A QUIC-Enabled Reliable Video Transmission Scheme in Ultra-Dense LEO Satellite Networks

Mengyang Zhang*, Ting Ma*, Zitian Zhang†, Haibo Zhou*, Lian Zhao‡
*School of Electronic Science and Engineering, Nanjing University, Nanjing, China, 210023.
†School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou, China, 310018.
‡Department of Electrical, Computer, Biomedical Engineering,
Toronto Metropolitan University, Toronto, ON M5B 2K3, Canada.
Email: Kolimn_Zhang@163.com, majiawan27@163.com,
zitian.zhang@mail.zjgsu.edu.cn, haibozhou@nju.edu.cn, l5zhao@torontomu.ca

2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall) | 979-8-3503-2928-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/VTC2023-Fall60731.2023.10333594

***Abstract*—The Ultra-Dense LEO Satellite Networks (UDLSN) has immense potential to provide low-latency and high-reliability services in future communication networks, owing to its global coverage, high capacity and reliable connectivity. However, the LEO networks usually suffer relatively high and variable transmission errors due to multipath, shadowing and handover. For delay-constrained video transmission, existing packet protection mechanisms frequently violate the constraint and degrade quality in such environments. In this paper, we propose a QUIC-Enabled Reliable Video Transmission Scheme (QRVTS) with adaptive Forward Error Correction (FEC) to reduce loss recovery time and enhance transmission performance especially for delay-constrained videos. Specifically, QRVTS incorporates an adaptive mechanism to dynamically adjust FEC redundancy based on the prevailing channel loss conditions and frame types. We evaluate our mechanism under multiple satellite scenarios with different network characteristics. The simulation shows significant gains in overview completion time and frame-level delivery delay for delay-constrained video transmission in LEO satellite scenarios.**

***Index Terms*—Ultra-Dense LEO Satellite Networks, QUIC, forward error correction, adaptive, delay-constrained, priority**

I. Introduction

The development of Ultra-Dense LEO Satellite Networks (UDLSN) have advanced the field of satellite communications. Compared to traditional geostationary satellites, LEO satellites operate at lower altitudes, offering global coverage [1] and high backhaul capacity [2]. UDLSN represent a promising technology for seamless global connectivity and advanced applications to meet the requirements of massive communication and personalized user experience in 6G [3]. With the increasing volume of user data, the UDLSN are being envisioned to provide low-latency and highly reliable communications. However, the network faces challenges particularly in terms of relatively high and variable transmission errors due to multipath, shadowing and frequent handover, which significantly impacts the transmission performance especially for delay-constrained videos [4] in LEO constellations.

QUIC is a protocol designed to reduce handshake overhead and transport delay primarily [5]. QUIC incorporates the features of TCP, TLS and HTTP/2 on top of UDP protocol offering the benefits of reliable data transmission, secure communication and efficient multiplexing [6]. While the technology of retransmission in QUIC protocol is hard to protect packets in harsh network environments especially for delay-constrained video transmission [7]. Forward Error Correction (FEC) is an effective technique to address the challenge of packet transmission reliability [8]. Google recognizes the importance of FEC within QUIC protocol and introduced a native XOR-based coding in early work [9]. Pablo et al. integrates FEC into QUIC protocol with multiple alternative coding frameworks providing excellent results [10]. To optimize overhead, rQUIC [11] proposes an adaptive mechanism that adjusts FEC ratio based on loss information feedback. This method is designed for bulk transfer and web browsing and uses a progressive adjustment approach with XOR-based coding. Additionally, FLEC [12] implements a flexible FEC framework into QUIC protocol to meet the delay requirements for different applications.

