Deadline-Aware Transmission Control for Real-Time Video Streaming

Lei Zhang, Yong Cui*, Junchen Pan, Yong Jiang
Department of Computer Science and Technology, Tsinghua University, Beijing, China
{leizhang16,panjc17}@mails.tsinghua.edu.cn, cuiyong@tsinghua.edu.cn, jiangy@sz.tsinghua.edu.cn

2021 IEEE 29th International Conference on Network Protocols (ICNP) | 978-1-6654-4131-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICNP52444.2021.9651971

***Abstract*—The deadline requirements of real-time applications rapidly increase in recent years (e.g., cloud gaming, cloud VR, online conferencing). Due to diverse network conditions, meeting deadline requirements for these applications has become one of the research hotspots. However, the current schemes focus on providing high bitrate instead of meeting deadline requirements. In this paper, we propose D3T, a flexible deadline-aware transmission mechanism that aims to improve user quality of experience (QoE) for real-time video streaming. To fulfill the diverse deadline requirements over fluctuating network conditions, D3T uses a deadline-aware scheduler to select the high priority frame before the deadline. To reduce congestion and retransmission delay, we leverage a deep reinforcement learning algorithm to make decisions of sending rate and FEC (forward error correction) redundancy ratio based on observed network status and frame information. We evaluate D3T via trace-driven simulator spanning diverse network environments, video contents and QoE metrics. D3T significantly improves the frame completion rate by reducing the bandwidth waste before the deadline. In the considered scenarios, D3T outperforms previously approaches with the improvements in average QoE of 57%.**

***Index Terms*—Transmission control, Real-time video streaming, Deadline-aware scheduling, Congestion control, Forward error correction.**

I. Introduction

In recent years, the end-to-end stringent delay constraint requirements (i.e., deadline) have rapidly increased in real-time video streaming, including online conferencing, multi-part interactive live streaming, cloud VR, cloud games, etc. For example, Cloud VR applications should provide the motion-to-photon delay within 25 ms [1]. Other real-time video streaming applications also have similar demands. Differentiated deadline requirements are pervasive in real-time applications.

Many transmission control approaches for video streaming have been proposed to achieve high bitrate and low latency in complicated network environments. The methods such as loss-based approaches [2] [3] and model-based approaches [4] [5] get bitrate as high as possible under the permission of network conditions. However, if a frame has missed its deadline when reaching the receiver, it will not be submitted to the upper application. In this case, the measured throughput is very high, but the bandwidth is actually wasted. Hence achieving a high bitrate does not lead to high QoE. These schemes may cause a significant waste of bandwidth resources.

There has been another class of methods providing low-latency transmission. For example, WebRTC [6] provides a real-time communication service, which incorporates video codec, FEC [7], and google congestion control to optimize video data delivery. Salsify [8] couples the codec and transport process to decide the encoding rate according to the available bandwidth. Delay-based transmission control algorithms [9] [10] deliver data based on TCP. Although these solutions have provided low-latency, they cannot guarantee that the data reach the receiver before the deadline.

The latest IETF drafts [11] [12] propose QUIC-based protocols to try to meet the application's delay requirements. However, these schemes still use an inflexible FEC redundancy strategy and the default congestion control, i.e., Reno [3]. The inflexible algorithms achieve low throughput in lossy networks. Therefore, the frames in real-time applications are either delayed waiting for retransmission or discarded so that these frames cannot reach the receiver before the deadline.

In this paper, we focus on the fundamental transmission issues for the deadline-aware real-time video streaming applications under complicated network conditions. The end-to-end delay consists of three parts: (1) Queueing delay at the sender. If the rate of application data is higher than the sending rate of the sender, the data may queue at the sender's packet buffer. (2) Retransmission delay. In the lossy or high-RTT (round trip time) networks, congestion and random packet loss frequently occur in packet switching networks. Both fast recovery and retransmission timeout will add at least an extra round-trip delay. (3) Queueing delay in the network. If the sending rate of the sender is higher than the available bandwidth, the data may queue at the network buffer.

