EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

Time-Sensitive Service Scheduling Method for LEO Broadband Satellite-Based Power Emergency Networks

Xing Yang[1, *], Linhui Yang[2, a]

[1]Science and Technology Project of State Grid Qinghai Electric Power Company, Baoding, Hebei, 071000, China
[2]Science and Technology Project of State Grid Qinghai Electric Power Company, Changzhi, Shanxi, 046000, China

Abstract

INTRODUCTION: The frequent occurrence of extreme natural disasters poses severe threats to power grids, necessitating robust emergency communication. Low Earth Orbit (LEO) broadband satellite networks offer a promising solution but suffer from high dynamics and bit error rates, leading to jitter and disorder in time-sensitive services.
OBJECTIVES: To address these challenges, this paper proposes a time-sensitive service scheduling method for LEO satellites in power emergency networks. The primary goal is to facilitate a transition from "traffic-driven" to "service-driven" paradigms, ensuring the prioritized and orderly delivery of critical control instructions.
METHODS: An improved QUIC scheduling framework is introduced, incorporating the Acknowledgment Fast Return and Distinguish Loss (AFR-DL) mechanism to optimize congestion control. Furthermore, the Multiplex Data Arrive In Order (MDAIO) algorithm is proposed, which combines throughput prediction with dynamic error correction and service priority strategies to manage multipath transmission.
RESULTS: Simulation results demonstrate that the AFR-DL algorithm improves throughput compared to OLIA, while the MDAIO algorithm significantly outperforms traditional schedulers like Lowest-RTT and DPSAF. The proposed method effectively reduces data disorder and transmission delay for high-priority services.
CONCLUSION: The proposed method significantly enhances throughput and reliability, effectively meeting the high real-time requirements of power emergency communication. This work provides a viable solution for guaranteeing the stability of time-sensitive services in harsh LEO satellite environments.

Keywords: LEO broadband satellite, power emergency communication, time-sensitive service scheduling

Received on 01 September 2025, accepted on 21 December 2025, published on 28 April 2026

Copyright © 2026 Xing Yang *et al.*, licensed to EAI. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/ew.12735

1. Introduction

As extreme natural disasters induced by global climate change become increasingly frequent, the security of power grids faces severe challenges. In scenarios such as earthquakes and typhoons, terrestrial power private networks and public communication infrastructure are prone to damage, leading to the formation of "information islands"[1]. This severely constrains the real-time capabilities of relay protection information backhaul, emergency command scheduling, and on-site panoramic situational awareness. As a crucial component of next-generation wireless communication technology (6G), Low Earth Orbit (LEO) broadband satellite constellations have emerged as key infrastructure for reconstructing power emergency communication systems and ensuring the smooth operation of the grid's "lifeline," owing to their geographically unrestricted global coverage, low transmission latency comparable to optical fibers, and rapid deployment capabilities[2][3]. In particular, the construction of mega-constellations such as SpaceX, OneWeb, and China Satellite Network has provided robust physical layer support for building a ubiquitous power Internet of Things integrating space, air, and ground[4][8].

*yangx332@163.com, [a]yanglinhui1988@163.com

However, applying LEO satellite networks to power emergency scenarios is not trivial, as their unique network characteristics pose rigorous challenges to data transmission. First, LEO satellite constellations exhibit a highly dynamic topology; the high-speed motion of satellites relative to the ground results in extremely short connection times between users and a single satellite (typically around 10 minutes), while frequent inter-satellite link handovers and beam switching induce drastic fluctuations in network parameters[5]. Second, the inherent openness of satellite channels exposes them to high bit error rate (BER) challenges, which, combined with the latency accumulation from long-distance transmission, constitutes typical "Long Fat Network" (LFN) characteristics[6][9]. More critically, power emergency services exhibit significant heterogeneity and differences in time sensitivity: on one hand, high-definition panoramic video backhaul from the site requires high throughput support; on the other hand, critical control services such as relay protection differential signals and distribution network automation remote control commands are extremely sensitive to delay jitter and packet loss, demanding millisecond-level deterministic transmission[7]. Coordinating the concurrent transmission of multi-class services and guaranteeing the real-time performance of time-sensitive services in a dynamic and heterogeneous LEO network environment remains an urgent problem to be solved.

Existing satellite transmission and scheduling strategies are predominantly based on traditional TCP/IP protocol stacks or general-purpose multipath protocols, such as Multipath TCP (MPTCP) and Multipath QUIC (MPQUIC). Among them, the traditional TCP protocol incurs huge latency overhead in long satellite links due to its three-way handshake mechanism, and its congestion control mechanism based on packet loss feedback is prone to misjudgment in high BER environments, leading to unnecessary throughput degradation[10]. More severely, TCP's strict in-order delivery mechanism tends to trigger severe Head-of-Line (HOL) blocking during link handovers or packet loss retransmissions, failing to meet the rigorous real-time requirements of time-sensitive power services. Although MPTCP improves robustness by aggregating bandwidth across multiple links, issues such as middlebox ossification and the complexity of kernel implementation limit its deployment on emergency terminals that require high flexibility[11].

As an alternative to TCP, the QUIC (Quick UDP Internet Connection) protocol proposed by Google provides new opportunities for satellite network transmission through features such as user-space implementation based on UDP, 0-RTT rapid connection establishment, connection migration, and stream multiplexing[12]. MPQUIC, developed based on this, further combines the advantages of multipath transmission and QUIC, theoretically making it more suitable for LEO satellite scenarios[15]. However, existing MPQUIC research mostly focuses on the heterogeneous synergy of terrestrial Wi-Fi and cellular networks, and directly transplanting it to power emergency satellite networks faces two major technical bottlenecks.

First, the adaptability of congestion control algorithms in satellite environments is insufficient. The OLIA algorithm, adopted by default in MPQUIC, aims to ensure multipath fairness but lacks the ability to distinguish packet loss types[13]. At power emergency sites, severe weather often leads to non-congestion random bit error packet loss, which OLIA tends to misjudge as network congestion, blindly halving the window and severely constraining the utilization of scarce satellite bandwidth. Second, existing data scheduling strategies lack deep awareness of power service QoS. Mainstream scheduling algorithms like Low-RTT and STMS are primarily designed for broadband services such as web browsing, using latency as the sole scheduling metric[14]. In heterogeneous paths with significant differences, this easily leads to a large number of out-of-order data packets arriving at the receiver, causing receive buffer overflow and reassembly delays. When facing bursty high-concurrency traffic, these algorithms often fail to distinguish between video streams and critical control instructions, resulting in strongly time-sensitive services—such as relay protection differential signals or remote open/close commands—being blocked by non-critical traffic, violating the "safety first" scheduling principle of power services.

