J Inf Process Syst, Vol.19, No.6, pp.817~829, December 2023
https://doi.org/10.3745/JIPS.04.0298

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Quality Adaptation of Intra-only Coded Video Transmission over Wireless Networks

Shu Tang*, Yuanhong Deng, and Peng Yang

Abstract

Variable wireless channel is a big challenge for real-time video applications, and the rate adaptation of real-time video streaming becomes a hot topic. Intra-video coding is important for high-quality video communication and industrial video applications. In this paper, we proposed a novel adaptive scheme for real-time video transmission with intra-only coding over a wireless network. The key idea of this scheme is to estimate the instantaneous remaining capacity of the network to adjust the quality of the next several video frames, which not only can keep low queuing delay and ensure video quality, but also can respond to bandwidth changes quickly. We compare our scheme with three different schemes in the video transmission system. The experimental results show that our scheme has higher bandwidth utilization and faster bandwidth change response, while maintaining low queuing delay.

Keywords

Bandwidth Change Response, Bandwidth Utilization, Intra-only Coding, Queuing Delay, Real-Time Video Transmission

1. Introduction

With the rapid advance of communication and video technology, applications and technologies based on real-time video are increasingly emerging. Wireless communication has become a popular mode of communication. The video camera has the potential to replace many sensors for process control, which has the advantages of low cost and universally usable. Therefore, real-time video transmission becomes the basis of many industrial applications [1].

The real-time video streaming transmission requires higher bandwidth and lower latency. However, the rapid variation of the link capacity brings challenges for real-time video streaming transmission over a wireless network. The mismatch between video rate and bandwidth will lead to low video quality or network congestion. To solve such problems, an effective adaptive transmission control strategy, which can respond to bandwidth changes quickly and guarantee the requirements of the video quality and the delay, is desperately needed [2].

The inter-prediction coding is a coding method of most applications because of its high compression efficiency, easy storage, and transmission. However, in the unreliable transmission system, the loss of

Manuscript received July 29, 2021; first revision November 29, 2021; accepted December 23, 2021.
* **Corresponding Author:** Shu Tang (tangshu@cqupt.edu.cn)
Chongqing Key Laboratory of Computer Network and Communications Technology, Chongqing University of Posts and Telecommunications, Chongqing, China (tangshu@cqupt.edu.cn, dyhms@outlook.com, 1432877634@qq.com)

reference frames (I frame) will lead to many subsequent frames (P frame and B frame) cannot be decoded and the video cannot play for a period of time, which will reduce the user quality of experience (QoE). To ensure the reliability of transmission, retransmission is needed, which will result in a long delay and is not suitable for delay-sensitive applications. The intra prediction coding is one of the most popular coding methods, which encodes and decodes the frame independently of other frames and uses spatial redundancy for intra-prediction coding [3]. The intra-video coding not only can enhance the video quality, but also has low complexity, error propagation prevention, and low energy consumption. Therefore, intra-only coding becomes the choice for many low-latency and resource-constrained applications [4].

In this paper, we proposed an adaptive scheme for real-time video transmission with intra-only coding strategy over a wireless network. The key idea of this scheme is to estimate the instantaneous network capacity via the feedback information of the receiver, which can respond to the bandwidth changes faster and optimize each frame quality according to the network capacity. We use H.264 encoder to encode the video, and adjust the quality of the video frames by the quantization parameter (QP).

The rest of the paper is organized as follows: Section 2 introduces related work, Section 3 describes the design of our proposed adaptive scheme, Section 4 presents the experimental results, and Section 5 concludes the paper and future works.

2. Related Work

In recent years, the bitrate adaptive mechanism of video streaming based on HTTP has attracted wildly studies, such as HTTP Live Streaming (HLS) [5], Dynamic Adaptive Streaming over HTTP (DASH) [6]. The original HLS and DASH cannot fully meet the requirements of real-time rate adjustment, so some researchers proposed algorithms or methods to improve their performance [7-9]. However, most of the algorithms based on DASH and HLS are designed for the video on demand or live video. They usually need a long delay for buffering and retransmission, which are not suitable for end-to-end delay sensitive real-time video transmission.