Existing works has demonstrated the significant importance to implement FEC into QUIC with a flexible way. In spite of these endeavors, delay-constrained video transmission in hash packet loss environments (such as satellite networks) still face a deficiency in a resilient mechanism for handling losses. Furthermore, it is imperative to take into account priority-aware protection measures in order to enhance the quality of service [13]. In this paper, we propose an adaptive FEC mechanism which improves packet recovery time to meet the requirements of delay-constrained video transmission and prioritize frame priority to ensure better video quality. Our main contributions are summarized as:

- We propose a flexible QRVTS to enhance the transmission performance for delay-constrained videos, which involves an adaptive FEC mechanism to effectively handle variable loss conditions in UDLSN.
- We focus on the frames importance and introduce additional redundancy to ensure timely arrival for high-priority frames, while the FEC overhead of other frames is reduced to maintain optimal overall throughput.
- We evaluate the advantages of QRVTS in diverse satellite network scenarios by simulating in different bandwidth, one-way delay and loss rate network conditions.We compare the performance of QRVTS with QUIC in terms of completion

979-8-3503-2928-5/23/$31.00 ©2023 IEEE

time, frame delivery time and other relevant factors in the context of delay-constrained video transmission

The remaining of this paper is organized as follows: Section II describes the framework and principles of QRVTS. Section III presents the evaluation of our QRVTS framework in multiple satellite scenarios. Section IV concludes our work and outlooks the future development.

II. Design and Implementation: A QUIC-Enabled Reliable Video Transmission Scheme

Figure 1 shows the system model of UDLSN. The proposed QRVTS is a flexible mechanism designed to reduce loss recovery time in such scenarios. Figure 1 also shows the generic framework of our mechanism. This framework mainly includes the modules of loss monitor, frames check and FEC calculating. When a video frame is sent to the send buffer, the frame type is detected first. Then the loss rate associated with the frame type is determined by monitor module. Last, the framework calculte FEC value leveraging the loss rate and the frame type. This proposed framework considers two crucial aspects for FEC value calculation: the delay-constrained features and various frame types. By taking these factors into account, QRVTS is expected to meet specific time requirements and enhance video quality in LEO networks.

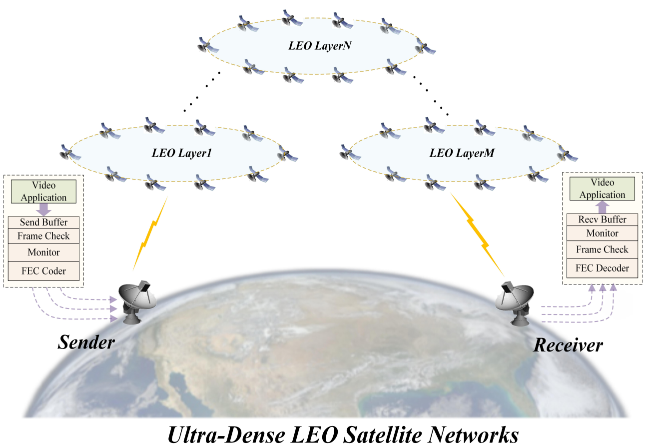


Fig. 1. UDLSN System Model.

A. OverView

The key component of QRVTS is the adaptive and priority-aware FEC generating mechanism. The FEC mechanism generates repair symbols based on packet loss rate and frame types. Specifically, in conditions of higher loss rate and with important frames, the required FEC redundancy is increased. To further optimize the FEC redundancy allocation, we proposed the use of a controller denoted by $target\_rate$ which determines the expected residual loss rate after FEC protection. Residual loss rate is a crucial factor for delay-constrained video transmission because the packet loss rate is not stable and vary significantly in UDLSN. By adjusting $target\_rate$, the FEC redundancy can be dynamically optimized to achieve the desired level of protection while minimizing overhead and enhancing the video quality.

B. Packet Loss Process

We utilize the Bernoulli loss model and Gilbert loss model to describe the packet loss process. The Bernoulli loss model is a statistical model used to describe binary classification problems. It assumes that each loss can be viewed as a random variable independently sampled from a Bernoulli distribution. In this model, the packet loss rate is denoted as p and each packet event is assumed to be independent. The probability of observing l losses in n packets can be calculated as

$$Pr(N_{lost} = l) = \binom{n}{l} p^l (1-p)^{n-l}. \tag{1}$$

In the context of a FEC block consisting of k source symbols and $n-k$ repair symbols, the probability of experiencing more than $n-k$ packets lost can be obtained as

$$Pr(N_{lost} > n-k) = \sum_{l=n-k+1}^{n} \binom{n}{l} p^l (1-p)^{n-l}. \tag{2}$$