To address the aforementioned challenges, this paper is dedicated to researching a time-sensitive service scheduling method for LEO satellites in power emergency networks, aiming to break the limitations of traditional protocols that "prioritize throughput over timeliness" and establish a new transmission architecture that is "service-aware and priority-first." By introducing an improved QUIC protocol architecture, this paper first proposes a congestion control mechanism based on Acknowledgment Fast Return and Distinguish Loss (AFR-DL) targeting the long latency and high error characteristics of LEO links. This mechanism utilizes an ACK path decoupling strategy to accelerate the feedback loop and introduces a joint decision model based on delay gradients and bandwidth variations to accurately identify random loss and congestion loss, thereby maximizing the throughput of emergency links while ensuring fairness. On this basis, targeting the heterogeneous requirements of power services, a smart scheduling strategy based on Multiplex Data Arrive In Order (MDAIO) is designed. This strategy realizes in-order pre-allocation of data packets at the sender by constructing a forward throughput prediction model incorporating retransmission mechanisms, combined with a reverse ACK error correction mechanism; simultaneously, it introduces a service priority awareness mechanism to establish the absolute priority of time-sensitive control services.

2. Congestion Control Algorithm Based on Acknowledgment Fast Return and Distinguish Loss (AFR-DL)

At power emergency repair sites, Low Earth Orbit (LEO) broadband satellites serve as a critical access method for the "space-air-ground integrated" ubiquitous power Internet of Things (IoT), bearing the responsibility of backhauling high-

definition on-site video, relay protection differential signals, and dispatching control commands. However, the power emergency communication environment is extremely complex. On one hand, the high-speed motion of LEO satellites leads to frequent fluctuations in satellite-ground and inter-satellite link latency, and multi-hop transmission via Inter-Satellite Links (ISLs) introduces significant cumulative latency. On the other hand, disastrous weather conditions such as typhoons and rainstorms cause severe channel rain attenuation, resulting in high random bit error rates.

The existing MPQUIC protocol adopts the OLIA congestion control algorithm by default. However, this algorithm exhibits significant defects in the aforementioned scenarios: first, the growth of the Congestion Window (CWND) on long-latency links is sluggish, failing to meet the instantaneous throughput demands of high-bandwidth time-sensitive services such as panoramic video; second, it is unable to distinguish between congestion loss and random error loss, where blind window reduction leads to low link utilization. To address these issues, this section proposes an improved algorithm based on Acknowledgment Fast Return and Distinguish Loss (AFR-DL), aiming to enhance the transmission efficiency of power emergency networks.

2.1. Analysis of Limitations of the OLIA Algorithm in Power Emergency Scenarios

The OLIA (Opportunistic Linked Increases Algorithm), adopted by default in MPQUIC, aims to balance loads across multiple paths. Its window growth strategy during the congestion avoidance phase is shown in Equation (1), and its window reduction strategy upon detecting packet loss is shown in Equation (2):

$$w_r' = \frac{\frac{w_r}{RTT_r^2}}{\left(\sum_{p\in R}\frac{w_p}{RTT_p^2}\right)^2} + \frac{\alpha_r}{w_r} \quad (1)$$

$$w_r' = \frac{w_r}{2} \quad (2)$$

Considering the characteristics of power emergency networks, the above mechanisms present two fatal defects:

- **Lag in Window Growth on Long-Latency Links:** In LEO satellite networks, certain paths may traverse multiple Inter-Satellite Links (ISLs), causing a significant increase in Round-Trip Time (RTT). As indicated by Equation (1), the window growth rate is proportional to $\frac{1}{RTT_r^2}$. This implies that on long-latency paths, the window growth of OLIA is extremely slow. For power emergency video streams requiring the rapid establishment of high-bandwidth transmission channels, this "sluggish" characteristic is unacceptable.
- **Throughput Collapse Due to Random Loss Misjudgment:** Equation (2) indicates that OLIA employs a loss-based feedback mechanism, halving the window immediately upon detecting packet loss. However, at power emergency sites, packet loss rates as high as 5% are often caused by signal attenuation (random errors) under severe weather conditions rather than network congestion. Indiscriminate window reduction leads to the extreme waste of already scarce satellite bandwidth resources, severely threatening the continuity of strongly time-sensitive services such as relay protection.

Therefore, targeting the long latency and high error characteristics of LEO satellite networks, specific optimizations must be made to the acknowledgment mechanism and congestion decision logic.

2.2. ACK Fast Return Based on Best Reverse Path

To address the problem of slow window growth caused by long delays in heterogeneous LEO links, this paper proposes an ACK path decoupling strategy. In the traditional MPQUIC protocol, data packets and acknowledgment (ACK) packets are bound to transmit on the same path. This causes the ACK return on the slow path to take too long, delaying the sender's control loop.

The AFR-DL algorithm breaks the path binding restriction by monitoring the status of all available satellite links in real-time and prioritizing the "fast path" (usually a LEO satellite direct link) with the minimum current Round-Trip Time (RTT) for ACK return.

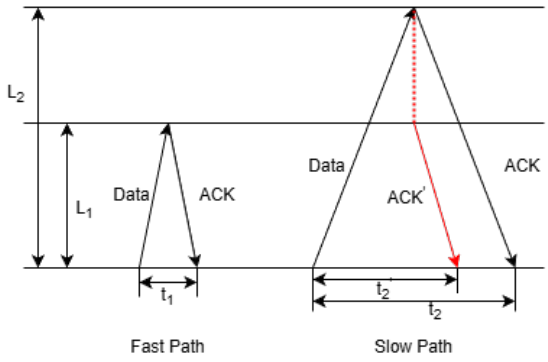

Figure 1. ACK Fast Return

Figure 1 illustrates the specific timing logic of ACK fast return. Assume that a power emergency terminal accesses the satellite network via two links: a "slow path" via multi-hop inter-satellite links and a "fast path" via LEO direct connection. Let the lengths of the fast and slow paths be L_1 and L_2, and their round-trip times be t_1 and t_2, respectively.

In the traditional mode, after a data packet arrives at the receiver via the slow path, the ACK must return via the slow path, resulting in a total closed-loop time of t_2. In the AFR-

DL mode, upon receiving data from the slow path, the receiver uses the fast path to return the ACK. In this case, the total acknowledgment time t_2' for this data packet can be expressed as the sum of the one-way transmission time of the slow path and the one-way transmission time of the fast path:

$$t_2' = (t_1 + t_2)/2 \quad (3)$$

Compared to the original transmission mode, the perceived round-trip time for the slow path is reduced by:

$$\Delta t = t_2 - t_2' = (t_2 - t_1)/2 \quad (4)$$

Since $t_2 > t_1$, it follows that $\Delta t > 0$. This means that by returning the ACK via the fast path, the sender can receive confirmation information Δt time earlier, thereby sliding the sending window faster and significantly increasing the growth rate of the congestion window on the slow path.

Since the ACK takes a "shortcut," the traditional RTT measurement method becomes inaccurate. To accurately measure the true physical state of each heterogeneous path, the algorithm extends a timestamp field in the MPQUIC packet header. The receiver parses this field and uses the difference between the reception time and transmission time to estimate the one-way delay of the path in real-time:

$$RTT_r \approx (T_{recv} - T_{send}) \cdot 2 \quad (5)$$

where RTT_r is the estimated Round-Trip Time of path r; T_{recv} is the data reception time; and T_{send} is the transmission time.