Without actively dropping frames, the key to video adaptation is to control the bitrate of encoder output, which is related to many factors. The Q-STAR model [10] proves that if the frame rate and resolution of the video are unchanged, the video quality and bitrate depend on the quantization step (QS), and the QP is the serial number of QS. He et al. [11] proposed an adaptive video transmission mechanism that analyzes the queue length of the receiver buffer and its future variation trend, the terminal fed back the judgment and prediction, and the video's encoding QP. But this scheme also needs to set the buffer, which increases the end-to-end delay.

For some video adaptive schemes based on bitrate control, bandwidth estimation is the most basic and important process. Estimation of available bandwidth has shown much interest in the real-time multimedia application that needs some quality of service (QoS) guarantees. These schemes estimate the bandwidth and change the encoding bitrate for adaptation. Bandwidth estimation in wireless networks is a more challenging issue than wired networks due to variable wireless conditions [12]. The efficiency of adaptation depends largely on the efficiency of bandwidth estimation. Besides, some researchers proposed congestion control for real-time video streaming, including network assisted dynamic adaptation

(NADA) [13] proposed by Cisco Systems and Google Congestion Control (GCC) [14] proposed by Google. These methods judged network congestion by the delay and the packet loss rate, which were used to control the video transmission rate [15,16]. These methods controlled the rate after the appearance of the congestion signal, such as the delay or packet loss rate exceeds the threshold, which made their congestion responses reactive.

There are many delay components in the time from video capture to computer response or display. Bachhuber et al. [1] analyzed various delays in end-to-end video transmission systems, and proposed methods to decrease the delay at hardware and software levels. However, it did not consider the transmission problem caused by network fluctuation. In video multicast, the sender needs to send appropriate video streams to the terminals with different conditions. To solve this problem, Schwarz et al. [17] proposed algorithms that use Scalable Video Coding (SVC) in software defined networking (SDN) to minimize the bandwidth consumption in the core network and achieve adaptive layer loss [18]. This algorithm has two versions: minimizing the latency, and minimizing the bandwidth consumption. The SVC does not need network feedback information but uses the number of enhancement layers received by terminals with different conditions to achieve video adaptation. But due to the limitation of the layer, SVC cannot use the bandwidth with high accuracy, and it has low coding efficiency and high decoding complexity. Fouladi et al. [19] proposed a framework, called Salsify, to integrate the transport protocol and the codec, which selects the frame to be sent according to the instantaneous network capacity. It eschews explicit adjustment of bitrate and controls the video streaming transmission from frame level. But its invariable delay parameter made stationary queuing delays, and could not be applied to wireless networks well.

3. Proposed Scheme

The general video transmission system consists of two components: video codec and transport protocol, and they work independently. The transport protocol estimates the network conditions through the network information and updates the bitrate of the encoder. The encoder reads frames from the camera and compresses frames to achieve a particular average bit rate. In our scheme, the transport protocol communicates network information to the codec before each frame is compressed, so that the codec adjust the quality of video frames to match the network's capacity.

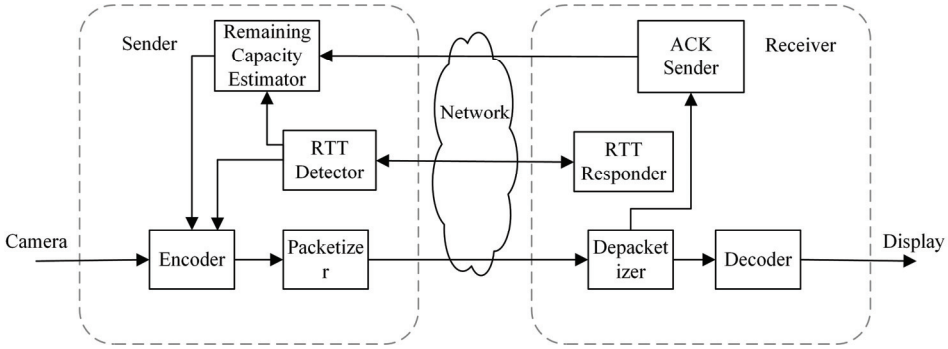

Fig. 1. The proposed adaptive video transmission system framework.