To address these fundamental delay issues, we consider providing a scheduler to tackle the queueing delay at the sender, leveraging the FEC scheme to alleviate the retransmission delay, and utilizing congestion control algorithm to avoid queueing delay in the network. However, predetermined policies in scheduler, FEC and congestion control schemes fail to handle the complicated scenarios caused by various network conditions and application requirements. Machine learning methods bring opportunities to complex problems and enable flexible decisions. Notwithstanding that using a single model to learn the scheduler, FEC and congestion control

This work is supported by the National Key Research and Development Program of China under Grant 2017YFB1010002, in part by National Natural Science Foundation of China under Grant 61872211, and in part by the National Key Research and Development Program of China under Grant 2018YFB1800303.

*Yong Cui is the corresponding author.

978-1-6654-4131-5/21/$31.00 ©2021 IEEE

policies through deep reinforcement learning (DRL) algorithm is feasible, the training process is faced with high-dimensional state/action space and complex optimization function.

To overcome the above problems, we propose D3T, a novel flexible deadline-aware transmission system that aims to improve QoE by meeting deadline requirements. To make the agent training easier, we design a heuristic scheduling algorithm and train a DRL agent for FEC and congestion control due to their tight coupling.

To summarize, our contributions are listed as follows:

- We propose a novel architecture named D3T, which can provide flexible transmission control for real-time video streaming. It leverages domain knowledge and machine learning that reduces the bandwidth waste to meet the deadline requirements (§III).
- We present a deadline-aware scheduling algorithm that can choose the high priority frame before the deadline and discard invalid frames to avoid the waste of bandwidth resources (§IV-A).
- We design a joint decision model based on DRL to adaptively select redundancy ratio and sending rate, which considered deadline requirements and network status(§IV-B).

To mitigate the risk of rebuffering in real-time video streaming, we use H.264 SVC (temporal and spatial scalable video coding) for deadline-aware applications. The evaluation of D3T is via trace-driven simulator. The results show that D3T outperforms the existing approaches with improvements in average QoE of 57% over a wide range of network environments and different QoE metrics.

II. Motivation

Real-time applications have the deadline requirements for their data transmission. Current scheduling, FEC and congestion control algorithms reduce the queueing delay at the sender, retransmission delay and queueing delay in the network, respectively. Towards the deadline requirements, these current schemes may encounter performance degradation caused by bandwidth waste in the data transmission process.

(1) Bandwidth waste due to overdue frames.

Case1: If the frames have missed their deadline requirements when they reach the receiver, they cannot be submitted to the upper layer application. In the view of transport layer, the bandwidth utilization is very high, but it is futile to transmit these overdue frames.

Case2: If the scheduling algorithm always chooses non-urgent frames to send, many urgent frames are discarded due to missed the deadline. With a finite number of frames, it may cause bandwidth waste because there is no frame to send.

(2) Bandwidth waste due to the undecodable frames.

Case1: In video streaming, the dependency relationships are critical to decode the streams, such as the I/P frames in H.264 or base/enhance layers in SVC [13]. If a frame does not arrive on time, its dependent frames are useless even if they arrive at the receiver on time. Transmitting undecodable frames wastes the bandwidth. Therefore, the scheduling algorithm should consider these dependencies when choosing the frames to send.

Case2: FEC is well-known to be very effective in reliable transmission without retransmission. For example, in RS (Reed-Solomon) code [14], m source packets are selected from a frame. Then the encoder generates n redundant packets based on the source packets. These $m + n$ packets form an FEC group. Once m or more packets, including source and redundant packets, are collected at the receiver, the m original data can be recovered from a matrix equation. If the sender sends k ($k < m$) packets and the remaining packets in the FEC group are dropped due to missed deadline, the k packets sent are undecodable. It is another case of bandwidth waste.

(3) Bandwidth waste due to inappropriate congestion control.