Due to the switch of ACK return to the "fast path," the change in the physical circuit results in the RTT measured at the sender being smaller than the true RTT of the original path, which may mislead the congestion control algorithm. Therefore, it is necessary to correct the RTT of the original path to restore its true physical link state. The corrected RTT for each path is calculated as shown in the formula:

$$RTT_r = \begin{cases} RTT_{min}, & RTT_r = RTT_{min} \\ (RTT_r - RTT_{min}/2) \cdot 2, & RTT_r > RTT_{min} \end{cases} \quad (6)$$

where RTT_r is the Round-Trip Time of path r, and RTT_{min} is the Round-Trip Time of the fastest path.

Through the above mechanism, the algorithm maintains the accuracy of link state perception while significantly shortening the ACK feedback loop, achieving rapid growth of the congestion window for slow paths in power emergency networks. The process of this algorithm is shown in Table 1, and its time complexity is O(n).

Table 1. ACK Fast Return Based on Best Reverse Path

Algorithm 1: ACK Fast Return Based on Best Reverse Path
01. **for** each path **then**
02. $rtt = (curTimeStamp - path[i].timeStamp) \cdot 2$
03. **if** $min\,RTT > rtt$ **then**
04. $min\,RTT = rtt$
05. $min\,RttPathID = pathId$
06. **end if**
07. **end for**
08. **for** each path **then**
09. $ackPath = min\,RttPathID$
10. $sendAck(ackPath)$ // Return ACK via the path with minimum RTT
11. **end for**

2.3. Congestion Adjustment Strategy Based on Distinguish Loss

At power emergency repair sites, random packet losses caused by severe weather are often misjudged as congestion by traditional algorithms, leading to blind window halving. To address this, this section draws on the BBR algorithm idea to construct a joint decision mechanism based on delay and bandwidth variations. This mechanism utilizes the physical characteristic that real congestion is inevitably accompanied by buffer accumulation (RTT increase) and bandwidth limitation. It accurately identifies packet loss scenarios where the physical link capacity is undiminished and RTT does not increase abnormally as random packet losses, thereby implementing differentiated window adjustment strategies to avoid unnecessary throughput degradation.

Based on this logic, the condition for determining network congestion is set as:

$$\begin{cases} RTT_i > \overline{RTT} \\ deliveryRate_i < deliveryRate_{i-1} \end{cases} \quad (7)$$

where RTT_i is the Round-Trip Time of the i-th probe; $\overline{RTT}$ is the average Round-Trip Time; and $deliveryRate_i$ is the bandwidth of the i-th probe.

The average Round-Trip Time formula is:

$$\overline{RTT_r} = \frac{\sum RTT_{r,i}}{i} \quad (8)$$

The bandwidth formula is:

$$deliveryRate_{r,i} = \frac{ACK_{r,k} - ACK_{r,i-1}}{RTT_{r,i}} \quad (9)$$

where $ACK_{r,i-1}$ is the acknowledgment number received for the (i-1) -th data transmission, and $ACK_{r,i}$ is the acknowledgment number received for the i-th transmission. The transmission rate of the i-th transmission is used as the current probed bandwidth.

The pseudocode for the RTT and bandwidth probing algorithm is shown in Table 2. The time complexity of this algorithm is O(1).

Table 2. Flow of RTT and Bandwidth Probing Algorithm

Algorithm 2: RTT Probing Algorithm
01. **for each** $path$ **do**
02. $rtt = getPathRtt(path)$ //Get current path RTT
03. $lastAverageRtt = getLastAverageRtt(path)$ //Get previous average RTT
04. $averageRtt = (lastAverageRtt \times (k-1) + rtt)/k$ //Update average RTT
05. $ack = getPathAckNum(path[i])$ //Get current ACK
06. $lastAck = getPathLastAckNum(path[i])$ //Get previous ACK
07. $deliveryRate = (ack - lastAck)/rtt$
08. **end for**

Based on the above precise decision, the improved AFR-DL algorithm retains the fairness growth strategy of OLIA but executes window halving only when congestion packet loss is confirmed. For random bit error packet losses, the window remains unchanged. Thus, it maximizes the throughput of emergency communication links while ensuring inter-flow fairness. The improved congestion control strategy is as follows:

The congestion window growth is:

$$w_r = \frac{w_r/{RTT_r}^2}{\left(\sum_{p\in R} w_p/{RTT_p}^2\right)^2} + \frac{\alpha_r}{w_r} \quad (10)$$

Furthermore, in scenarios with concurrent access by multiple terminals, maintaining inter-flow fairness is crucial. The AFR-DL algorithm retains the window growth strategy of OLIA, which ensures that when multiple paths compete for bottleneck bandwidth, the algorithm can shift traffic from congested paths to idle paths through the coupling factor α_r, thereby achieving load balancing and fair resource allocation among multiple users.

The window adjustment strategy is:

$$w_r = \begin{cases} w_r/2, & if\ congestion\ decision\ holds \\ w_r, & if\ congestion\ decision\ does\ not\ hold \end{cases} \quad (11)$$

This strategy maximizes the throughput of emergency communication links in harsh environments while guaranteeing inter-flow fairness, ensuring that critical video streams are not interrupted by rain attenuation.

The time complexity of this algorithm is O(1). The flow of the congestion window reduction algorithm based on distinguish loss is shown in Table 3.

Table 3. Flow of Congestion Window Reduction Algorithm

Algorithm 3: Congestion Window Reduction Algorithm
01. **for each** $path$ **do**
02. $averageRtt = getAverageRtt(path)$ //Get average RTT
03. $rtt = getPathRtt(path)$ //Get current RTT
04. $deliveryRate = getPathDeliveryRate(path)$ //Get current bandwidth
05. $lastDeliveryRate = getPathLastDeliveryRate(path)$ //Get previous bandwidth
06. **if** $rtt > averageRtt$ **and** $deliveryRate < lastDeliveryRate$ **then**
07. $oliaCown = oliaCown/2$ //Reduce congestion window
08. **else**
09. $oliaCown = oliaCown$ //Do not reduce congestion window
10. **end if**
11. **end for**

3. Multiplex Data Arrive In Order (MDAIO) Scheduling Algorithm for Satellite Networks

In power emergency communication scenarios, time-sensitive services such as relay protection differential signals and distribution network automation remote control commands impose stringent requirements on transmission timing logic (typically requiring an out-of-order rate below 0.1%). However, Low Earth Orbit (LEO) broadband satellite networks exhibit significant link heterogeneity: power terminals may simultaneously transmit data via low-latency satellite-ground direct links (fast paths) and high-latency inter-satellite multi-hop links (slow paths). This substantial latency difference (jitter), superimposed with high bit error packet loss under disastrous weather, easily leads to severe data packet disorder at the receiver. This, in turn, can trigger malfunction or refusal of operation in protection devices, threatening the security of the power grid.

To address this critical issue, this section proposes a Multiplex Data Arrive In Order (MDAIO) algorithm for satellite networks. This algorithm achieves "predictive" scheduling at the sender by constructing a multipath throughput prediction model that incorporates retransmission mechanisms. It introduces a closed-loop feedback mechanism based on reverse ACKs to correct prediction errors and

implements differentiated resource allocation combined with power service priorities, ensuring the orderly and deterministic transmission of critical instructions.

