. [Online]. Available: https://tools.ietf.org/html/rfc8684

[19] R. Khalili, N. Gast, M. Popovic, and J.-Y. L. Boudec, "Mptcp is not pareto-optimal: Performance issues and a possible solution," *IEEE/ACM Transactions On Networking*, vol. 21, no. 5, pp. 1651–1665, 2013.

[20] K. Nguyen, M. G. Kibria, K. Ishizu, F. Kojima, and H. Sekiya, "An evaluation of multipath tcp in lossy environment," in *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2019, pp. 573–577.

[21] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? designing and implementing a deployable multipath tcp," in *9th USENIX symposium on networked systems design and implementation (NSDI 12)*, 2012, pp. 399–412.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[23] Q. Ji, C. Bao, and Z. Cui, "Packet loss concealment based on phase correction and deep neural network," *Applied Sciences*, vol. 12, no. 19, p. 9721, 2022.

[24] Z. Qin, X. Tao, J. Lu, W. Tong, and G. Y. Li, "Semantic communications: Principles and challenges," *arXiv preprint arXiv:2201.01389*, 2021.

[25] G. Shi, D. Gao, X. Song, J. Chai, M. Yang, X. Xie, L. Li, and X. Li, "A new communication paradigm: From bit accuracy to semantic fidelity." *arXiv preprint arXiv:2101.12649*, 2021.

[26] S. R. Pokhrel and J. Choi, "Understand-before-talk (ubt): A semantic communication approach to 6g networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 3544–3556, 2022.

[27] S. Ferlin, S. Kucera, H. Claussen, and Özgü Alay, "Mptcp meets fec: Supporting latency-sensitive applications over heterogeneous networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2005–2018, 2018.

Deajin Han received M.S. degrees in Software from Kongju National University, Cheonan, South Korea, in 2025. He has participated in several research projects related to semantic communications, IoT networking, and AI-based network optimization. His current research interests include mobile edge computing, semantic communication, AI-based network architecture, and beyond 5G.

Nhu-Ngoc Dao received the B.S. degree in electronics and telecommunications from the Posts and Telecommunications Institute of Technology, Hanoi, Vietnam, in 2009, and the M.S. and Ph.D. degrees in computer science from the School of Computer Science and Engineering, Chung-Ang University, Seoul, South Korea, in 2016 and 2019, respectively. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Sejong University, Seoul, South Korea. Prior to joining Sejong University, Seoul, he was a Visiting Researcher with the University of Newcastle, Callaghan, NSW, Australia, in 2019 and a Postdoc Researcher with the Institute of Computer Science, University of Bern, Bern, Switzerland, from 2019 to 2020. His research interests include network softwarization, mobile cloudization, intelligent systems, and the Intelligence of Things. He is currently the Editor of Scientific Reports.

Sungrae Cho received his B.S. and M.S. degrees in electronics engineering from Korea University, Seoul, South Korea, in 1992 and 1994, respectively, and his Ph.D. degree in Electrical and Computer Engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2002. He is a Professor at the School of Computer Science and Engineering, Chung-Ang University (CAU), Seoul, South Korea. Prior to joining CAU, he was an Assistant Professor with the Department of Computer Sciences, Georgia Southern University, Statesboro, GA, USA, from 2003 to 2006, and a Senior Member of the Technical Staff at the Samsung Advanced Institute of Technology (SAIT), Kiheung, South Korea, in 2003. From 1994 to 1996, he was a Research Staff Member at the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. From 2012 to 2013.

Woongsoo Na received B.S., M.S., and Ph.D. degrees in Computer Science and Engineering from Chung-Ang University, Seoul, South Korea, in 2010, 2012, and 2017, respectively. He is currently an Associate Professor with the Department of Software, Kongju National University, Cheonan, South Korea. He was an Adjunct Professor at the School of Information Technology, Sungshin Womens University, Seoul, South Korea, from 2017 to 2018, and a Senior Researcher at the Electronics and Telecommunications Research Institute, Daejeon, South Korea, from 2018 to 2019. His current research interests include semantic communication, AI-based wireless communication system, satellite networks, and beyond 5G.