Congestion control is a fundamental mechanism for data transmission. The transport protocol such as TCP or QUIC employs the additive increase multiplicative decrease (AIMD) algorithm for congestion control (e.g., Cubic, Reno). The algorithms achieve lower throughput in lossy networks as they regard the stochastic packet loss as an indicator of network congestion. Hence a large number of bandwidth resources may be idle. It is the third case of bandwidth waste due to the inappropriate congestion control scheme.

III. D3T Architecture

Motivated by the phenomenon of bandwidth waste due to inappropriate policies, we need to design a flexible transmission mechanism to mutually collaborate among scheduling, FEC redundancy and congestion control. Our goal is to deliver as many higher-priority frames as possible before the deadline. Toward this goal, there are various trade-offs among priority, frame size, deadline, frame dependency and dynamic network status. (1) Under the combined influence of these trade-offs, the first challenge is how to pick the frames to send and drop frames to meet the deadline. (2) The second challenge is how to determine appropriate redundant packets over the diverse deadline requirements, frame size and network status to meet the deadline. (3) Beyond that, the performance of congestion control can be dominated by many factors, including traffic patterns, RTT, packet loss and deadline requirements and so on. The third challenge is how to make a decision of the desired sending rate from a complex network environment to meet the deadline requirements.

Inspired by machine learning algorithms, the naive approach is that *one* DRL-based agent learns all of the policies among the scheduler, FEC redundancy and congestion control. The agent needs to get the near-optimal policy from the complex and various frames information, network conditions and their trade-offs. Obviously, it is a hard job to get the agent due to high-dimensional state/action space and complicated design of reward function.

To tackle these challenges, we propose D3T, a flexible transmission mechanism for real-time video streaming. D3T includes a deadline-aware scheduler to avoid bandwidth waste and a DRL-based agent to get an adaptive policy of FEC

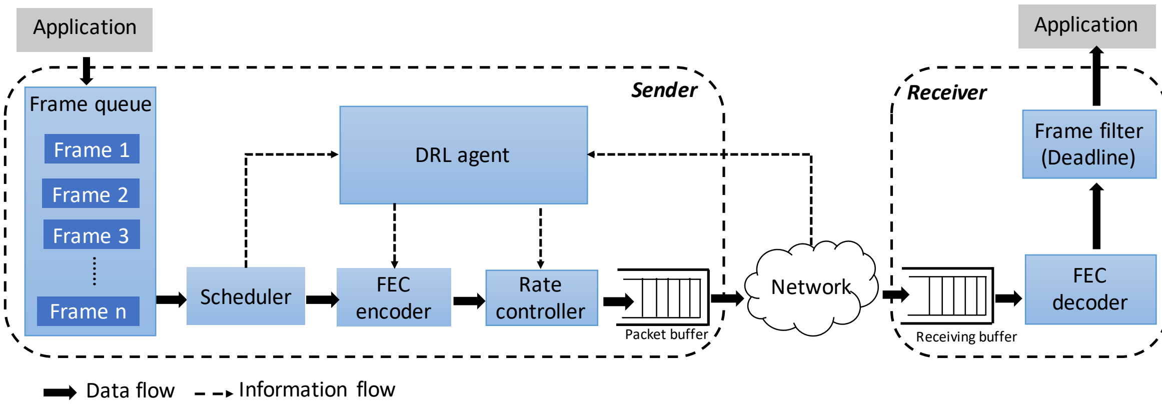


Fig. 1: D3T architecture.

redundancy and congestion control. Separating the scheduler from the DRL model reduces the state/action space. Meanwhile, the reward function only needs to give feedback to the frame completion information and current network status.

In this paper, we design the D3T architecture for end-to-end deadline-aware transmission of real-time video streaming. The goal of D3T is to achieve the highest frame completion ratio before deadline with the flexible transmission control for frame selection, redundancy ratio and sending rate. Motivated by above challenges, D3T system is composed of a deadline-aware scheduler and a DRL agent. The deadline-aware scheduler dynamically selects an urgent frame and drops overdue frames according to the deadline and priority. The DRL agent learns the adaptive redundancy ratio and sending rate.

