SCIENCE CHINA
Information Sciences

• LETTER •

December 2025, Vol. 68, Iss. 12, 229301:1–229301:2
https://doi.org/10.1007/s11432-025-4574-3

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

Jianhao YU[1], Ye LI[2], Wenfeng LI[1] & 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*

Received 16 April 2025/Revised 7 August 2025/Accepted 28 August 2025/Published online 24 October 2025

Citation Yu J H, Li Y, Li W F, et al. QUIC-Space: adaptive FEC-enhanced QUIC for reliable deep space communication. Sci China Inf Sci, 2025, 68(12): 229301, https://doi.org/10.1007/s11432-025-4574-3

Deep space communication is characterized by extremely long propagation delays, high packet loss rates, and intermittent connectivity. Recently, space agencies have proposed deploying Internet protocol (IP) networks on and around celestial bodies [1], spurring interest in assessing the feasibility of using the IP protocol stack in deep space networks [2]. Quick UDP Internet connections protocol (QUIC) [3] is a transport-layer protocol based on the user datagram protocol (UDP). Its architecture offers high scalability, making it more suitable than the transmission control protocol (TCP) in deep space networks. However, its loss recovery mechanism requires additional attention. The long round-trip times (RTT) significantly prolong QUIC's retransmission-based loss recovery, causing receive buffer bloat, increased delivery delay, inefficient bandwidth utilization, and exacerbated tail loss and head-of-line blocking.

To address these challenges, we propose QUIC-Space, an optimized extension of QUIC designed for deep space networks. QUIC-Space first adjusts QUIC's parameters to better suit the deep space environment. Second, it relies entirely on a packet-level forward erasure correction (FEC) code called streaming code (SC) [4] for loss recovery, thereby providing reliable transmission services without retransmissions. By adopting an adaptive coding control mechanism, QUIC-Space effectively decouples loss recovery time from RTT, mitigating the adverse impacts of deep space environments. The main contributions of this work are as follows. (1) A detailed analysis of QUIC's limitations when operating in deep-space networks. (2) The design and implementation of QUIC-Space, developed to overcome these identified limitations. (3) An analytical model that derives upper bounds on goodput for both QUIC and QUIC-Space. (4) A comprehensive simulation study conducted in an Earth-Moon network, demonstrating that, with a modest increase in energy consumption, QUIC-Space achieves significantly higher goodput, reduced delivery delay, and relieved buffer bloat.

QUIC-Space system model. Common packet-level FEC schemes like Reed-Solomon codes and random linear network codes still require retransmissions whenever the number of lost symbols in an encoding block exceeds their repair capability. To overcome this, we adopt a sliding window FEC algorithm called SC. By employing a sliding window to encode all unacknowledged source symbols, SC can rely on repair symbols for data recovery even when the loss rate temporarily exceeds the recovery capability. Let S_i denote the source symbol with sequence index (ID) i, and let i_{seq} be the index of the last transmitted source symbol (initially $i_{\text{seq}} = -1$). Define w_s as the ID of the oldest unacknowledged source symbol and update upon the arrival of ACKs. Define $w_e \equiv i_{\text{seq}}$, and the interval $[w_s, w_e]$ is referred to as the encoding window (EW) of the current encoder. Each repair symbol R_k for $k = 0, 1, \ldots$ is a random linear combination of all unacknowledged source symbols in the EW: $R_k = \sum_{i=w_s}^{i_{\text{seq}}} g_{k,i} S_i$. Here the encoding coefficients $g_{k,i}$ are randomly selected from a finite field $\mathbb{F}_{2^m}$ of size 2^m, typically generated by a pseudo-random number generator (PRNG). The vector $[g_{k,w_s}, \ldots, g_{k,i_{\text{seq}}}]$ is referred to as the encoding vector (EV) of the repair symbol. The source symbol can be considered as a special repair symbol with EV = [1]. In practice, the repair symbol only needs to carry its ID and EW information, as the encoding vector can be retrieved via the PRNG based on an agreed seed. Since each repair symbol contains information about all unacknowledged source symbols, the decoder can recover symbols even if the number of lost packets briefly exceeds its repair capacity.

