

QUIC-Space: adaptive FEC-enhanced QUIC for reliable deep space communication

Jianhao Yu¹, Ye Li², Wenfeng Li¹ & Kanglian Zhao^{1*}

¹*School of Electronic Science and Engineering, Nanjing University, Nanjing 210093, China*

²*School of Information Science and Technology, Nantong University, Nantong 226019, China*

Appendix A Research background and problem statement

QUIC is a UDP-based transport protocol that has emerged as a key candidate for next-generation internet communication. It inherits critical features from TCP, UDP, and SCTP, including a reliable byte stream abstraction, stream multiplexing, and unreliable datagrams. QUIC's modular architecture offers significant flexibility and extensibility. First, it encrypts most of its packet headers and all payloads, effectively addressing the ossification problem that can prevent protocol evolution due to middlebox interference. Second, QUIC packets are composed of frames with specific functions (e.g., the **STREAM** frame carries application data and marks the end of the stream with the FIN bit). This frame-based design allows for implementing extensive protocol extensions by defining new frames with distinct behaviors.

Similar to the Bundle Protocol/Licklider Transmission Protocol (BP/LTP) stack commonly employed in deep-space communications, QUIC also adopts a store-and-forward mechanism to withstand intermittent links. The Bundle Protocol (BP) operates at the application layer, encapsulating data into Bundles and explicitly implementing store-and-forward functionality. BP itself relies on a convergence-layer protocol—such as the Licklider Transmission Protocol (LTP), traditional TCP/UDP, or even QUIC [1]—for actual data delivery across networks. A fundamental limitation of the BP/LTP stack is its strictly point-to-point operation, which impedes support for advanced network functions such as multipath transmission and dynamic topology management.

By contrast, QUIC runs at the transport layer and leverages IP-layer store-and-forward capabilities to furnish applications with a stable socket interface. Owing to its heritage in terrestrial networks, QUIC can interoperate seamlessly with ground infrastructure without requiring BP-to-IP translation at each node. Moreover, QUIC readily reuses mature ground-network technologies—most notably TLS 1.3 for security and the HTTP/3 application-layer protocol—thereby accelerating deployment. QUIC also exhibits high extensibility: its plugin architecture permits the flexible integration of forward-error-correction (FEC) schemes (e.g., QUIC-FEC [2]) or disruption-aware mechanisms. As a result, QUIC has the potential to match BP in reliability while offering broader interoperability and benefiting from a more active development ecosystem.

Congestion control and flow control are critical factors influencing the performance of QUIC. Congestion control regulates the network load by adjusting the congestion window to prevent congestion, while flow control manages the receiver's buffer through a sliding window to prevent data overflow. However, due to orbital dynamics, deep space links may intermittently disconnect. Existing congestion control algorithms all face challenges in this situation. Given that space missions are typically scheduled, and parameters such as maximum Round-Trip Times (RTTs) or bandwidth are predetermined, in this letter, congestion and flow control are managed by setting a fixed window size during connection establishment.

Research [3] provides recommendations for the configuration of QUIC in deep space networks. By adjusting protocol parameters, QUIC can be deployed in deep space networks. However, the loss recovery mechanism in QUIC requires additional attention. QUIC's default loss recovery mechanism is entirely dependent on retransmission [4]. Unlike TCP, QUIC typically considers a packet X lost if either its retransmission timeout (often set to $9/8 * RTT$) expires, or if an acknowledgment is received for a packet sent after X (packet threshold loss detection). To mitigate the effects of reordering often exacerbated by long delays in deep space, this work primarily employs a modified time threshold for loss detection, set to $2 * RTT$.

In traditional terrestrial networks, QUIC's retransmission-based recovery mechanism quickly recovers lost packets. However, in deep-space networks, the extremely long RTT significantly increases packet recovery time, introducing several unique challenges that severely impact goodput and delay. First, the extended recovery delay exacerbates tail loss issues, severely degrading goodput in short-flow transmission scenarios. Second, the prolonged recovery time results in significant head-of-line blocking (HOL) within each stream, increasing data delivery delay and causing goodput fluctuations. Furthermore, even for delay-tolerant tasks, the extended recovery time leads to substantial packet buildup in the receive buffer, necessitating the receiver to reserve sufficient buffer space and efficiently perform data reordering. If the receiver buffer size is insufficient, the flow control mechanism may block new packet transmissions—even during periods of network idleness—to prevent buffer overflow, which can lead to severe underutilization of available network bandwidth. These challenges underscore the need for a more robust loss recovery mechanism in QUIC for deep-space environments.

* Corresponding author (email: zhaokanglian@nju.edu.cn)

Appendix B Modeling QUIC and QUIC-Space goodput over lossy networks

To accurately analyze the theoretical upper bound of goodput over lossy networks, we construct a discrete-time transmission model. This section discusses the expected goodput of QUIC and QUIC-Space respectively under this model.

Appendix B.1 Transmission model and assumptions

A discrete-time transmission system is considered under the following core parameters and assumptions:

1. **Sender Behavior:** The sender is *ideal* in that it can transmit at a constant rate of BW packets per time slot.
2. **Round-Trip Time (RTT):** The network round-trip time is assumed to be constant and equals RTT time slots.
3. **Loss Model:** Packet losses occur independently according to a Bernoulli process, with each packet being lost with probability e .
4. **Loss Detection:** We assume that the sender can accurately determine whether each packet has been successfully delivered or lost exactly RTT time slots after transmission.

We make these assumptions to analyze the performance upper bounds of QUIC and QUIC-Space. A more realistic model is left for our future work.