3.1. Data Pre-allocation Based on Forward Prediction

To avoid receiver disorder at the source, it is necessary to plan the data allocation ratio for each path in advance at the sender. Traditional scheduling algorithms often ignore the impact of packet loss retransmission on transmission timing. Therefore, this section proposes a data pre-allocation strategy based on forward prediction. By accurately calculating the parallel transmission capability of the fast path within the time consumed for the slow path to transmit one round of data, in-order pre-allocation is achieved.

Assume the Round-Trip Time, Retransmission Timeout, and Congestion Window of the fast path f and slow path s are RTT_f, RTO_f, w_f and RTT_s, RTO_s, w_s, respectively. The packet loss probability on a certain path r in the satellite network is p_r. Let $p_r(m|w_r)$ denote the probability of losing m packets when the path r congestion window is w_r. Since packet loss follows a Bernoulli distribution, it is expressed as:

$$p_r(m|w_r) = C_{w_r}^m \cdot p_r^m \cdot (1-p_r)^{w_r-m} = \frac{w_r!}{m!(w_r-m)!} \cdot p_r^m \cdot (1-p_r)^{w_r-m} \quad (12)$$

(1) Slow Path Transmission Time Calculation

Given the high packet loss characteristics of satellite channels, the modeling of slow path transmission time must include retransmission overhead, mainly covering three typical transmission scenarios:

Scenario 1: No-loss transmission. This is the ideal case where data is transmitted successfully at once. The physical time for all packets to reach the receiver only includes the one-way transmission delay. The probability of this scenario occurring is:

$$P_{s,1} = (1-p_s)^{w_s} \quad (13)$$

The time for all packets to reach the receiver is:

$$T_{s,1} = RTT_s/2 \quad (14)$$

Scenario 2: Fast retransmission recovery. When a small amount of packet loss occurs (the amount of lost data$m \leq w_s - 3$), the receiver can trigger the fast retransmission mechanism by sending duplicate ACKs. In this case, the retransmitted packets need to undergo additional round-trip delay. To accelerate recovery, this algorithm utilizes the fast path to return ACKs. The probability of this scenario is:

$$P_{s,2} = \sum_{m=1}^{w_s-3} p_s(m|w_s) = \sum_{m=1}^{w_s-3} C_{w_s}^m \cdot p_s^m \cdot (1-p_s)^{w_s-m} \quad (15)$$

The time for all packets to reach the receiver is:

$$T_{s,2} = RTT_s + RTT_f/2 \quad (16)$$

Scenario 3: Timeout retransmission. When packet loss is severe (the number of lost packets $m > w_s - 3$) or network congestion causes ACK loss, the sender cannot receive enough duplicate ACKs and must wait for the Retransmission Timeout (RTO) timer to expire before retransmitting. This is the most time-consuming case. The probability of this scenario is:

$$P_{s,3} = \sum_{m=w_s-2}^{w_s} p_s(m|w_s) = \sum_{m=w_s-2}^{w_s} C_{w_s}^m \cdot p_s^m \cdot (1-p_s)^{w_s-m} \quad (17)$$

The time for all packets to reach the receiver is:

$$T_{s,3} = RTO_s + RTT_s/2 \quad (18)$$

Synthesizing the above three scenarios, the expected time T_s for the slow path to complete one round of data transmission is calculated as follows:

$$T_s = \sum_{k=1}^{3} P_{s,k} \cdot T_{s,k} \quad (19)$$

(2) Fast Path Throughput Recursive Prediction

Within the time window of the slow path transmission time T_s, the fast path can usually perform multiple rounds of data transmission due to its shorter RTT. To realize data pre-allocation for each path, it is necessary to evaluate the amount of data $W_f(T_s)$ that can be transmitted on the fast path within time T_s.

Assume that the time taken for all data in the first round of transmission on the fast path to successfully reach the receiver is t_1, and the transmittable data volume is $w_f(t_1)$. Next, the transmittable data volume $W_f(T_s - t_1)$ on the fast path within the remaining time $T_s - t_1$ is calculated. In the second round of transmission, assuming the time taken for all data transmission is t_2 and the transmittable data volume is $w_f(t_2)$, then within the remaining time $T_s - t_1 - t_2$, the transmittable data is $W_f(T_s - t_1 - t_2)$. By analogy, this continues until the remaining time is 0. The sum of the data volume transmitted in all rounds is $W_f(T_s)$, and the recursive formula for this process is as follows:

$$W_f(T_s) = w_f(t_1) + W_f(T_s - t_1) = w_f(t_1) + w_f(t_2) + W_f(T_s - t_1 - t_2) = w_f(t_1) + w_f(t_2) + w_f(t_3) + \cdots \quad (20)$$

Next, this paper considers the solution process of the formula under different packet loss conditions. Let $j(j = 1,2,3\ldots)$ denote the transmission round of the fast path. $T_f(j)$ (where $T_f(j) = T_s$ when $j = 1$, and $T_f(j) < T_s$ when $j > 1$) is the remaining data transmission time at the start of the j-th round. $W_f(j)$ is the amount of data capable of being transmitted on the fast path within time $T_f(j)$, and $n_f(j)$ is the amount of data that can be transmitted in the j-th round. The recursive formula is thus modified as:

$$W_f(j) = n_f(j) + W_f(j+1) \quad (21)$$

This model simulates the transmission process of the fast path over time. For each round of the fast path, based on the length of the remaining available time $T_f(j)$, the algorithm divides the transmission capability into four typical cases for detailed derivation:

Case 1: Insufficient time ($T_f(j) < RTT_f/2$). This corresponds to an extremely short remaining time, insufficient to complete one-way data transmission (i.e., less than half an RTT). Physically, data packets cannot reach the receiver before the deadline, so the effective transmission volume for this round is 0. Its calculation formula is: $W_f(j) = 0$

Case 2: Only one round can be transmitted ($RTT_f/2 \leq T_f(j) < 3RTT_f/2$). The remaining time is sufficient to support only one round of normal data transmission but insufficient to support any form of packet loss retransmission (fast retransmission requires at least 1.5 RTTs). Therefore, if

packet loss occurs, lost packets cannot be recovered within this time window. The effective throughput is calculated as the current congestion window size minus the expected number of lost packets:

$$W_f(j) = w_f(j) - \sum_{m=0}^{w_f(j)} p_f\left(m|w_f(j)\right) \cdot m \quad (22)$$

Where $w_f(j)$ is the congestion window size of the fast path in the j-th round.

Case 3: Fast retransmission available ($3RTT_f/2 \leq T_f(j) < RTO_f + RTT_f/2$). The time is relatively ample. If a small amount of packet loss occurs during transmission, the system has enough time to recover via the fast retransmission mechanism. The throughput prediction in this case requires recursively calculating the expectations of three sub-branches. The total expected calculation formula is:

$$W_f(j) = \sum_{k=1}^{3} W_{f,k}(j) \quad (23)$$

Where $W_{f,k}(j)$ represents the expected transmission volume of the three sub-branches, specifically:

3-a (No-loss branch): The j-th round of transmission is completed successfully. The system enters the next round, the congestion window grows according to rules, and the remaining time is deducted by one RTT. The probability of this scenario is:

$$P_{f,1}(j) = \left(1 - p_f\right)^{w_f(j)} \quad (24)$$

The parameters for the next round (j+1) are updated as:

$$T_{f,1}(j+1) = T_f(j) - RTT_f \quad (25)$$

$$w_{f,1}(j+1) = \begin{cases} 2 \cdot w_f(j) & w_f(j) < ssthresh_f \\ w_f(j) + 1 & w_f(j) \geq ssthresh_f \end{cases} \quad (26)$$

Where $T_{f,1}(j+1)$ is the remaining time at the start of the (j+1)-th round, and $w_{f,1}(j+1)$ is the congestion window for the (j+1)-th round, $ssthresh_f$ is the slow start threshold for the fast path.