At the receiver, let i_{ord} denote the ID of the last in-order received source symbol. In-order transmission is interrupted if the next received symbol is neither the source symbol $S_{i_{\text{ord}}+1}$ nor a repair symbol with $w_e \leqslant i_{\text{ord}} + 1$. In this scenario, the decoder is activated, and incoming symbols are buffered for recovery. We define the decoding window (DW) as the interval $[\hat{w}_s, \hat{w}_e]$, where $\hat{w}_s \triangleq i_{\text{ord}} + 1$ and $\hat{w}_e$ denotes the largest EW endpoint observed among received repair symbols. As more symbols are received, $\hat{w}_e$ may grow accordingly. The decoder attempts recovery using an on-the-fly Gaussian elimination algorithm [5]. The system becomes solvable, enabling full recovery, once the number of linearly independent repair symbols received within the DW equals the number of missing source symbols. Subsequently, in-order delivery resumes, and i_{ord} is updated to

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

$\hat{w}_e$. Additionally, the decoder can periodically feed back i_{ord} to the sender, allowing dynamic EW adjustment and reducing redundant coding overhead.

Adaptive coding control. When packet transmission opportunities arise, adaptive coding control determines which symbols to send. Due to QUIC's stream multiplexing mechanism, an open connection may have multiple open or closed streams. Adaptive coding control applies different symbol scheduling strategies depending on the connection state. When one or more streams are transmitting data, repair symbols are sent based on the packet loss rate p_e, which is estimated at the receiver and fed back to the sender via the reverse path, as described later. At the sender, the coding controller maintains counts of the transmitted source and repair symbols, denoted as N_S and N_R, respectively. A repair symbol is transmitted whenever the current redundancy falls below $p_e + \delta$, where $\delta \in (0, 1)$ is an additional redundancy compensation factor. Note that δ is a small value intended to proactively counteract inaccurate or unstable packet loss rate estimations. In memoryless random loss scenarios, the decoding delay is inversely proportional to $1/\delta$ [5].

When no new data is to be sent but at least one stream awaits ACKs, N_S remains unchanged as no new source symbols are transmitted. To ensure reliability, the coding controller continuously sends repair symbols whenever QUIC permits packet transmission until all source symbols are acknowledged. Once all source symbols are confirmed, the repair symbol transmission stops until either new data becomes available or the connection is terminated.

Implementation in QUIC. Before detailing our SC integration, a brief review is presented. In QUIC, each packet consists of a series of frames, with each frame serving a specific function. The QUIC design allows the definition of custom frame types to extend its functionality [3]. Leveraging this extensibility, three new frame types have been introduced to integrate SC into QUIC: the `SOURCE_SYMBOL`, `REPAIR_SYMBOL`, and `SC_ACK` frames. The `SOURCE_SYMBOL` frame encapsulates source data requiring encoding protection. Specifically, it includes the following fields. (1) **Type** (1 byte): indicating the frame type. (2) **Source ID** (8 bytes). (3) **Length** (2 bytes): representing the valid payload length. (4) **Payload**: in which all frames requiring protection are padded to a fixed size and encapsulated. This payload, together with the Length field, is encoded into source symbols by the SC encoder. Notably, padding is introduced solely to satisfy encoding requirements and does not increase the packet's effective payload size. Compared to an unmodified QUIC packet, a `SOURCE_SYMBOL` frame incurs only an additional 11 bytes of overhead. The `REPAIR_SYMBOL` frame carries repair symbols and includes the following fields. (1) **Type** (1 byte). (2) **Repair ID** (8 bytes). (3) **EW** (16 bytes): denoting w_s and w_e. (4) **Repair symbols** generated by the encoder. Each `REPAIR_SYMBOL` frame occupies a separate packet and incurs an additional bandwidth overhead of $p_e + \delta$. The `SC_ACK` frame consists of the fields: **Type** (1 byte), i_{ord} (8 bytes), and p_e (8 bytes), which provide feedback to the sender. The QUIC-Space process is illustrated in Figure 1.