Appendix B.2 Modeling QUIC goodput over lossy networks

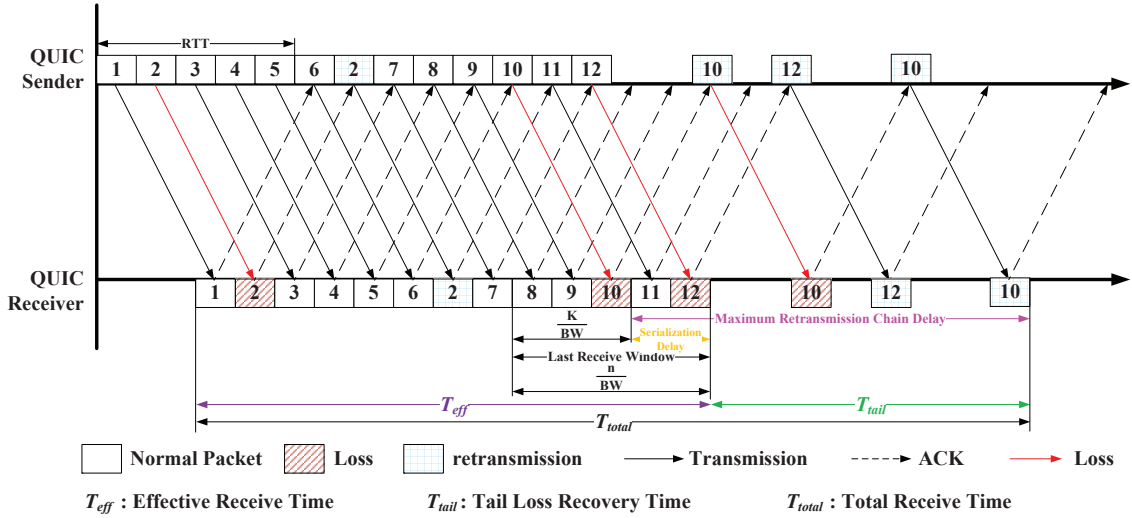


Figure B1 QUIC transmission example.

In this section, a model is proposed to analyze the goodput of QUIC under the proposed transmission framework. Special consideration is given to the impact of tail loss recovery time on QUIC goodput in this model. Fig. B1 illustrates a transmission example with $BW = 1$ (pkt./time slot) and $RTT = 5$ time slots. In this scenario, a total of 12 packets are transmitted. Among these, packets 2, 10, and 12 are lost (red blocks) but are successfully recovered through subsequent retransmissions (blue blocks). The average goodput can be defined as the total number of successfully transmitted data divided by the total receive time T_{total}^{QUIC} . To facilitate quantitative analysis, T_{total}^{QUIC} is decomposed into two components: the effective receive time T_{eff} and the tail loss recovery time T_{tail} , with $T_{total}^{QUIC} = T_{eff} + T_{tail}$. This decomposition isolates the bulk data transfer from the delays incurred specifically by the tail packet losses.

1. **Effective Receive Time (T_{eff}):** Defined as the duration from the arrival of the first packet to the first arrival of the final packet, as shown in purple in Fig. B1. From the sender's perspective, it spans the elapsed time from the first send of the first packet to the first send of the last packet. This metric captures the sender's active, continuous phase of original data transmission—including any retransmissions triggered within this period—while excluding any subsequent, dedicated retransmission rounds for tail-loss recovery. The length of T_{eff} is primarily determined by the total number of packets initially sent and the link's bandwidth.
2. **Tail Loss Recovery Time (T_{tail}):** Defined as the duration from the first arrival of the final packet to the point at which all packets have been successfully recovered. As depicted in green in Fig. B1. Its length depends on the RTT and the tail window's loss behavior and is independent of link bandwidth.

We decompose total receive time because the two time components are governed by distinct factors, which are modeled separately below.

Appendix B.2.1 Modeling of effective receive time

Let S denote the total number of packets that must be delivered successfully. On a link with packet-loss probability e , the expected total number of transmissions required for S packets to be eventually delivered is $E\{N_s\} = \frac{S}{1-e}$. Let $BDP = BW \times RTT$ denote the path bandwidth delay product (BDP). We define the size of the final receive window as $n = \min(S, BDP)$, representing the maximum number of packets that can be received within one RTT. Let $L_{last} \sim \text{Binomial}(n, e)$ denote the number of packets lost within this window. T_{eff} is determined by the total number of transmissions, $E\{N_s\}$, minus the retransmissions that are dedicated to recover the tail losses packets. We approximate these specific retransmissions by the expected number of initial losses in the final window, $E\{L_{last}\} = ne$. Therefore, the expected effective receive time is given by:

$$E\{T_{eff}\} = \frac{E\{N_s\} - E\{L_{last}\}}{BW} = \frac{\frac{S}{1-e} - e \cdot \min(S, BDP)}{BW}. \quad (B1)$$

Appendix B.2.2 Modeling of tail loss recovery time

The duration of T_{tail} is determined by the loss-recovery process within the final receive window. Two key random variables govern this process: the maximum number of retransmissions required by any lost packet in that window, denoted by M , and the position K of the last-arriving packet within the window. Due to the i.i.d. nature of packet losses, the number of retransmissions required for any single packet is statistically independent of its position within the window.