Consequently, the average residual loss rate, denoted as $PLR_{RES}(p)$, is derived by

$$PLR_{RES}(p) = \frac{1}{n} \sum_{l=n-k+1}^{n} l \binom{n}{l} p^l (1-p)^{n-l}. \tag{3}$$

The Gilbert model [14] as described in Figure 2 is a two-state Markov chain used to characterize burst-loss conditions. In this model, a good state represents a successful packet reception, while a bad state represents a packet loss. Let p

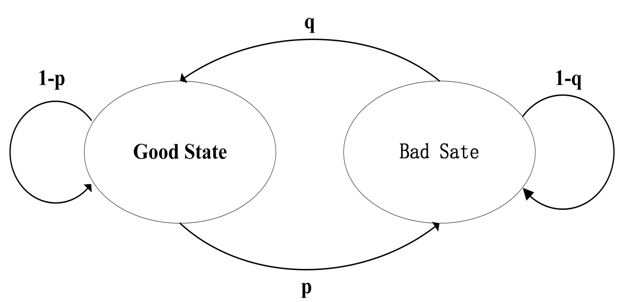


Fig. 2. Glibert Loss Model.

denotes the probability from good to bad and q denotes the probability from bad to good, a sample trace estimation for p and q are [15]:

$$\hat{p} = \frac{n_{01}}{n_0}, \tag{4}$$

$$\hat{q} = \frac{n_{10}}{n_1}, \tag{5}$$

where n_{01} is the number of times that bad state after good state consecutively and n_{10} is the number of times that good state after bad state in the trace window. The average burst length is obtained by

$$average\_length = \frac{1}{q}. \tag{6}$$

The Gilbert model presents complexities in computing the average residual loss rate, which results in increased computation delay. To simplify the process, we adopt Equation (3) to estimate the residual loss rate. Additionally, we only employ the Bernoulli loss model in the initial version to showcase the advantages of this framework in the following Section III. However, it is important to notice that this framework can be easily extended to incorporate a burst loss model just as the description in the following Algorithm 1.

C. Adaptive and Priority-Aware FEC Redundancy Generating

For highly lossy networks, such as satellite constellations, it becomes crucial to minimize the delivery time for delay-constrained video frames without significantly compromising the goodput. This objective forms the core principles of the FEC algorithm, which dynamically adjusts the redundancy based on the prevailing lossy conditions and provide optimal protection for important frames. Algorithm 1 shows the generic idea of this framework. And the algorithm parameters used are defined in TABLE I.

To implement an adaptive FEC mechanism, we follow a controller called $target\_rate$ to determine FEC redundancy. More than a fixed value, the $target\_rate$ is more flexible which varies with the frame type and loss conditions. When a new packet arrives at sender buffer, QRVTS selects the (n, k) value which brings the lowest overhead while ensuring the residual loss rate calculated by Equation (3) remains below $target\_rate$. Choosing $n \leq 9$ allows for shorter FEC blocks, which can promptly respond to packet loss and maintain frame integrity better to satisfy the requirements of delay and it's experimental solution. If the stream data is classified as of high priority, an additional step is taken to ensure the timely and reliable delivery. Then the value of (n, k) is recalculated with lower $target\_rate$ and "high priority uniform loss rate" which is denoted as p_h. This adjustment is made specifically for high-priority streams, allowing for a more stringent FEC redundancy selection. It ensures that high-priority streams receive enhanced protection and are delivered within the specified time constraint more possibly. To prevent frequent FEC parameters oscillations, we update (n, k) every two RTTs, this interval provides a balance between responsiveness to changing network conditions and stability in parameter updating. When encountering burst packet losses, we use Gilbert loss model as described in Section II.B. The redundancy is adjusted to a larger value than the burst length while remaining the same FEC ratio.

D. QRVTS Implementation

We implement QRVTS in an open source version of QUIC called PQUIC [16]. The implementation of PQUIC consists of a collection of protocol operations that can be enhanced or substituted by protocol plugins. It is easy to plug flexible FEC implementation in PQUIC framework. To better address the loss problem, we consider Random Linear Code (RLC) based on sliding-window which can generate multiple repair symbols from the same set of source symbols, allowing for recovering multiple packets within one FEC window. In the following evaluation, RLC is used in our framework with the window size less than 10 symbols to achieve faster recovery and we set $target\_rate$ to $\frac{loss\_rate}{2}$ for generic frames and $\frac{loss\_rate}{4}$ for important frames.