The architecture of D3T is shown in Fig. 1 that places on the sender. In the beginning, the real-time applications continuously generate new frames on the sender side. Each frame stores in the frame queue. The scheduler selects a frame to send and drops overdue frames. If the frame will miss the deadline, the FEC encoder will generate redundant packets according to the ratio from the DRL agent. These redundant packets along with the original packets will be transmitted to the congestion control. If the frame does not need to add redundant packets, it will directly be sent to congestion control. Then congestion control is responsible for monitoring the network status and getting the sending rate from the DRL agent. The sender sends these packets to the receiver via the network based on the decision. On the other side, the receiver will receive data and decode each frame.

We leverage the soft action-critic algorithm (SAC) [15], a state-of-the-art reinforcement algorithm that deals with the continuous action space to train the agent. The agent takes the frame information and network status and outputs the FEC redundancy ratio and sending rate in each decision interval. The sender is responsible for periodically estimating network status. The network status including average delay, packet loss rate and throughput is counted at the sender. The receiver sends back ACKs to the sender. The frame completion ratio and network status are computed according to ACKs. The above information is provided not only to the scheduler but also to the DRL agent.

IV. Detailed Design

In this section, we present the detailed design of D3T. We begin with introducing the deadline-aware scheduler. Then we describe adaptive FEC redundancy and congestion control.

A. Deadline-aware scheduler

When the bandwidth resource is not enough, the frames continuously generated from the real-time applications will accumulate at the sender side. The scheduler determines the order in which frame is sent and discards the obsolete frames. The goal of the deadline-aware scheduler is to transmit as many high-priority frames before the deadline. To achieve the goal, the priority, deadline and network status should be taken into consideration when designing the scheduling algorithm.

The first step of the scheduler is to calculate the $remaining\_time$ and $weight$ of all frames in the frame queue according to the equation (1) and (2). Then the scheduler chooses a frame with the smallest $weight$. If $deadline$ minus $passed\_time$ of a frame is less than 0, it means that the frame has been obsoleted and will be dropped from the frame queue.

$$\begin{aligned} remaining\_time &= deadline - passed\_time \\ &- \frac{estimated\_rtt}{2} - \frac{remaining\_size}{current\_sending\_rate} \end{aligned} \tag{1}$$

$$weight = \frac{remaining\_time/deadline}{1 - priority/max\_priority} \tag{2}$$

where $pass\_time$ is the waiting time of a frame in the frame queue. And $remain\_size$ is the number of unsent packets of the frame. $estimated\_rtt$ represents the current RTT. The priority of the frame from high to low is defined from 0 to N-1 (for example N priority levels). For example, in video streaming, the priority of I frame, B frame, and P frame are defined as 0, 1, and 2, respectively. The $max\_priority$ in equation (2) is equal to N.

B. Adaptive redundancy and congestion control

1) Deep reinforcement learning for D3T:

The FEC redundancy and congestion control problems are formulated as deep reinforcement learning tasks. The DRL agent provides a dynamic policy to map current observations (i.e., *state*) to the redundant ratio and sending rate (i.e., *action*). As shown in Fig. 2, we present the components of the actor-critic framework of D3T.

State space. The state is a snapshot of the environment observed by the agent after taking action over a period. Although a large amount of state may reflect the network state more comprehensively, it will also increase the difficulty of model learning. Therefore we choose the performance metric that represents the network condition as the agent's state. The state s_t at time step t is represented as (1) the current frame deadline, (2) the frame remaining size, (3) the frame priority, (4) the average throughput, (5) the current RTT, (6) the packet loss, and (7) the last action. The above statistics information can be easily obtained from the sender by tracing the ACKs at each decision period.

Action. The agent gets an action to respond to the observed state of the environment according to the *policy*. The policy is represented as a neural network with a manageable number of adjustable parameters. In our formulation, the action space is continuous and the action a_t is defined as a tuple that includes the redundant ratio fec_ratio and the sending rate $rate$ of the sender. In general, the agent takes an action at the end of each interval period.