Maximum number of retransmissions (M)

Suppose that ℓ packets are lost in the final receive window. Let X_i denote the number of transmissions needed to recover the i -th lost packet. The random variables X_i are independent and identically distributed according to the geometric distribution

$$P(X_i = k) = (1 - e)e^{k-1}, \quad k = 1, 2, \dots \quad (B2)$$

We are concerned with the largest of these retransmission counts. Define $M = \max\{X_1, X_2, \dots, X_\ell\}$. Conditioning on $L_{last} = \ell$, the cumulative distribution function (CDF) of M is

$$P(M \leq m \mid L_{last} = \ell) = \prod_{i=1}^{\ell} P(X_i \leq m) = \left[\sum_{j=1}^m (1 - e)e^{j-1} \right]^\ell = (1 - e^m)^\ell. \quad (B3)$$

Marginalizing over ℓ yields

$$P(M \leq m) = \sum_{\ell=0}^n P(M \leq m \mid L_{last} = \ell) P(L_{last} = \ell) = [(1 - e) + e(1 - e^m)]^n = (1 - e^{m+1})^n. \quad (B4)$$

Finally, by applying the tail-sum formula, the expectation of M is given by

$$E\{M\} = \sum_{k=0}^{\infty} P(M > k) = \sum_{k=0}^{\infty} [1 - (1 - e^{k+1})^n]. \quad (B5)$$

Position of the last packet to arrive (K)

Packets within the final receive window are indexed $1, 2, \dots, n$. Let A_i denote the total number of attempts required for the i -th packet to be delivered successfully, which also follows a geometric distribution: $P(A_i = j) = (1 - e)e^{j-1}$. The final arrival time for packet i is defined as $T_i = T_{start} + \frac{i}{BW} + RTT \times (A_i - 1)$. Among them, T_{start} represents the start time of the last receiving window. To simplify the derivation, we normalize T_{start} to 0 and $\frac{1}{BW}$ to 1.

The random variable K denotes the position of the last packet to arrive, such that $T_K = \max_{1 \leq i \leq n} T_i$. Our goal is to calculate its expected value, $E\{K\}$.

A packet k becomes the last to arrive if:

1. $A_k = m$, meaning the k -th packet was successfully delivered on its m -th attempt.
2. For any $i \neq k$, $T_i < T_k$.

This implies:

$$T_i < T_k \implies i + RTT \times (A_i - 1) < k + RTT \times (m - 1) \implies A_i < m + \frac{k - i}{RTT}. \quad (B6)$$

Define $L_i(k, m) = \left\lfloor m + \frac{k - i}{RTT} \right\rfloor$. Then:

$$P(K = k \mid A_k = m) = \prod_{i \neq k} P(A_i \leq L_i(k, m)) = \prod_{i \neq k} [1 - e^{L_i(k, m)}]. \quad (B7)$$

By the law of total probability:

$$P(K = k) = \sum_{m=1}^{\infty} P(A_k = m) P(K = k \mid A_k = m) = \sum_{m=1}^{\infty} (1 - e)e^{m-1} \prod_{i \neq k} [1 - e^{L_i(k, m)}]. \quad (B8)$$

Therefore, the expected position is:

$$E\{K\} = \sum_{k=1}^n kP(K=k). \quad (B9)$$

Note that the calculations for K and M involve convergent infinite series, which can be truncated for practical computation. These terms are essential as they dominate the transmission time for small file transfers.

Combining the above, the expected tail recovery time is determined by the longest retransmission chain and the order position of the last packet:

$$E\{T_{tail}\} = E\{M\} \times RTT - \left(\frac{n - E\{K\}}{BW}\right). \quad (B10)$$

This expression intuitively reflects that $E\{T_{tail}\}$ is the additional delay incurred by tail losses. It accounts for the expected delay from the maximum retransmission chain ($E\{M\} \times RTT$, illustrated in magenta in Fig. B1), which represents the critical path for the last-arriving packet's recovery. This retransmission time, however, can partially overlap with the serialization of subsequent packets within the final receive window. The term $\frac{n - E\{K\}}{BW}$ quantifies the expected time required to serialize the packets occupying positions from $E\{K\}$ to n (orange in Fig. B1). By subtracting this concurrent serialization component, the formula accurately captures the portion of the tail recovery time that is not masked by ongoing data transmission.

Appendix B.2.3 Expected goodput of QUIC

Based on the above analysis, the total receive time T_{total}^{QUIC} can be expressed as

$$\begin{aligned} E\{T_{total}^{QUIC}\} &= E\{T_{eff}\} + E\{T_{tail}\} \\ &= \frac{\frac{S}{1-e} - e \cdot \min(S, BDP)}{BW} \\ &\quad + \sum_{m=1}^{\infty} m([1 - e^m]^n - [1 - e^{m-1}]^n) \times RTT \\ &\quad - \frac{\min(S, BDP) - \sum_{k=1}^n k \sum_{m=1}^{\infty} (1-e)e^{m-1} \prod_{i \neq k} [1 - e^{L_i(k,m)}]}{BW}. \end{aligned} \quad (B11)$$

Therefore, the expected average goodput $E\{G_{QUIC}\}$ can be expressed as:

$$E\{G_{QUIC}\} = S/E\{T_{total}^{QUIC}\}. \quad (B12)$$

Appendix B.3 Modeling QUIC-Space goodput over lossy networks

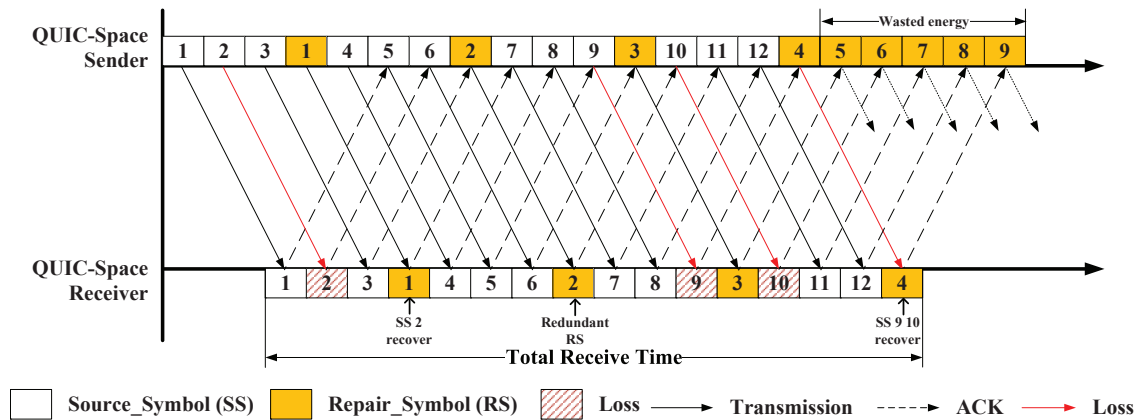


Figure B2 QUIC-Space transmission example.