TABLE I
DEFINITION OF ALGORITHM PARAMETERS

p, p_h	frames loss rate and high-priority frames loss rate
μ_l, μ_h	$target\_rate$ for low and high-priority frames
$priority$	the frame priority
$CalResLoss(n, k, p)$	residual loss rate calculated by Equation (3)
$CheckUpdate()$	check whether to update FEC value

III. EVALUATION

A. Experiment Design

For evaluation, we utilize Mininet tool [17] to construct the network environment and simulate satellite scenarios. This tool provides flexibility in adjusting network loss rate and introducing random loss behaviors. Our implementation of QRVTS is built upon the PQUIC, which offers flexible plugins to achieve the FEC framework. Furthermore, the video is conducted by the H.264/AVC standard reference software JM 19.0 [18]. Video frames are encoded at 30 fps with a GOP consisting of IPBPB frames. Each frame is transmitted by a separate QUIC stream, where the streams with I-frame are classified of high priority and others are classified of low priority.

To comprehensively evaluate our framework, we have selected a methodology that involves conducting experiments across a range of variable network parameters [19]. To ensure comprehensive testing, we conduct each set of experiments using 100 different parameter combinations. Specifically, we choose one-way delay, loss rate and bandwidth as the variable parameters, the specific values range are given by Section III.B. We choose three metrics to evaluate our framework: Download Completion Time (DCT), frames delivery time and frames on-time ratio. DCT refers to the total transmission time of the video stream. In the presence of high and indefinite packet losses, using less FEC overhead cannot recover any loss but increase the download time. So adaptive FEC should be considered to bring less overhead and promise the overview completion time. Furthermore, we also prioritize frame-level delivery to meet delay constraints and ensure video quality. In terms of delivery time, we evaluate the effectiveness of frame-level FEC protection. We record the delay between when the frame is sent to the buffer and when it arrives at the peer. For the on-time ratio, high-priority frames should have a higher probability to be transmitted on time in our experiments.

Algorithm 1 FEC Symbols Generating Algorithm

Require: $\mu_l, \mu_h, p, p_h, q, priority$
Require: $CalResLoss(n, k, p)$, $CheckUpdate()$
Init $n_l = 2$; $k_l = 1$;$RSL_{result\_l} = 0$
while $2 \le n_l \le 9$ and $1 \le k_l \le n_l - 1$ **do**
  $RSL = CalResLoss(n_l, k_l, p)$
  if $RSL \le \mu_l$ and $RSL_{result\_l} < RSL$ **then**
    $RSL_{result\_l} = RSL, n_{temp} = n_l, k_{temp} = k_l$
  end if
end while
$n_l = n_{temp}, k_l = k_{temp}$
if $priority = high$ **then**
  Init $n_h = 2$; $k_h = 1$;$RSL_{result\_h} = 0$
  while $2 \le n_h \le 9$ and $1 \le k_h \le n_l - 1$ **do**
    $RSL = CalResLoss(n_h, k_h, p_h)$
    if $RSL \le \mu_h$ and $RSL_{result\_h} < RSL$ **then**
      $RSL_{result\_h} = RSL$
      $n_{temp} = n_h, k_{temp} = k_h$
    end if
  end while
  $n_h = n_{temp}, k_h = k_{temp}$
  if $CheckUpate\_h() = True$ **then**
    if $\frac{n_h}{k_h} < \frac{n_l}{k_l}$ **then**
      $n = n_l, k = k_l$
    else
      $n = n_h, k = k_h$
    end if
  else
    $n = n_{lasth}, k = k_{lasth}$
  end if
else
  if $CheckUpdate() = True$ **then**
    $n = n_l, k = k_l$
  else
    $n = n_{last}, k = k_{last}$
  end if
end if
if $Gilbertlossmodel$ **then**
  if $\frac{1}{q} > n - k$ **then**
    $k = k \times \frac{\frac{1}{q}}{n-k}, n = k + \frac{1}{q}$
  end if
end if