Reward. The reward is the award or punishment for positive and negative actions. In each interval period, the sender updates the observed state and executes an action to adjust the redundant ratio and sending rate, which results in an instantaneous reward r_t. In this paper, we adopt a reward function that awards the frame completion ratio and throughput while penalizing loss and delay, aiming to achieve a high frame completion ratio, high throughput, low delay and low congestion loss rate. The reward r_t is defined as:

$$\begin{aligned} r_t = f_{norm}(\frac{throughput_t}{ave\_delay_t}) - f_{norm}(loss\_rate_t) \\ + f_{norm}(completion\_ratio_t) \end{aligned} \tag{3}$$

where $throughput_t$ is the instantaneous observed throughput of the sender. avg_delay_t is the average delay which is calculated with RTT samples measured within 0.1s. $loss\_rate_t$ is the observed packet loss rate form the sender. $completion\_ratio_t$ is the frame completion ratio at time step t. f_{norm} is the normalization function. Generally, the agent selects an action to maximize the expected cumulative reward.

Training algorithm. To train D3T agent, we leverage the SAC [15] algorithm which is an actor-critic algorithm with off-policy for maximum entropy reinforcement learning. As shown in Fig. 2, the actor network is responsible for choosing the proper action. The critic network estimates the value of an action, and conducts it to update the parameters of actor and critic networks.

2) Adaptive redundancy:

If the $remaining\_time$ of the current frame is beyond the deadline, the FEC codec will not execute. Otherwise, if the $remaining\_time$ of the frame has less than two RTT or the frames that could be corrupted under the larger available bandwidth, the redundant packets will generate from the original packets. In detail, we adopt the $remaining\_size$ of the frame as the size of the original packets m. Then the n redundant packets generate from m original packets according to the redundancy ratio fec_ratio taken from the DRL agent. The FEC group is assembled according to the above (m, n). After the current frame has been sent, the new frame needs to get a new fec_ratio from the agent again.

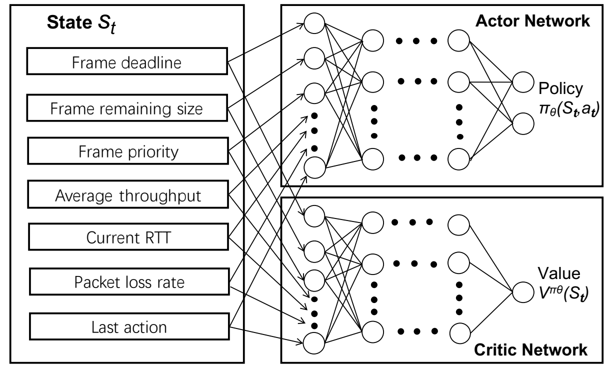


Fig. 2: The actor-critic algorithm that D3T uses to generate the policies of FEC redundant ratio and sending rate.

3) Congestion control:

Congestion control regulates the sending rate at each sender to maximize the throughput, minimize the queueing delay and the packet loss. Specifically, congestion control monitors the network conditions and provides the network measurements, such as throughput, RTT and loss rate. These network conditions is provided to the scheduler and the agent for decision making. In this paper, we adopt the sending rate as the decision-making. At each decision period, congestion control gets the sending rate from the DRL agent. The sender persistently sends the packets according to the current sending rate until it takes a new action from the agent.

V. Evaluation

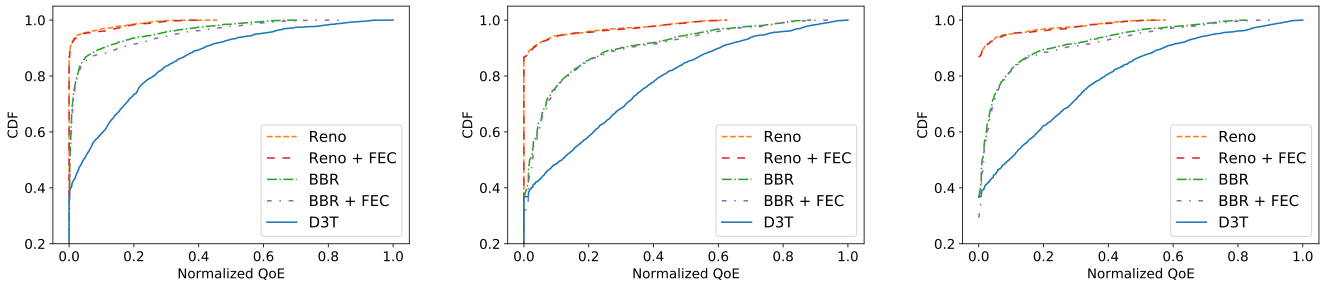
In this section, we evaluate D3T under various network conditions and QoE metrics. Our experiments are conducted in our simulator using Python[1].