In this section, the goodput of the QUIC-Space under the proposed transmission framework is analyzed. Fig. B2 illustrates a representative transmission scenario with bandwidth $BW = 1$ packet per time slot and round-trip time $RTT = 5$ time slots. In this example, a repair symbol is inserted after every three source symbols, corresponding to 25% redundancy. Although the packet-loss time slots coincide with those shown in Fig. B1, the sequence numbers of lost packets may differ due to the additional repair symbols.

On a link exhibiting packet-loss rate e , the QUIC-Space system transmits S source symbols with redundancy $e + \delta$. Consequently, the total number of symbols sent is $\frac{S}{1-(e+\delta)}$. Noted that upon completion of this initial transmission, some source symbols may remain unrecovered.

Building on our prior work [5], when losses follow a Bernoulli process and repair symbols are inserted at fixed intervals, the reception and decoding of source symbols can be modeled as a virtual $M/D/1$ queue. In this model, each lost source symbol corresponds to a customer arrival, and each received repair symbol corresponds to a service completion. The arrival rate is $\lambda = BW \times e \times (1 - e - \delta)$, and the service rate is $\mu = BW \times (e + \delta) \times (1 - e)$. Defining the system utilization as $\rho = \lambda/\mu$, the Poisson Arrivals See Time Averages (PASTA) theorem implies that the expected number of unrecovered source symbols in the system at any given time is $E\{L\} = \rho + \frac{\rho^2}{2(1-\rho)}$. To recover these $E\{L\}$ symbols, an additional $\frac{E\{L\}}{1-e}$ repair symbols must be transmitted.

The total reception time $T_{\text{total}}^{\text{QUIC-Space}}$ comprises the time to transmit the initial symbol set plus the extra time incurred by the additional repair symbols required to clear the decoding backlog. Dividing the total symbol count by the link bandwidth yields:

$$E\{T_{\text{total}}^{\text{QUIC-Space}}\} = \frac{S}{1 - (e + \delta)} + \left(\rho + \frac{\rho^2}{2(1-\rho)} \right) \frac{1}{1-e}. \quad (\text{B13})$$

Therefore, the expected average goodput $E\{G_{\text{QUIC-Space}}\}$ can be expressed as:

$$E\{G_{\text{QUIC-Space}}\} = S/E\{T_{\text{total}}^{\text{QUIC-Space}}\}. \quad (\text{B14})$$

Appendix B.4 Model verification and analysis

To corroborate our analytical model, we validate the proposed scheme within a discrete-time transmission framework. The accompanying verification code is publicly available¹⁾. Fig. B3 compares simulation results with theoretical predictions from Eq. (B12) and Eq. (B14) for both QUIC-Space and standard QUIC. We conduct these evaluations under a bandwidth $BW = 1$ pkt/slot, a round-trip time $RTT = 4000$ slots, varying packet-loss rates, and diverse file sizes.

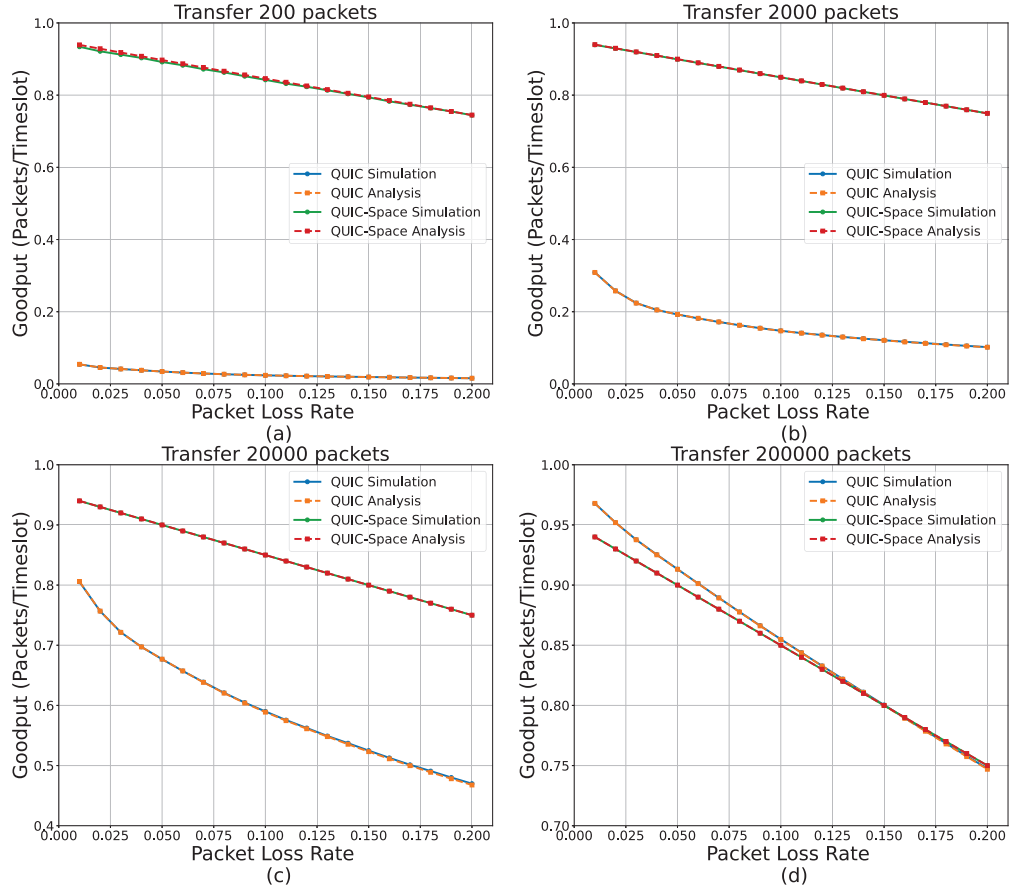


Figure B3 QUIC-Space and QUIC goodput: simulation vs. analysis.