The expected transmission volume $W_{f,1}(j)$ is:

$$W_{f,1}(j) = P_{f,1}(j) \cdot w_f(j) + W_{f,1}(j+1) \quad (27)$$

3-b (Fast retransmission branch): A small amount of packet loss occurs but is successfully recovered via fast retransmission. This process increases time consumption, and the congestion window is halved before entering the next round of recursion. The probability of this scenario is:

$$P_{f,2}(j) = \sum_{m=1}^{w_f(j)-3} C_{w_f(j)}^{m} \cdot p_f^m \cdot \left(1 - p_f\right)^{w_f(j)-m} \quad (28)$$

The parameters for the next round state update are:

$$T_{f,2}(j+1) = T_f(j) - 2 \cdot RTT_f \quad (29)$$

$$w_{f,2}(j+1) = max\left(\lfloor w_f(j)/2 \rfloor, 2\right) \quad (30)$$

Where $T_{f,2}(j+1)$ is the remaining time at the start of the (j+1)-th round, and $w_{f,2}(j+1)$ is the congestion window for the (j+1)-th round. The calculation formula for the expected transmission volume $W_{f,2}(j)$ is:

$$W_{f,2}(j) = P_{f,2}(j) \cdot w_f(j) + W_{f,2}(j+1) \quad (31)$$

Where $W_{f,2}(j+1)$ is the data size that can be transmitted within the remaining time $T_{f,2}(j+1)$ at the start of the (j+1) -th round on the fast path in Case 3-b.

3-c (Timeout failure branch): Packet loss is severe and requires timeout retransmission, but the remaining time is insufficient to support the RTO wait. This means the transmission in this round is blocked and cannot proceed to the next round. The effective throughput only includes the successfully arrived portion. The probability of this scenario is:

$$P_{f,3}(j) = \sum_{m=w_f(j)-2}^{w_f(j)} C_{w_f(j)}^{m} \cdot p_f^m \cdot \left(1 - p_f\right)^{w_f(j)-m} \quad (32)$$

The data volume receivable at the receiver is:

$$W_{f,3}(j) = P_{f,3}(j) \cdot \left(w_f(j) - m\right) \quad (33)$$

Case 4: Timeout retransmission available ($T_f(j) \geq RTO_f + RTT_f/2$). Time is extremely ample. Even if severe congestion leads to a timeout (RTO), the system still has enough time to complete retransmission and continue to the next round. Its calculation logic is similar to Case 3, with the first two branches (no loss, fast retransmission) being the same. The main difference lies in the third branch:

4-a: The j-th round of transmission has no packet loss. Refer to Case 3-a.

4-b: The j-th round of transmission has packet loss, and the number of lost packets $m \leqslant w_f(j) - 3$. Refer to Case 3-b.

4-c (Timeout retransmission success): Severe packet loss triggers RTO. Since time is sufficient, the system completes retransmission after the RTO wait, the window is reset to the initial value, and recursion proceeds to the next round. The probability of this scenario is:

$$P_{f,3}(j) = \sum_{m=w_f(j)-2}^{w_f(j)} C_{w_f(j)}^{m} \cdot p_f^m \cdot \left(1 - p_f\right)^{w_f(j)-m} \quad (34)$$

he time remaining allows for timeout retransmission to complete the j-th round of data transmission, taking $RTO_f + RTT_f/2$. If there is still time remaining after timeout retransmission, the (j+1)-th round of transmission can proceed. The parameters for the (j+1) -th round are obtained as:

$$T_f(j+1) = T_f(j) - \left(RTO_f + RTT_f/2\right) \quad (35)$$

$$w_{f,3}(j+1) = 2 \quad (36)$$

Where $T_f(j+1)$ is the remaining time at the start of the (j+1)-th round, and $w_{f,3}(j+1)$ is the congestion window for the (j+1)-th round. The data volume receivable at the receiver is:

$$W_{f,3}(j) = P_{f,3}(j) \cdot w_f(j) + W_{f,3}(j+1) \quad (37)$$

The total expected calculation formula is:

$$W_f(j) = \sum_{k=1}^{3} W_{f,k}(j) \quad (38)$$

By combining the above 4 cases, the final mathematical expression for the data volume $W_f(j)$ that can be transmitted on the fast path within time $T_f(j)$ is derived as follows:

$$W_f(j) = \begin{cases} 0, & T_f(j) < RTT_f/2 \\ w_f(j) - \sum_{m=0}^{w_f(j)} p\left(m|w_f(j)\right) \cdot m, & RTT_f/2 \leq T_f(j) < 3RTT_f/2 \\ \sum_{i=1}^{3} W_{f,i}(j), & 3RTT_f/2 \leq T_f(j) < RTO_f + RTT_f/2 \\ \sum_{i=1}^{3} W_{f,i}(j), & T_f(j) \geq RTO_f + RTT_f/2 \end{cases} \quad (39)$$

Finally, by recursively calculating the transmission data volume for each round, the transmittable data volume W_f on the fast path is derived as:

$$W_f = W_f(j = 1) \quad (40)$$

Based on the above analysis, the transmittable data volumes Num_f and Num_s for the fast and slow paths can be calculated at the start of each new scheduling round on the slow path, using parameters such as RTT_f, RTT_s, RTO_f, RTO_s, w_f, w_s, p_f, and p_s. Then, in actual scheduling, data is allocated to the fast and slow paths according to Num_f and Num_s to realize the in-order arrival of data at the receiver.

Table 4. Flow of Data Pre-allocation Algorithm Based on Forward Prediction

Algorithm 4: Data Pre-allocation Algorithm Based on Forward Prediction
Input: Fast path parameter set :$path_f$=$\{RTT_f, RTO_f, w_f, p_f\}$ Slow path parameter set : $path_s$=$\{RTT_s, RTO_s, w_s, p_s\}$ **Output**: Allocatable data size for fast and slow paths : Num_f, Num_s
01. $P_{s,1}, P_{s,2}, P_{s,2} = getPathLossProbability(w_s, p_s)$ // Probability of three packet loss scenarios for slow path
02. $T_s = RTT_s/2 \cdot P_{s,1} + (RTT_s + RTT_f/2) \cdot P_{s,2} + (RTT_s/2 + RTO_s) \cdot P_{s,3}$
03. $Num_s = w_s$
04. $T_f = T_s$
05. $Num_s = getTransportPacketNum(path_f)$ // Data volume transmittable in remaining time
06. **Return** Num_f, Num_s

Table 4 shows the flow of the data pre-allocation algorithm based on forward prediction, with a time complexity of O(n).