A. Dataset and metrics

Video dataset. We train and test D3T on public video dataset [16] In our experiments, through the open-source encoder JSVM [17], we successfully encode the AVC video streaming into SVC streaming that supports different frame rates and resolutions. Leveraging JSVM, we obtain the SVC streams that include the resolution of 360P, 720P, 1440P and 7.5fps, 15fps, 30fps streams through the 1440P AVC video streaming.

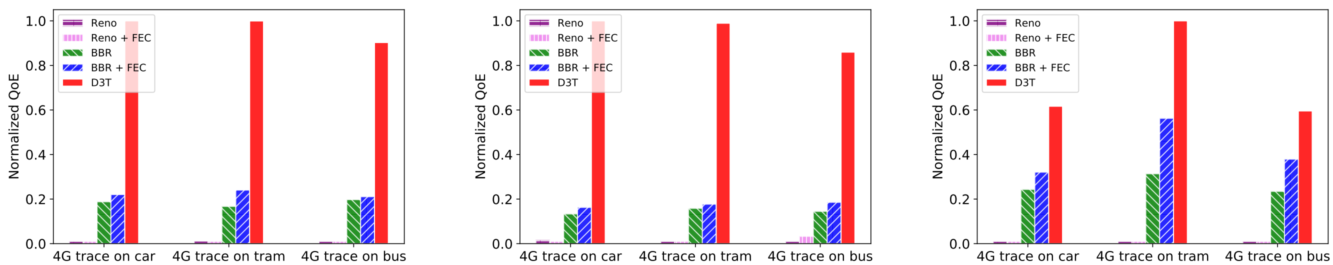
Network traces. To train and evaluate D3T, we also use the Belgium dataset [18] which consists of throughput measurements in cellular networks. The dataset is collected based on different types of transportation such as tram, car and bus.

[1] https://github.com/AItransCompetition/Meet-Deadline-Requirements-Challenge



(a) QoE_{lin} (b) QoE_{log} (c) QoE_{hd}

Fig. 3: Comparing D3T with existing schemes on different QoE metrics.



(a) Games (b) Movies (c) Sports

Fig. 4: QoE performance using different content videos under the cellular traces.

QoE metrics. To evaluate the D3T, we consider using QoE metrics with essential factors based on previous schemes [19] [20] which have been widely accepted by researchers in network systems. A general QoE metric is evaluated with bitrate, rebuffering, smoothness or frame-skipping. Considering the frame-deadline condition, we add some parameters into general QoE function to make them suitable for deadline evaluation. The QoE metrics is defined as following:

$$\begin{aligned} QoE = &\sum_{n=1}^{N}(\alpha(1-D_n)f(R_n) - \beta D_n T_n) \\ &- \sum_{n=1}^{N-1}\gamma|f(R_{n+1})(1-D_{n+1}) - f(R_n)(1-D_n)| \end{aligned} \tag{4}$$

where N is the number of GoPs. R_n represents the bitrate of a GoP and $f(R_n)$ maps that bitrate to the quality perceived by a user, including three choices provided by Pensieve [19]. In our experiment, we analyze the max bitrate of a GoP with arrived layers according to SVC layers dependencies. T_n represents the sum of rebuffering and frame-skipping time of GoP n. The last term represents the smoothness of video quality. The coefficients α, β and γ are the different trade-offs of application scenarios or user preferences. Specifically, the values of these coefficients we consider in our evaluation are provided in Table I.