Fig. B3 illustrates that QUIC-Space's goodput exhibits independence from RTT , a core advantage of the proposed scheme. Consistent with Eq. (B12) and Eq. (B14), Fig. B3 demonstrates that QUIC-Space achieves higher goodput than standard QUIC for small files or high loss rates by effectively mitigating tail-loss delays. Conversely, standard QUIC slightly

1) <https://github.com/JianHaoYu/ModingQUICandQUICSpace>

outperforms QUIC-Space under low loss rates or large file transfers. These findings confirm the accuracy of our theoretical model and lay the groundwork for the subsequent performance evaluation.

Appendix C Performance evaluation

Appendix C.1 Simulation setup

This section presents the evaluation results of QUIC-Space in a simulated Earth-Moon network. We implement the proposed extension in quic-go, a widely used open-source QUIC implementation. The parameter δ is set to 0.02. For comparison, we select the default quic-go, as well as interleaved XOR and RS algorithms in the existing QUIC-FEC framework [2].

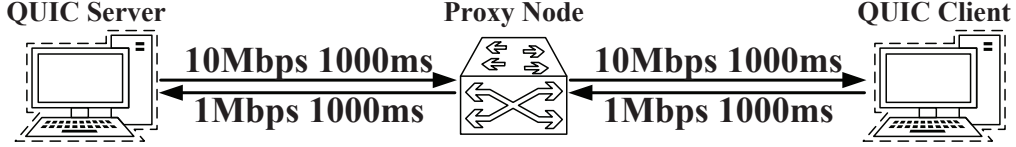


Figure C1 Network topology for performance evaluation.

The evaluation is conducted using the network emulation software Mininet, with the topology shown in Fig. C1. In our experiments, the QUIC client requests files of 3.5 MB, 35 MB, and 700 MB to simulate traffic flows of varying durations. Given the asymmetric bandwidth characteristics of Earth-Moon communication, the downlink is set to 10 Mbps, the uplink to 1 Mbps, and the one-way propagation delay to 2 seconds. Assuming uncorrelated losses [6], we introduce a 1% packet loss rate to simulate the data losses caused by channel error over space communication links²⁾ We acknowledge that more realistic models, such as the Gilbert–Elliott burst loss model, are worth considering. However, the long delays and intermittent connectivity of deep-space networks necessitate more sophisticated adaptive coding control mechanisms. We plan to explore this in future research. The evaluation considers two scenarios: continuous connectivity and network disruptions. In the disrupted network scenario, following the store-and-forward paradigm described in [7], we introduce an intermediary proxy node that intercepts and stores all passing packets when the network is disconnected. Once the network is restored, the proxy forwards the stored packets to the client.

To compare the performance across these scenarios, we employ four key metrics. Let B_r denote the number of bytes received in order by time t_r . The goodput at t_r can be expressed as $goodput(t_r) = \frac{B_r}{t_r}$. Unlike throughput (total received data over elapsed time), it excludes retransmissions and out-of-order arrivals, reflecting the application-perceived rate. Let t_s denote the time at which these B_r bytes were first transmitted. The in-order delivery delay D_i is defined as $D_i = t_r - t_s$, which captures end-to-end delay for successfully delivered data. Furthermore, we assess resource cost by tracking the total transmitted bytes as a proxy for network overhead and energy consumption, and by measuring the receiver buffer size required to store out-of-order packets.

Appendix C.2 Results and discussion

Fig. C2 illustrates the performance of QUIC-Space compared to the other three protocols when using a flow control window of 5 MB, equivalent to the link Bandwidth-Delay Product (BDP). As shown in Fig. C2, the goodput of QUIC-Space significantly exceeds that of the other three protocols. Specifically, QUIC-Space achieves goodput that is 348.3%, 266.4%, and 263.1% of the best-performing comparison option, RSQUIC, when transmitting 3.5 MB, 35 MB, and 700 MB files, respectively. Additionally, Fig. C2 also shows that as the file size increases from 3.5 MB to 35 MB, QUIC-Space's goodput decreases by 18.1%. This decline occurs because a 3.5 MB file is smaller than the 5 MB flow control window, ensuring that transmission is not blocked by flow control. However, for larger file transfers, the flow control becomes a limiting factor, blocking packet transmissions even when the link is idle.

To further examine the impact of flow control, Fig. C3 presents the goodput of QUIC-Space and three other protocols when transferring a 35 MB file under flow control windows ranging from 5 MB to 40 MB. The results reveal a positive correlation between goodput and flow control window size for all four protocols. As shown in Fig. C3, QUIC-Space consistently achieves the highest goodput with minimal fluctuations, primarily due to its FEC-based loss recovery mechanism that eliminates retransmission delays. In contrast, XORQUIC and RSQUIC, despite incorporating FEC, exhibit inferior performance. This discrepancy arises because QUIC-Space employs an adaptive redundancy control mechanism that dynamically adjusts redundancy levels based on network conditions, thereby avoiding unnecessary overhead. Under conditions of a sufficient flow control window (40 MB), QUIC's goodput remains 26.8% lower than that of QUIC-Space. This is because, although QUIC avoids flow control blocking, its prolonged tail loss recovery time degrades goodput, especially during small file transfers. Due to the long propagation delays characteristic of deep space networks, each tail loss event introduces a retransmission delay of at least 4 seconds, significantly reducing goodput. In contrast, by relying entirely on FEC-based loss recovery, QUIC-Space decouples loss recovery from RTT, mitigating the goodput degradation caused by tail losses.