B. Performance and Results

In this section, we evaluate the performance of QRVTS following the principles described in Section III.A. First, we compare DCT between QRVTS and QUIC. Figure 3 displays the Cumulative Distribution Function (CDF) of DCT, where each curve represents the distribution with different one-way delay of QRVTS or QUIC. The loss rate 3-8% is to account for variable packet loss, including heavier losses and lighter losses. One way delay is 70-200ms because the satellite constellation is dynamic and may introduce higher delay. As shown in Figure 3, in most cases QRVTS performs a quicker action compared with QUIC. QRVTS generally achieves faster download time thanks to its ability to reduce packet loss recovery time. When with larger Round-Trip Times (RTTs), a single lost packet can take at least one more RTT to retransmit from the sender, leading to increased delay. When multiple packets are lost during transmission, FEC can enable faster downloads compared to QUIC. With an adaptive FEC mechanism, this framework will not bring extra redundancy and more delay. The average download completion time for QUIC is 52853.38ms and for QRVTS is 40917.29ms, where QRVTS improve 29.17% on the average.

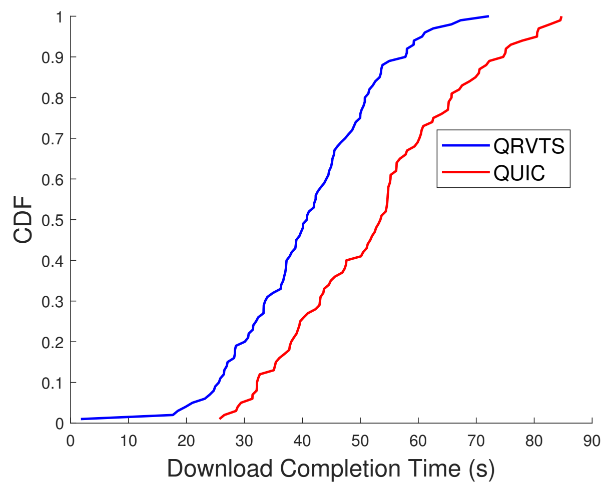
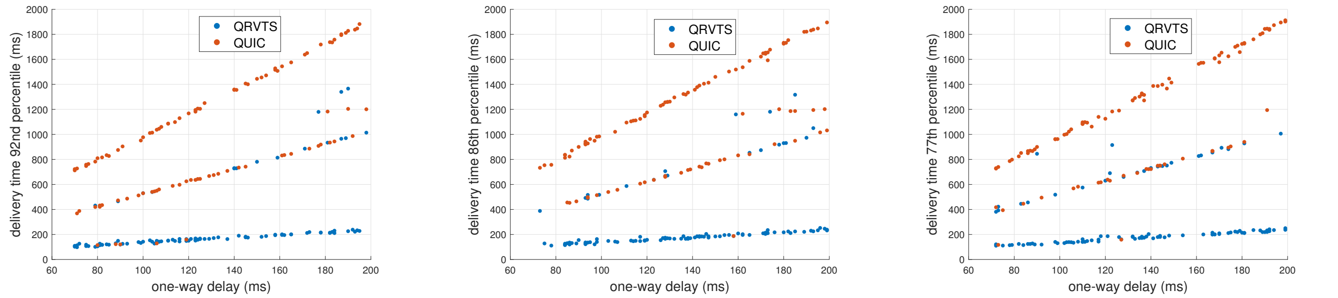


Fig. 3. Download Completion Time comparing QRVTS and QUIC ($bandwidth \in [3, 8]Mbps,\ loss\ rate \in [3, 8]\%,\ RTT \in [140, 400]ms$).