This algorithm is invoked once every time the slow path receives an ACK, and updates the parameters of each path based on the current network status. The recursive algorithm flow for obtaining the data volume that can be transmitted on the fast path within the remaining time T_f is shown in Table 5, with a time complexity of O(n).

Table 5. Flow of Algorithm for Calculating Transmittable Data Volume in Remaining Time

Algorithm 5: getTransportPacketNum()
Input: Fast path parameter set: $path_f$=$\{RTT_f, RTO_f, w_f, p_f\}$, Remaining time T_f
Output: Allocated data size W_f for fast path $path_f$ within remaining time T_f
01. **if** $T_f < RTT_f/2$**then** //case 1
02. $W_f = 0$
03. **else if** $RTT_f/2 \le T_f < 3 \cdot RTT_f/2$ **then** //case 2
04. $W_f = w_f - \sum_{m=0}^{w_f} p_f(m|w_f) \cdot m$
05. **else if** $3 \cdot RTT_f \le T_f < RTO_f + RTT_f/2$ **then** //case 3
06. **if** $isPacketLoss() == 0$ **then** //case 3-a
07. $P_{f,1} = (1 - p_f)^{w_f}, T_{f,1} -= RTT_f, w_f = w_f + 1 \; or \; 1/w_f$
08. $W_{f,1} = P_{f,1} \cdot w_f + getTransportPacketNum(path_f, T_{f,1})$ //Recurse for remaining time
09. **else if** $isPacketLoss() \le w_s - 3$ **then** //case b
10. $P_{f,2} = \sum_{m=1}^{w_f-3} C_{w_f}^m \cdot p_f^m \cdot (1 - p_f)^{w_f-m}, T_{f,2} = T_f - 2 \cdot RTT_f, w_f = w_f/2$
11. $W_{f,2} = P_{f,2} \cdot w_f + getTransportPacketNum(path_f, T_{f,2})$ //Recurse for remaining time
12. **else** //case c
13. $P_{f,3} = \sum_{m=w_f-2}^{w_f} C_{w_f}^m \cdot p_f^m \cdot (1 - p_f)^{w_f-m}$
14. $W_{f,3} = P_{f,3} \cdot (w_f - m)$
15. **end if**
16. $W_f = \sum_{k=1}^{3} W_{f,k}$
17. **else** //case 4 case 4-a 4-bcalculation omitted (similar to Case 3)
18. **if** $isPacketLoss() > w_f - 3$ **then** //case 4-c
19. $P_{f,3} = \sum_{m=w_f-2}^{w_f} C_{w_f}^m \cdot p_f^m \cdot (1 - p_f)^{w_f-m}, T_{f,3} -= (RTO_f + RTT_f/2), w_f = 2$
20. $W_{f,3} = P_{f,3} \cdot w_f + getTransportPacketNum(Path_f, T_{f,3})$ //Recurse for remaining time
21. **end if**
22. $W_f = \sum_{k=1}^{3} W_{f,k}$
23. **end if**
24. **Return** W_f

3.2. Dynamic Adjustment Based on Reverse ACK

Due to the highly dynamic changes in LEO network topology, relying solely on a static forward prediction model inevitably leads to deviations. The accumulation of errors will cause the disorder problem to gradually emerge. Therefore, this section introduces a dynamic adjustment mechanism based on reverse ACK, which corrects the prediction model in real-time through closed-loop feedback, ensuring that the scheduling algorithm can adapt to the real-time jitter of satellite links.

This mechanism utilizes the sequence number information carried in the reverse ACK to compare the actual transmission boundary with the predicted boundary and calculate the scheduling error of the previous round, $e(n-1)$. If the actual transmission is faster than predicted, it implies that the prediction was conservative (positive error); otherwise, it was aggressive (negative error). The algorithm introduces an adjustment factor $\Delta(n)$ to dynamically correct the current predicted value:

The corrected predicted value $\widehat{W}_f(n)$ is calculated as follows:

$$\widehat{W}_f(n) = W_f(n) + \Delta(n) \quad (41)$$

Where $\widehat{W}_f(n)$ is the adjusted predicted value for the n-th round, and $\Delta(n)$ is the adjustment value for the slow path in the n-th round, updated as follows:

$$\Delta(n) = \Delta(n-1) + \alpha \cdot e(n-1) \quad (42)$$

Where $\Delta(n-1)$ is the adjustment value for the slow path in the n-th round, $e(n-1)$ is the scheduling deviation of the slow path in the (n-1) -th round, calculated as $e(n-1) = s - fseq$, and α is the update parameter, set to $\alpha = 1/8$.

It is worth noting that the highly dynamic topology of LEO satellite networks leads to frequent link handovers. To adapt to such rapid changes, the dynamic adjustment mechanism of the MDAIO algorithm can respond quickly to fluctuations in link status. When a satellite handover causes a sudden change in latency or packet loss rate, the sequence number information carried by the reverse ACK immediately reflects a significant scheduling error $e(n-1)$. Through the rapid iteration of the adjustment factor $\Delta(n)$, the prediction model can converge to the new link state within a few RTTs, thereby maintaining transmission continuity and orderliness during topology reconfiguration.

The dynamic adjustment algorithm based on reverse ACK is shown in Table 6, and the time complexity of this algorithm is O(1).

Table 6. Flow of Dynamic Adjustment Algorithm Based on Reverse ACK

Algorithm 6: Dynamic Adjustment Algorithm Based on Reverse ACK
Input: Slow path transmission round n, predicted value $W_f(n)$, fast path sequence number $fseq$, Slow path sequence number s **Output:** Adjusted predicted value $\widehat{W}_f(n)$
01. **if** $n == 1$ **then**
02. $\Delta(1) = 0$
03. $\widehat{W}_f(1) = W_f(1) + \Delta(1)$
04. **else**
05. $e(n-1) = s - fseq$
06. $\Delta(n) = \Delta(n-1) + \alpha \cdot e(n-1)$
07. $\widehat{W}_f(n) = W_f(n) + \Delta(n)$
08. **end if**
09. **Return** $\widehat{W}_f(n)$

3.3. Multi-Stream Scheduling Based on Service Priority Awareness

In power emergency networks, the importance of different traffic flows varies significantly. If transmission resources are allocated equally, critical control instructions are prone to being blocked by large-volume video data. Therefore, this section proposes a multi-stream scheduling strategy based on flow priority awareness, introducing service priority weights to ensure that strongly time-sensitive services can be transmitted preferentially when resources are limited.

To achieve differentiated scheduling, this paper draws on the priority dependency tree concept of HTTP/2 and extends a priority field in the QUIC Stream data structure to establish a mapping relationship between power services and stream priorities. Specifically, strongly time-sensitive services such as relay protection differential signals and distribution network automation remote control commands, which are extremely sensitive to delay and jitter and require zero-wait transmission, are assigned the Highest Priority (P1). Interactive services like emergency command voice and SCADA telemetry data, which need to guarantee fluency but allow slight jitter, are assigned Medium Priority (P2). Broadband services, including on-site panoramic video backhaul and fault recording file transmission, which mainly consume the remaining bandwidth, are assigned the Lowest Priority (P3).