²⁾ This assumption is based on the fact that bit errors, even correlated within the same frame, will result in the discarding of that frame upon Cyclic Redundancy Check validation. When multiple frames are affected by fading, interleaving techniques can distribute errors across different frames, eventually restoring the pattern of uncorrelated losses [6]. We acknowledge that correlated losses are a common phenomenon in free-space optical communication. However, this letter does not focus on a specific transmission technology but rather on establishing the baseline performance of QUIC-Space.

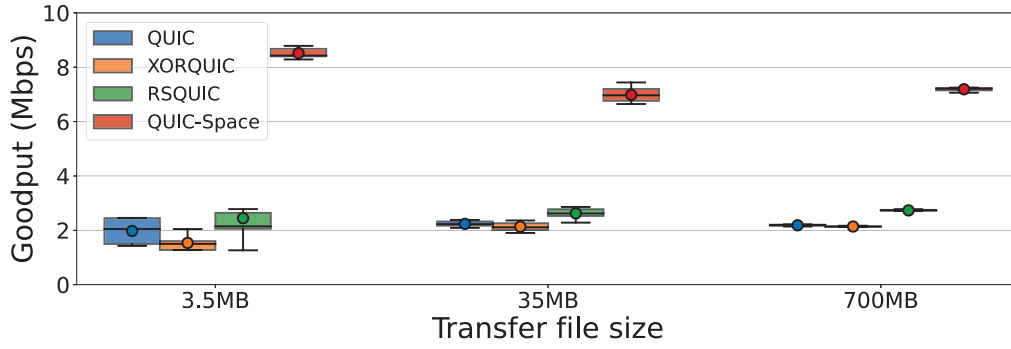


Figure C2 Goodput under a 5 MB flow control window.

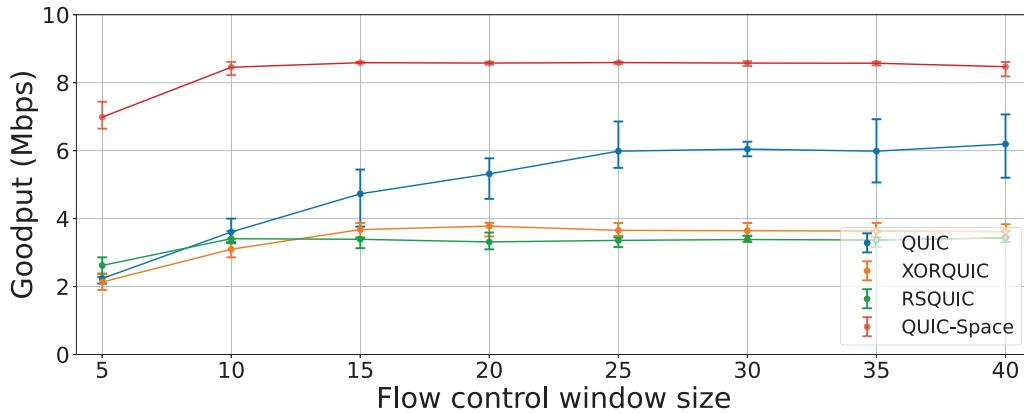


Figure C3 Impact of flow control window on goodput performance.

Additionally, Fig. C3 shows that once the flow control window satisfies the transmission demand, further increasing its size no longer improves goodput. Among the four protocols, QUIC-Space stabilizes its goodput at around a 10 MB flow control window, whereas XORQUIC and RSQUIC require at least 15 MB, and the original QUIC needs at least 25 MB. This implies that the original QUIC requires at least 25 MB of receive buffer space to accommodate out-of-order packets, which introduces additional reordering overhead. In contrast, QUIC-Space's faster FEC-based loss recovery significantly reduces buffer requirements, as confirmed in Fig. C4.

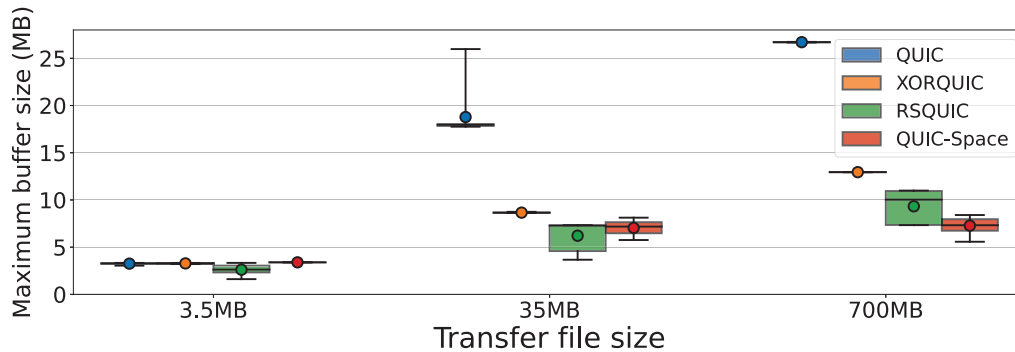


Figure C4 Maximum receive buffer size under a 40 MB flow control window.

Fig. C4 illustrates the maximum buffer size used when transmitting 3.5 MB, 35 MB, and 700 MB files under a 40 MB flow control window. The results indicate that all FEC-based schemes substantially reduce buffering requirements. However, due to the poor goodput of existing FEC schemes, they are not analyzed further. Compared to the original QUIC, QUIC-Space reduces buffer overhead by up to 68.5%. The buffer overhead of QUIC-Space primarily consists of symbols within the DW and partially ordered source symbols that have already been delivered. These source symbols need to be buffered for a while because future repair symbols may encode across them, making them necessary for decoding. The size of these source symbols is approximately equivalent to the BDP. Additionally, considering the symbols in DW and the delayed ACKs required to accommodate asymmetric bandwidth conditions, the overall buffer usage in practice may slightly exceed the BDP.