Figure 4 shows the frame-level delivery time in different conditions with 3%, 5% and 8% loss rate. In the three cases, the 92nd, 86th and 77th percentiles of video frames are all almost successfully delivered before the deadline which is set to be 300 milliseconds. However, with QUIC protocol only a few video frames are successfully delivered before deadline for the same percentiles in smaller RTTs and almost every video frame arrives later than deadline in larger RTTs. In QUIC protocol, lost packets will take more than two RTTs to be delivered considering loss detection mechanism. In the best case, lost packets have the possibility of being successfully delivered just before the deadline. However, in the presence of transmission delays lost packets are more prone to experiencing additional delays and arrive later than deadline especially in heavy lossy conditions. So we can see in larger RTTs and highly lossy conditions, the delivery time of QUIC can exceed 1500ms while those of QRVTS remains below 300ms for most frames. In fact, the number of frames delivered successfully within QUIC protocol is significantly lower than the percentiles shown in Figure 4. Besides, it's important to notice that we use different percentiles in the three cases. Actually, QRVTS cannot provide similar performance in different situations. That is because, for example, the loss rate of 10% does not necessarily mean that one packet is lost out of every ten packets sent, which is essentially an average

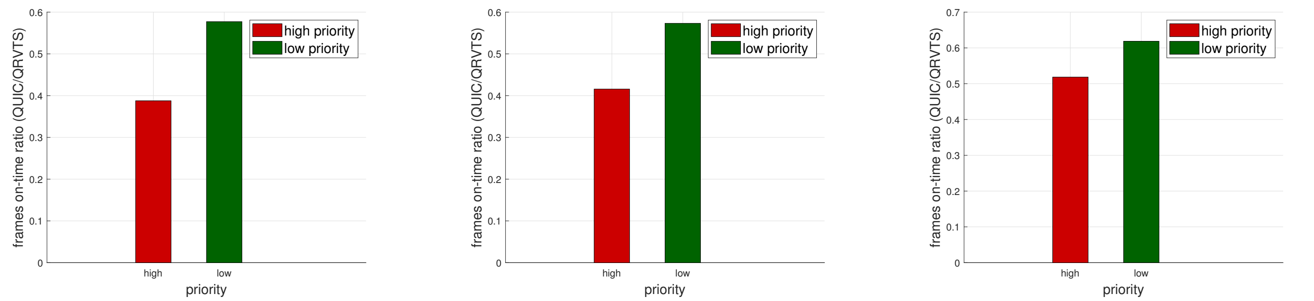


(a) Frames delivery time in 3% loss rate conditions comparing QRVTS with QUIC ($bandwidth = 8Mbps,\ loss\ rate = 3\%$).

(b) Frams delivery time in 5% loss rate conditions comparing QRVTS with QUIC ($bandwidth = 8Mbps,\ loss\ rate = 5\%$).

(c) Frams delivery time in 8% loss rate conditions comparing QRVTS with QUIC ($bandwidth = 8Mbps,\ loss\ rate = 8\%$).

Fig. 4. Frames Delivery Time.



(a) Frames on-time ratio in 3% loss rate conditions comparing QUIC with QRVTS ($bandwidth = 8Mbps,\ loss\ rate = 3\%,\ deadline = 300ms$).

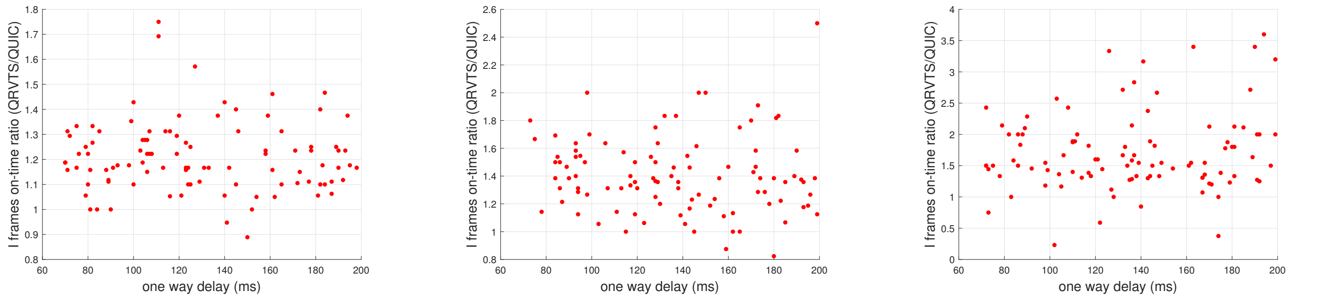
(b) Frames on-time ratio in 5% loss rate conditions comparing QUIC with QRVTS ($bandwidth = 8Mbps,\ loss\ rate = 5\%,\ deadline = 300ms$).