Within the predicted available bandwidth window w_r, the algorithm does not allocate resources equally but distributes the transmission quota $Num_{r,i}$ based on the priority ratio of each service flow:

$$Num_{r,i} = \frac{priority_i}{\sum priority_i} \cdot w_r \quad (43)$$

where the current congestion window size of path r is w_r, and $priority_i$ is the priority of the service flow.

Based on the above allocation strategy, the scheduler executes the following logic:

- **High Priority First:** In the packet assembly stage, the P1 service flow queue is scanned first. As long as there is data in the P1 queue and the path has remaining window, it immediately encapsulates and transmits the packet.
- **Weight Sharing:** Only when P1 services have been sent or there are no P1 services, the remaining window is allocated to P2 and P3 services according to weight proportions. This ensures that during large-volume video transmission, once a burst control command is generated, the scheduler can immediately preempt bandwidth for transmission, achieving "queue-jumping" transmission for critical services.

In large-scale concurrent scenarios, absolute fairness may compromise critical services. Therefore, the priority scheduling strategy proposed in this section actually implements a form of "weighted fairness." By assigning larger weights to high-priority services (P1), the algorithm ensures that the throughput of core services such as relay protection is not squeezed by non-critical services (e.g., P3) during network congestion. This mechanism maximizes the effective utilization of limited satellite bandwidth while ensuring the security of the power grid.

Table 7 presents the algorithm flow of the multi-stream scheduling algorithm based on stream priority awareness, and the time complexity of this algorithm is O(n).

Table 7. Flow of Multi-Stream Scheduling Algorithm Based on Stream Priority Awareness

Algorithm 7: Multi-Stream Scheduling Algorithm Based on Stream Priority Awareness
01. **for each** path **do**
02. **while** $Num_r > 0$ **do**
03. **for each** $stream_i \in \{actives\}$**do**
04. $Num_{r,i} = priority_i / \sum priority_i \cdot w_r$
05. $assemblePacket(Num_{r,i})$ // Assemble packet
06. **end for**
07. $Num_r --$
08. **end while**
09. **end for**

4. Simulation Experiment and Result Analysis

4.1. Simulation Environment and Parameter Settings

To verify the performance of the proposed algorithms, a LEO power emergency communication simulation platform was built based on Docker. The simulation constructs two heterogeneous paths: Path 1 simulates a near-earth direct connection main path with low latency (23ms) and 10Mbps bandwidth; Path 2 simulates a high-dynamic, long-latency (150-240ms) backup path involving inter-satellite multi-hop transmission. Additionally, the experiment utilizes the TC tool to set a packet loss rate of 0%-5% to simulate a disastrous high bit error rate environment, selecting OLIA, Lowest-RTT, D-OLIA, and AIOS as comparative algorithms.

4.2. Transmission Efficiency and Survivability Analysis (AFR-DL Performance)

Figure 2 shows the comparison of average throughput under dual-path concurrent transmission for different algorithms. The experimental results indicate that the bandwidth utilization of the AFR-DL algorithm on both the Main Path (Path 1) and Backup Path (Path 2) is significantly better than that of the traditional OLIA algorithm. Especially on the long-delay Backup Path, thanks to the "ACK Fast Return Based on Best Reverse Path" mechanism of AFR-DL, confirmation information is no longer constrained by long link delays, resulting in a substantial increase in the congestion window growth rate.

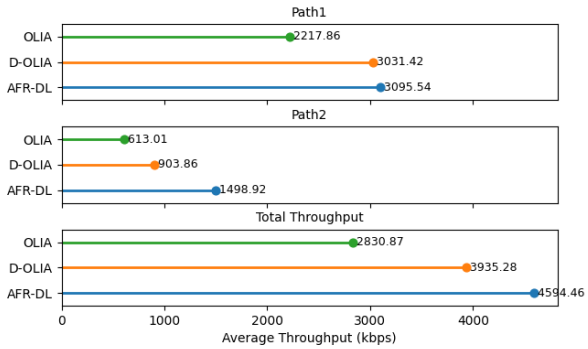

Figure 2. Comparison of average throughput for each path

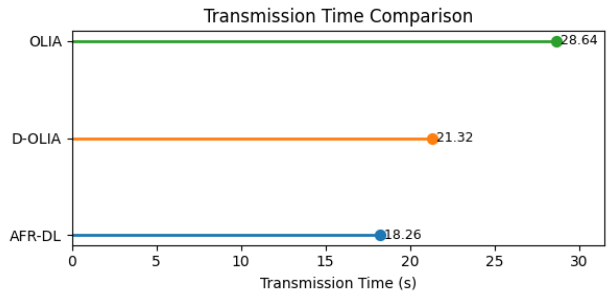

Figure 3. Comparison of file transfer times for various algorithms

Figure 3 further statistics the completion time for transmitting a 10MB power emergency monitoring data block (simulating panoramic sensing data). The transmission time of the AFR-DL algorithm is only 18.26s, shortened by 14.17% and 38.3% compared to D-OLIA and OLIA, respectively. This indicates that in emergency situations where every second counts, the proposed method can significantly shorten the return cycle of on-site situational data, winning valuable time for decision-making at the command center.

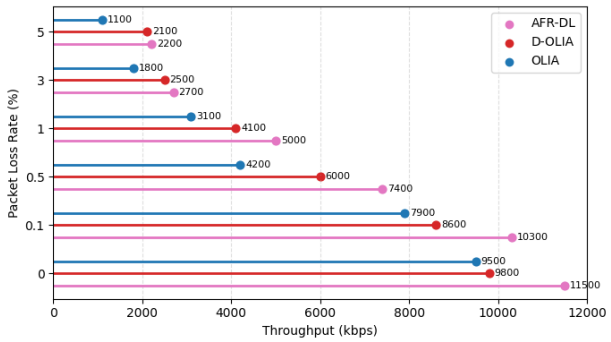

Figure 4. Comparison of average throughput of congestion algorithms under different packet loss conditions

Power emergency communication often faces severe weather such as typhoons and rainstorms, leading to soaring link bit error rates. Figure 4 shows the trend of total throughput changes for each algorithm under environments with packet loss rates ranging from 0% to 5%. As can be seen from the figure, as the packet loss rate increases, the traditional OLIA algorithm, unable to distinguish between congestion loss and error loss, incorrectly and frequently halves the congestion window, leading to a sharp collapse in throughput. In contrast, the AFR-DL algorithm introduces a packet loss decision mechanism based on bandwidth-delay gradients, maintaining a high throughput level even in severe packet loss scenarios up to 5%. This verifies that the algorithm possesses strong robustness and can guarantee the continuity of "lifeline" communication in extreme disaster environments.

4.3. Time-Sensitive Service Scheduling and Ordering Analysis (MDAIO Performance)

Figure 5 compares the throughput performance of different scheduling algorithms under dynamic packet loss environments. The Lowest-RTT algorithm suffers the most severe performance decay as the packet loss rate rises, as it lacks prediction for packet loss retransmission, causing the slow path to block the fast path.

In contrast, the MDAIO algorithm maintains the highest throughput across all packet loss rate intervals. Even at a high packet loss rate of 5%, MDAIO utilizes the forward throughput prediction model to accurately perceive the transmission capability of each path and dynamically adjust the data allocation ratio. This ensures that power service data can still be transmitted efficiently and concurrently in the LEO network with frequent link jitter.