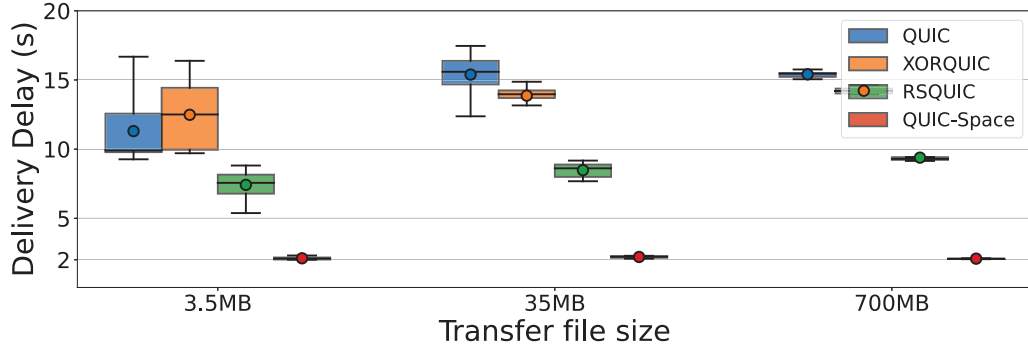


Figure C5 Delivery delay under a 40 MB flow control window.

Another key advantage of QUIC-Space is its low delivery delay. Fig. C5 illustrates the delivery delay of 3.5 MB, 35 MB, and 700 MB file transfers under a 40 MB flow control window. As shown in Fig. C5, QUIC-Space achieves near-optimal delivery delay, exceeding the one-way propagation delay only marginally. This also indicates that QUIC-Space's goodput is smoother than that of the other three protocols, resulting in more stable data delivery to the application. Specifically, for 3.5 MB, 35 MB, and 700 MB file transfers, QUIC-Space's delivery delay is 22.1%, 25.9%, and 28.4% of that of the best-performing alternative, RSQUIC, respectively. This advantage stems from its RTT-independent loss recovery mechanism. In contrast, retransmission-based recovery mechanisms introduce significantly higher delays, with lost packet recovery taking at least 1.5 times the RTT.

Table C1 Ratio of total transmitted bytes for three FEC-enhanced variants relative to original QUIC.

Protocol	3.5 MB	35 MB	700 MB
XORQUIC	1.043	1.041	1.042
RSQUIC	1.227	1.223	1.225
QUIC-Space	2.122	1.145	1.040

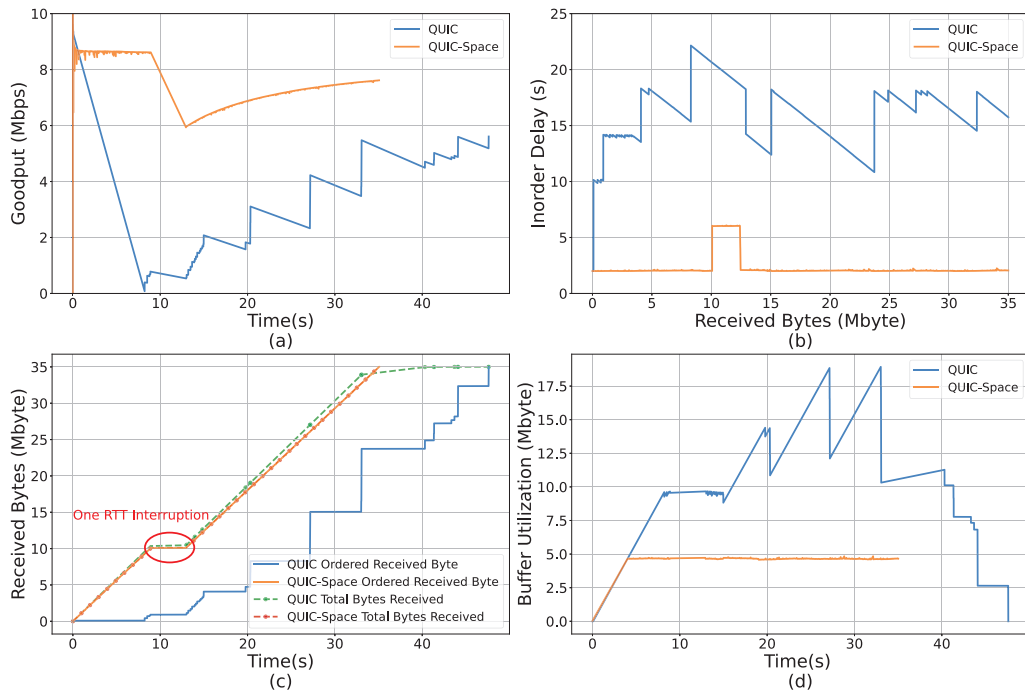


Figure C6 Performance of QUIC and QUIC-Space with 1-RTT interruption.

The primary trade-off for QUIC-Space's performance gains is energy consumption. Unlike terrestrial networks, deep space communication nodes operate with stringent energy constraints. Assuming a uniform energy cost per transmitted byte, total energy consumption depends on the total number of bits sent during transmission [8]. Table C1 presents the ratio of total transmitted data for the three FEC-enhanced QUIC variants and original QUIC. XORQUIC and RSQUIC