Expected goodput for QUIC and QUIC-Space. A discrete-time transmission system characterized by bandwidth BW, round-trip time RTT, and an independent packet-loss rate e is considered to model the theoretical upper bounds on goodput for both QUIC and QUIC-Space. The expected goodput of each protocol is defined as the ratio of the file size to its total transmission time. The expected total transmission times for QUIC, denoted $E\{T_{\text{total}}^{\text{QUIC}}\}$, and for QUIC-Space, denoted $E\{T_{\text{total}}^{\text{QUIC-Space}}\}$, are given by

$$E\{T_{\text{total}}^{\text{QUIC}}\} = \frac{\frac{S}{1-e} - e \cdot \min(S, \text{BDP})}{\text{BW}} + E\{M\} \times \text{RTT} - \left(\frac{n - E\{K\}}{\text{BW}}\right), \tag{1}$$

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

The definitions of S, $E\{M\}$, $E\{K\}$, n, and ρ, along with the proofs of (1) and (2), are provided in Appendix B.

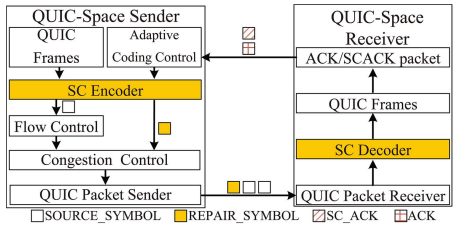

Figure 1 (Color online) Encoding/decoding of QUIC-Space.

Simulation. In an Earth-Moon scenario, QUIC-Space outperforms both original QUIC and the interleaved XOR/RS schemes of the QUIC-FEC framework. Simulation results show that QUIC-Space boosts goodput by up to 348.3%, reduces delivery delay by 77.9% and cuts buffer overhead by 68.5% versus original QUIC, and delivers up to 266.4% higher goodput, 71.6% lower delay and comparable energy consumption versus the QUIC-FEC baseline. Detailed simulation results are provided in Appendix C.

Conclusion. This work introduced QUIC-Space, an FEC-enhanced QUIC protocol for deep space networks. By replacing retransmissions with adaptive streaming codes, QUIC-Space decouples loss recovery from RTT, thereby mitigating tail losses, head-of-line blocking, and buffer bloat. Our experimental results confirmed that QUIC-Space significantly enhances goodput and reduces delivery delay.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 62131012) and Unveiling and Leading Project of Nanjing University Integrated Research and Development Platform of Ministry of Education. Part of Jianhao YU's work was conducted at Nantong University.

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 LCAWG. The future lunar communications architecture. Technical Report, IOAG, 2020. https://www.ioag.org/Public%20Documents/Lunar%20communications%20architecture%20study%20report%20FINAL%20v1.3.pdf
2 Blanchet M, Eddy W, Li T. An architecture for IP in deep space. Internet-Draft draft-manydeepspace-ip-architecture-01, Internet Engineering Task Force. 2024. https://datatracker.ietf.org/doc/draft-many-deepspace-ip-architecture/01/
3 Iyengar J, Thomson M. QUIC: a UDP-based multiplexed and secure transport. RFC 9000, 2021. https://www.rfc-editor.org/info/rfc9000
4 Yu J, Pan S, Gao R, et al. Low-delay transmission for non-terrestrial networks based on FEC and reinforcement learning. In: Proceedings of IEEE/CIC International Conference on Communications in China (ICCC), 2023. 1–6
5 Li Y, Chen L, Su L, et al. PEPesc: a TCP performance enhancing proxy for non-terrestrial networks. IEEE Trans Mobile Comput, 2024, 23: 3060–3076