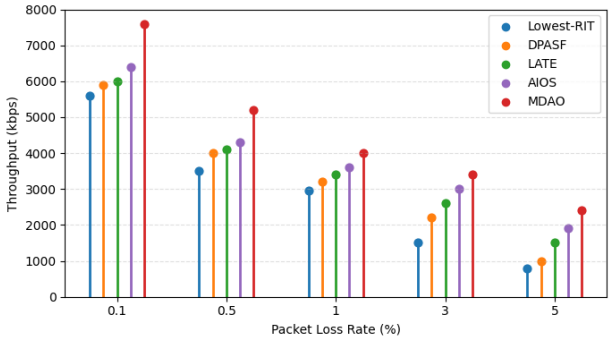

Figure 5. Throughput under different packet loss rate conditions

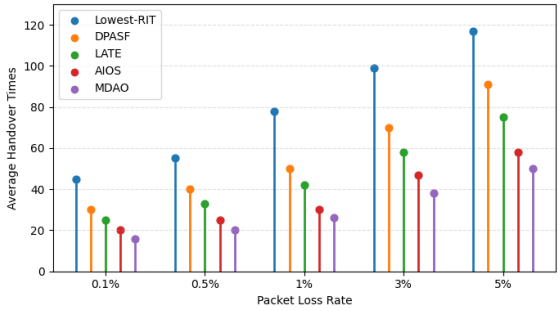

Figure 6. The number of out-of-order frames under different packet loss rates

Receiver data disorder is the core hidden danger threatening the safety of power time-sensitive services. Figure 6 shows the variation of the number of disordered frames at the receiver with the packet loss rate.

Experimental results show that as the link status deteriorates, the number of disordered frames for algorithms like Lowest-RTT and DPSAF rises exponentially. This will cause the receiver reordering buffer to overflow, increasing processing delay and even leading to service logic errors. However, the MDAIO algorithm consistently maintains the number of disordered frames at an extremely low level. This is attributed to the reverse ACK dynamic error correction mechanism introduced by MDAIO, which can compensate for prediction deviations in real-time.

For power emergency networks, this characteristic is crucial: it ensures that relay protection differential signals and remote open/close commands can arrive strictly in order, effectively avoiding secondary power grid accidents caused by timing disorders, verifying the excellent performance of the proposed method in ensuring the logical correctness of time-sensitive services.

5. Conclusions

This paper focuses on the scheduling problem of time-sensitive services for LEO broadband satellites in power emergency networks and proposes a transmission scheduling scheme integrating an improved QUIC protocol. The AFR-DL algorithm optimizes the congestion control mechanism under severe weather, avoiding blind window reduction caused by random packet loss and significantly improving the throughput of emergency links. Combined with the MDAIO algorithm's throughput prediction and service priority awareness strategy, it effectively guarantees the in-order arrival and deterministic transmission of strongly time-sensitive services. Simulation experiments verify the significant advantages of this method in improving transmission efficiency and reducing data disorder. Future work will further explore the seamless service handover mechanism under the periodic changes of the high-dynamic topology of LEO constellations and conduct in-depth research on fair resource scheduling strategies under multi-class power service concurrency scenarios to further enhance the robustness of the power emergency communication system.

Additionally, with the development of artificial intelligence, introducing Machine Learning (ML) for link quality prediction is a highly promising direction. Utilizing models such as LSTM or reinforcement learning to mine the spatiotemporal correlations of LEO satellite motion can enable long-term prediction of link quality (e.g., SNR, remaining visibility time). This would endow the scheduling strategy with greater foresight, such as performing traffic migration before a handover occurs, thereby further reducing transmission jitter. Future work will explore lightweight edge intelligence algorithms to achieve this goal under limited terminal computing power.

Acknowledgements

This work was supported by Science and Technology Project of State Grid Qinghai Electric Power Company (No:522800250001).

References

[1] Chen K, Liang G, Zhang H, et al. Resilient task offloading in integrated satellite-terrestrial networks with mobility-induced variability[J]. Digital Communications and Networks, 2025.

[2] Li J, Chai R, Gui K, et al. Joint Task Offloading and Resource Scheduling in Low Earth Orbit Satellite Edge Computing Networks[J]. Electronics, 2025, 14(5): 1016.

[3] Sun Y, Yang Y, Hawbani A, et al. QoS Optimization Strategy Based on D-GNN for LEO Satellite-Assisted Aviation Networks[J]. Computer Networks, 2025: 111741.

[4] Bhattacharjee D, Madoery P G, Naik A, et al. DSROQ: Dynamic Scheduling and Routing for QoE Management in LEO Satellite Networks[J]. arXiv preprint arXiv:2508.21047, 2025.

[5] Kamel V, Zhao J, Li D, et al. StarQUIC: Tuning congestion control algorithms for QUIC over LEO satellite networks[C]//Proceedings of the 2nd International Workshop on LEO Networking and Communication. 2024: 43-48.

[6] Khan F, Hervella C, Diez L, et al. Realistic assessment of transport protocols performance over LEO-based communications[J]. Computer Networks, 2023, 236: 110008.

[7] Wang L, Wang Z, Deng Z, et al. ALCS: An Adaptive Latency Compensation Scheduler for Multipath TCP in Satellite-Terrestrial Integrated Networks[J]. arXiv preprint arXiv:2503.07973, 2025.

[8] QIN Yuqing, LIU Haojun, ZHANG Chen, et al. User QoS-Oriented Resource Scheduling for Low Earth Orbit Satellite Hybrid NOMA[J]. Mobile Communications, 2025,49(7): 125-134.

[9] Yang W, Cai L, Shu S, et al. Mobility-aware congestion control for multipath QUIC in integrated terrestrial satellite networks[J]. IEEE Transactions on Mobile Computing, 2024, 23(12): 11620-11634.

[10] Ouyang M, Zhang R, Wang B, et al. Network coding-based multipath transmission for LEO satellite networks with domain cluster[J]. IEEE Internet of Things Journal, 2024, 11(12): 21659-21673.

[11] Elhachi H, Boumehrez F, Aymen Labiod M, et al. Smart cross-layer approach to multi-access terrestrial and non-terrestrial networks (NTNs): Real-time mobile-health use case[J]. International Journal of Communication Systems, 2024, 37(18): e5941.

[12] Iyengar J, Thomson M. QUIC: A UDP-based multiplexed and secure transport[M]//RFC 9000. 2021.

[13] Qin X, Zhang T, Yu K, et al. Dynamic Time-Difference QoS Guarantee in Satellite–Terrestrial Integrated Networks: An Online Learning-Based Resource Scheduling Scheme[J]. Engineering, 2025.

[14] Liu W, Xiao N, Liu B, et al. Optimizing Time-Sensitive Traffic Scheduling in Low-Earth-Orbit Satellite Networks[J]. Sensors, 2025, 25(14): 4327.

[15] Yang W, Shu S, Cai L, et al. MM-QUIC: Mobility-aware multipath QUIC for satellite networks[C]//2021 17th International Conference on Mobility, Sensing and Networking (MSN). IEEE, 2021: 608-615.