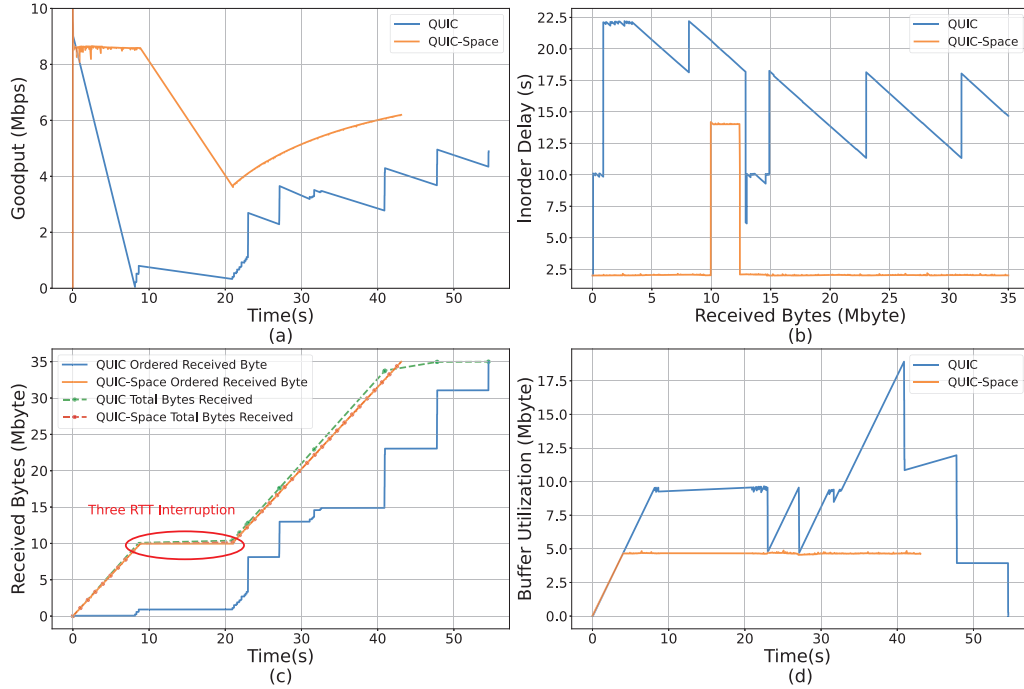


Figure C7 Performance of QUIC and QUIC-Space with 3-RTT interruption.

maintain a fixed ratio of total transmitted data to original QUIC because of their constant redundancy levels. In contrast, QUIC-Space continuously sends repair symbols while only awaiting ACKs, introducing a fixed energy overhead roughly equivalent to BDP. Consequently, for a 3.5 MB file transfer, QUIC-Space consumes more energy than the other protocols. As file size increases, this fixed overhead becomes negligible.

Fig. C6 and C7 present the performance of QUIC and QUIC-Space in transmitting a 35 MB file under network disruption scenarios of 1 RTT and 3 RTT durations, respectively, with a flow control window of 40 MB. For this analysis, we exclude XORQUIC and RSQUIC due to their suboptimal goodput performance. Our primary focus is on the 3 RTT disruption scenario. As shown in Fig. C7(a), the network interruption significantly reduces the goodput of both protocols. This reduction is further explained in Fig. C7(c), which compares the in-order received bytes with the total received bytes. The interruption introduces a time-shift in data reception, causing a noticeable delay in the total bytes received and thus degrading the overall goodput. Furthermore, Fig. C7(b) highlights the impact of the disruption on end-to-end in-order delivery delay. For QUIC-Space, the interruption directly increases the delivery delay of the corresponding data segment. In contrast, the impact on QUIC is less pronounced because QUIC's timeout-based retransmission mechanisms still operate during the disruption period. Consequently, the delay caused by the interruption overlaps with the head-of-line blocking delay, obscuring the direct effect of the disruption on the overall delivery time. Finally, Fig. C7(d) illustrates that, thanks to QUIC-Space's store-and-forward mechanism, the interruption does not lead to additional packet loss. This results in only a minimal impact on buffer occupancy.

Appendix D Conclusion

This letter proposes QUIC-Space, an FEC-enhanced QUIC protocol optimized for deep space networks. By replacing QUIC's retransmission-based loss recovery with streaming code and adaptive coding control, QUIC-Space decouples loss recovery from RTT, mitigating tail losses, head-of-line blocking, and buffer bloat. Experimental results indicate that QUIC-Space significantly improves goodput and reduces delivery delay. Our scheme presents a promising solution for deep space communications. For future work, we plan to incorporate more realistic channel models and consider improving TCP performance through Performance-Enhancing Proxies (PEPs).

References

- Sommer M, Sterz A, Vogelbacher M, et al. Quicrl: Disruption-tolerant networking via a quic convergence layer. *Computer Communications*, 2025, 241: 108247. URL <https://www.sciencedirect.com/science/article/pii/S014036642500204X>
- Michel F, De Coninck Q, Bonaventure O. Quic-fec: Bringing the benefits of forward erasure correction to quic. In: *2019 IFIP Networking Conference (IFIP Networking)*, 2019. 1–9
- Blanchet M. QUIC Profile for Deep Space. Internet-Draft draft-many-tiptop-quic-profile-00, IETF, 2025. URL <https://datatracker.ietf.org/doc/draft-many-tiptop-quic-profile/00/>
- Iyengar J, Swett I. QUIC Loss Detection and Congestion Control. RFC 9002, 2021. URL <https://www.rfc-editor.org/info/rfc9002>
- Yu J, Liu T, Li Y, et al. On the decoding cost of streaming forward erasure correction codes. *IEEE Communications Letters*, 2024, 28: 1760–1764
- Alessi N, Caini C, de Cola T, et al. Packet layer erasure coding in interplanetary links: The ltp erasure coding link service adapter. *IEEE Trans. Aerosp. Electron. Syst.*, 2020, 56: 403–414

- 7 Blanchet M, Eddy W, Li T. An Architecture for IP in Deep Space. Internet-Draft draft-many-deepspace-ip-architecture-01, Internet Engineering Task Force, 2024. URL <https://datatracker.ietf.org/doc/draft-many-deepspace-ip-architecture/01/>
- 8 Yang L, Fraire J A, Zhao K, et al. Optimizing deep-space dtn congestion control via deep reinforcement learning. *Comput. Networks*, 2024, 255: 110865