Baseline. We compare D3T with representative solutions.

(1) Reno: the default congestion control algorithm used by TCP and QUIC. It takes the packet loss as the congestion signal. A default scheduler FIFO and a zero FEC redundancy are used in this baseline.

TABLE I: QoE metrics in our evaluation.

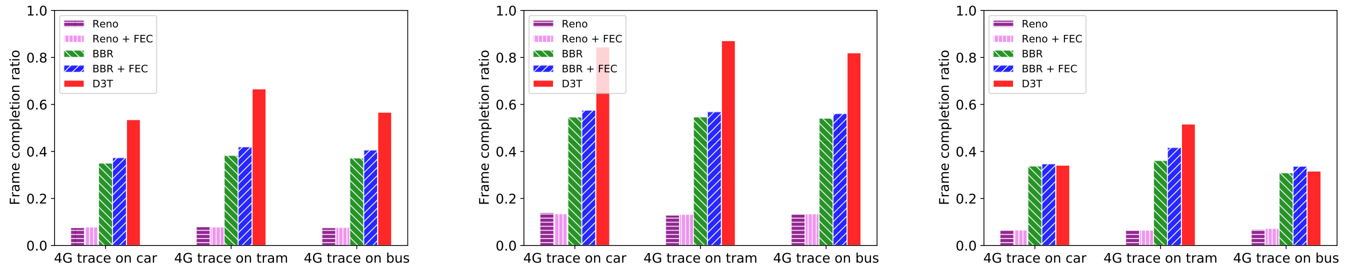
Name	Bitrate utility($f(R)$)	α	β	γ
QoE_{lin}	R	1.0	4.3	0.05
QoE_{log}	$\log(R/Rmin)$	1.0	2.66	0.01
QoE_{hd}	420->1,500->2,580->3, 1880->12,2220->13,2640->14 5040->23,6140->24,7440->25	1.0	8	0.2

(2) BBR [4]: a state-of-art congestion control algorithm given by google. BBR discards packet loss as the congestion signal and uses complex detection mechanisms. A default scheduler FIFO and a zero FEC redundancy are used in this baseline.

(3) Reno + FEC and BBR + FEC: respectively use congestion control algorithm and scheduler as approaches above, but they set up the FEC redundancy algorithm according to a heuristic algorithm [12] that take loss rate into consideration.

B. D3T performance

QoE performance. To evaluate D3T, we first compare it with existing transmission approaches with QoE metrics above. We set up net traces with a wide range of RTT and loss in order to test the flexibility of different solutions. Fig. 3 shows the average QoE that each solution achieves in traces set. D3T outperforms existing solutions with improvements in average QoE of 57% across various network conditions and QoE metrics. For QoE_{lin}, the average QoE for D3T is 11% higher than existing schemes on various network traces on average. For QoE_{log} and QoE_{hd}, D3T achieves 135% and 26% higher than other schemes. The baseline solution with fixed policy suffers from low bandwidth utilization in lossy networks and also is not able to selectively increase FEC redundancy



(a) Games (b) Movies (c) Sports

Fig. 5: Frame completion ratio using different content videos under the cellular traces.

for urgent frames. Whereas D3T gives better performance depending on its flexible policy about FEC redundancy and congestion control.

Frame completion ratio. Except for showing results with overall QoE performance figures, we also pick different content videos and conduct experiments for frame completion ratio and corresponding QoE. Fig. 4 and Fig. 5 illustrate the QoE performance and average frame completion ratio under various types of videos. We find that D3T outperforms the existing solutions to improve frame completion ratio by 16% to 52% on average, under the D3T's adaptive FEC redundancy and congestion control policy. It is also worth mentioning that the frame completion ratio of the BBR+FEC scheme is slightly higher than D3T on sports videos as shown in Fig. 5(c). The main reason is that the frame completion ratio is an index of transmission evaluation, which ignores the dependencies among video frames. To point this out, we also give Fig. 4, the corresponding QoE results in same test settings, which shows that D3T achieves better QoE than other schemes. BBR+FEC ignores the structure of video stream data and sends more useless frames that cannot be decoded according to considering frame dependencies. Therefore, the BBR+FEC scheme gives a poor QoE despite achieving a better frame completion ratio as shown in Fig. 4(c).