(c) Frames on-time ratio in 8% loss rate conditions comparing QUIC with QRVTS ($bandwidth = 8Mbps,\ loss\ rate = 8\%,\ deadline = 300ms$).

Fig. 5. Frames On-Time Ratio.

statistical value. It's really normal to see more packets lost than one every ten packets due to the random possibility even if the loss rate is 10%. Therefore, we cannot guarantee that the FEC redundancy is always greater than the number of lost packets in every round. In cases where frames cannot be recovered, they will need to be retransmitted and result in additional bandwidth overhead. And it is more likely to occur when the loss rate is relative high, which is a probabilistic issue. So we can see as the loss rate increases, fewer frames (92nd for 3%, 86th for 5%, 77th for 8%) can be delivered on time in QRVTS. Nevertheless, it still performs much better than QUIC protocol.

Figure 5 shows the on-time ratio for different types of frames which means the rate of frames arriving before deadlines. The deadline is set to be 300 ms and the loss rate is to be 3%, 5% and 8% respectively. In our experiments, we prioritize I frames over other frames as a measure to mitigate significant quality degradation caused by I frames loss. We apply protection mechanisms accordingly to safeguard different frames in transmission process. As depicted in Figure 5, we observe the noticeable improvement in delivery time for both I frames and non-I frames, regardless of the lossy conditions. This outcome aligns with results in Figure 4. Additionally, it is worth noticing that the on-time ratio of I frames is higher than that of other frames in all three cases about 30% average improvement. This improvement can be attributed to the lower $target\_rates$ set for I frames. By setting a lower $target\_rates$, we allocate more redundancy to account for unexpected packet losses. This redundancy helps ensure a greater proportion of I frames to be delivered on time, resulting in a higher on-time ratio. We adopt the principles of adaptive FEC, which explains why we don't set the same $target\_rates$ for all frames. Setting an excessively lower $target\_rates$ can introduce additional overhead, leading to increased queuing delay for subsequent frames. In turn, it degrades the overall completion time. Therefore, considering the trade-off between overhead and delay, we tailor the $target\_rates$ to adhere to the principles of adaptive FEC and allocate more redundancy just for I frames. Figure 6 displays the on-time ratio of I-frames for different scenarios. Across all three cases, the on-time ratio of I-frames in QRVTS under different one-way delay and loss rates are consistently better than QUIC. This indicates that QRVTS has a more robust and reliable performance for I-frames transmission, especially under harsh network conditions. The above results demonstrate the potential of QRVTS as an effective transport protocol for delay-constrained video transmission that require low latency and high reliability.



(a) I frames on-time ratio in 3% loss rate comparing QRVTS with QUIC ($bandwidth = 8Mbps$, $loss\ rate = 3\%$, $deadline = 300ms$).

(b) I frames on-time ratio in 5% loss rate comparing QRVTS with QUIC ($bandwidth = 8Mbps$, $loss\ rate = 3\%$, $deadline = 300ms$).

(c) I frames on-time ratio in 8% loss rate comparing QRVTS with QUIC ($bandwidth = 8Mbps$, $loss\ rate = 3\%$, $deadline = 300ms$).

Fig. 6. I Frames On-Time Ratio.

IV. Conclusion

In Ultra-Dense LEO Satellite Networks (UDLSN), variable transmission errors pose significant challenges that impact the performance of delay-constrained video transmission seriously. To address these challenges, we have incorporated an adaptive mechanism using FEC in QUIC protocol called QRVTS. This framework aims to reduce the packet recovery time, accelerate the delivery of frames before deadline and prioritize the protection for high-priority frames to ensure optimal video quality. We have implemented this framework by PQUIC. Our results indicate that QRVTS outperforms QUIC in different satellite scenarios and demonstrate superior performance in the entire transmission process and frame-level delivery time. Furthermore, QRVTS provides better protection for I frames and ensure their on-time arrival and optimal video quality. In the future, we will further optimize redundancy usage by leveraging on the multipath QUIC and ensure optimal goodput for other user-cases.