VI. Conclusion

In this paper, we propose D3T, a flexible deadline-aware transport system with a deadline-aware scheduler and an adaptive DRL agent for FEC redundancy and congestion control without any human instruction to meet the deadline requirements. The deadline-aware scheduler chooses the high-priority frame before the deadline and discards obsolete frames to avoid bandwidth waste. We train the DRL agent to achieve flexible policies to adapt to the dynamic network condition and diverse deadline requirements. To verify the D3T's behavior, we evaluate D3T in our simulator. The experiments with different video contents, network conditions and QoE metrics have been conducted to illustrate the D3T to enhance both QoE and frame completion ratio before the deadline.

References

[1] Z. Lai, Y. C. Hu, Y. Cui, L. Sun, N. Dai, and H.-S. Lee, "Furion: Engineering high-quality immersive virtual reality on today's mobile devices," *IEEE Transactions on Mobile Computing*, vol. 19, no. 7, pp. 1586–1602, 2019.

[2] S. Ha, I. Rhee, and L. Xu, "Cubic: a new tcp-friendly high-speed tcp variant," *ACM SIGOPS operating systems review*, vol. 42, no. 5, pp. 64–74, 2008.

[3] V. Jacobson, "Congestion avoidance and control," vol. 18, no. 4. ACM New York, NY, USA, 1988, pp. 314–329.

[4] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: congestion-based congestion control," *Communications of the ACM*, vol. 60, no. 2, pp. 58–66, 2017.

[5] G. Carlucci, L. De Cicco, S. Holmer, and S. Mascolo, "Analysis and design of the google congestion control for web real-time communication (webrtc)," in *Proceedings of the 7th International Conference on Multimedia Systems*, 2016, pp. 1–12.

[6] "Web real time communication," https://webrtc.org.

[7] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, 1998.

[8] S. Fouladi, J. Emmons, E. Orbay, C. Wu, R. S. Wahby, and K. Winstein, "Salsify: Low-latency network video through tighter integration between a video codec and a transport protocol," in *NSDI*, 2018, pp. 267–282.

[9] J. S.Shalunov, G.Hazel and M.Kuehlewind, "Low extra delay background transport," *IETF RFC 6817*, 2012.

[10] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "Tcp vegas: New techniques for congestion detection and avoidance," pp. 24–35, 1994.

[11] T. Pauly, E.Kinnear, and D.Schinazi, "draft-ietf-quic-datagram-00, an unreliable datagram extension to quic," *Internet Engineering Task Force*, 2020.

[12] Y.Cui, Z.Liu, H.Shi, J.Zhang, K.Zheng, and W.Wang, "draft-shi-quic-dtp-01, deadline-aware transport protocol," *Internet Engineering Task Force*, 2020.

[13] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h. 264/avc standard," *IEEE Transactions on circuits and systems for video technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[14] S. B. Wicker and V. K. Bhargava, *Reed-Solomon codes and their applications*. John Wiley & Sons, 1999.

[15] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[16] C. G. Bampis, Z. Li, I. Katsavounidis, T.-Y. Huang, C. Ekanadham, and A. C. Bovik, "Towards perceptually optimized end-to-end adaptive video streaming," *arXiv preprint arXiv:1808.03898*, 2018.

[17] "Jsvm project," https://github.com/daluege/jsvm.

[18] J. Van Der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. R. Alface, T. Bostoen, and F. De Turck, "Http/2-based adaptive streaming of hevc video over 4g/lte networks," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2177–2180, 2016.

[19] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 197–210.

[20] G. Yi, D. Yang, A. Bentaleb, W. Li, Y. Li, K. Zheng, J. Liu, W. T. Ooi, and Y. Cui, "The acm multimedia 2019 live video streaming grand challenge," in *ACM International Conference on Multimedia*, 2019, pp. 2622–2626.