V. Acknowledgment

This work was supported in part by the National Key &D Program of China under Grant 2020YFB1806104, in part by Natural Science Foundation of Jiangsu Province Youth Project (BK20180329) and in part by Innovation and Entrepreneurship High-level Team Program of Jiangsu Province under Grant JSSCTD20220.

References

[1] T. Ma, B. Qian, X. Qin, X. Zhang, N. Cheng, and H. Zhou, "Joint subchannel allocation and beamforming for multicast in ultra-dense leo backbone network," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 1625–1630.

[2] N. U. Hassan, C. Huang, C. Yuen, A. Ahmad, and Y. Zhang, "Dense small satellite networks for modern terrestrial communication systems: Benefits, infrastructure, and technologies," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 96–103, 2020.

[3] J. Zhao, Q. Yu, B. Qian, K. Yu, Y. Xu, H. Zhou, and X. Shen, "Fully-decoupled radio access networks: A resilient uplink base stations cooperative reception framework," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.

[4] S. Soltani, K. Misra, and H. Radha, "Delay constraint error control protocol for real-time video communication," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 742–751, 2009.

[5] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. B. Krasic, C. Shi, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, J. Bailey, J. C. Dorfman, J. Roskind, J. Kulik, P. G. Westin, R. Tenneti, R. Shade, R. Hamilton, V. Vasiliev, and W.-T. Chang, in *The QUIC Transport Protocol: Design and Internet-Scale Deployment*, 2017.

[6] L. Thomas, E. Dubois, K. Nicolas, and E. Lochin, "Google QUIC performance over a public SATCOM access," *International Journal of Satellite Communications and Networking*, vol. 37, no. 6, pp. 601–611, Nov. 2019. [Online]. Available: https://hal.science/hal-02401946

[7] J. Wu, C. Yuen, N.-M. Cheung, J. Chen, and C. W. Chen, "Streaming mobile cloud gaming video over tcp with adaptive source–fec coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 1, pp. 32–48, 2017.

[8] L. Vicisano, M. Watson, and M. Luby, "Forward Error Correction (FEC) Building Block," RFC 5052, Aug. 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc5052

[9] I. Swett, "Quic wg charter: Fec initially out of scope," in *document IETF99*, 2017.

[10] F. Michel, Q. De Coninck, and O. Bonaventure, "Quic-fec: Bringing the benefits of forward erasure correction to quic," in *2019 IFIP Networking Conference (IFIP Networking)*, 2019, pp. 1–9.

[11] P. Garrido, I. Sanchez, S. Ferlin, R. Aguero, and O. Alay, "rquic: Integrating fec with quic for robust wireless communications," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–7.

[12] F. Michel, A. Cohen, D. Malak, Q. De Coninck, M. Médard, and O. Bonaventure, "Flec: Enhancing quic with application-tailored reliability mechanisms," *IEEE/ACM Transactions on Networking*, vol. 31, no. 2, pp. 606–619, 2023.

[13] J. Wu, B. Cheng, M. Wang, and J. Chen, "Priority-aware fec coding for high-definition mobile video delivery using tcp," *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, pp. 1090–1106, 2017.

[14] E. N. Gilbert, "Capacity of a burst-noise channel," *The Bell System Technical Journal*, vol. 39, no. 5, pp. 1253–1265, 1960.

[15] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings.*, vol. 1, 1999, pp. 345–352 vol.1.

[16] Q. De Coninck, F. Michel, M. Piraux, F. Rochet, T. Given-Wilson, A. Legay, O. Pereira, and O. Bonaventure, "Pluginizing quic," in *Proceedings of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '19, 2019, p. 59–74. [Online]. Available: https://doi.org/10.1145/3341302.3342078

[17] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, "Reproducible network experiments using container-based emulation," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '12, 2012, p. 253–264. [Online]. Available: https://doi.org/10.1145/2413176.2413206

[18] JM Homepage, "JM Homepage," [Online], available: http://iphome.hhi.de/suehring/tml/.

[19] R. A. Fisher, *The Design of Experiments*. Oliver and Boyd, 1949.