3.1 System Framework

Fig. 1 shows the proposed adaptive video transmission system framework. At the sender, the Encoder compresses all frames with intra-prediction. Each compressed frame is packaged into one or more 1,500-byte packets by the Packetizer and the packet's head contains a sequence number and a flag bit which indicates the end of one frame. Then the packets are transmitted over UDP to the receiver, which replies with acknowledgments (ACK). These packets are combined, decoded, and played at the receiver. The round-trip time (RTT) Detector sends an RTT probe packet to the receiver, which contains a record of the sending time. The receiver returns the probe packet immediately after receiving it. The sender records the time of receiving the returned probe packet and calculates the RTT, which is provided to the encoder and the remaining capacity estimator (RCE) for further calculation. The frequency of sending an RTT probe packet is equal to the framerate. The RCE estimates the remaining network capacity (it is also called network capacity in the following) through the obtained information and provides it to the Encoder. Finally, the Encoder determines the change of quality of frames.

3.2 Network Capacity Estimation

The receiver records the arrival time of each packet, which is used to maintain a dynamic average of packet inter-arrival times. However, the sender does not send the packets continuously—it pause between two frames. As a result, the inter-arrival time between the first fragment of one frame and the last fragment of the previous frame cannot be used as an indicator of the network capacity, as shown in Fig. 2.

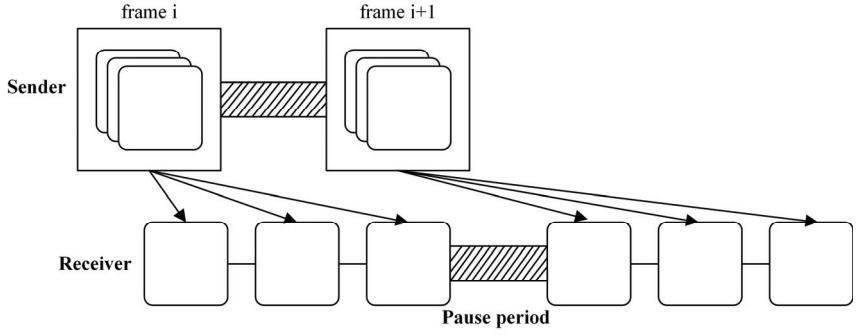

Fig. 2. The "pause period" is not helpful for estimating the inter-arrival time.

This pause could make the receiver underestimate the network capability. To avoid this situation, in the proposed adaptive scheme, the receiver ignores the next inter-arrival time when it receives the last fragment of one frame. As fragment i is received, the receiver calculates the smoothed inter-arrival time τ_i if the flag bit is not 1, as

$$\tau_i = \alpha(T_i - T_{i-1}) + (1-\alpha)\tau_{i-1}, \tag{1}$$

where T_i is the received time of fragment i. α is the weight coefficient. The value of α is 0.1 in our implementation, approximating to the moving average of the last 10 arrival times. Each time a fragment is received, the receiver sends an ACK message which contains this estimation and sequence number.

The sender estimates the remaining network capacity C_r based on the latest information provided by the receiver, as

$$C_r = (d/\tau_i - N_i) \times 1500, \tag{2}$$

where d is the estimation of one-way delay (OWD) from sender to receiver in a wireless network, whose value is half of the RTT. The τ_i is the latest average inter-arrival time reported by the receiver. N_i is the estimation of number of packets on the forward path, whose value is half of subtracting the sequence number of the last-sent packet and the last-acknowledged packet. The reason for taking half is that only the number of packets on the forward path is needed, and the number of packets on the forward path and the backward path is similar (the frequency of sending video data packets and ACKs is the same), as shown in Fig. 3(a). The size of one packet is 1500 bytes.

Therefore, there can be no more than d/τ_i packets on the forward path. The $d/\tau_i - N_i$ indicates the number of packets that the network can still accommodate. Furthermore, the remaining network capacity can be estimated by the prescribed packet size.

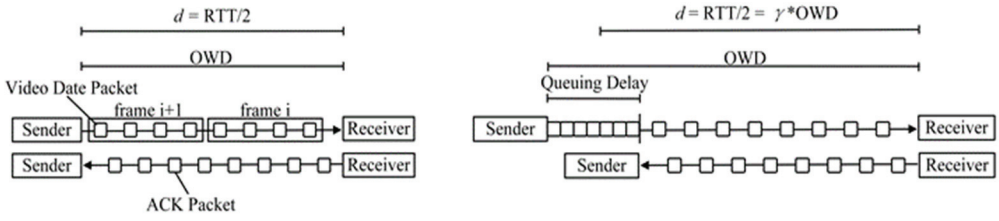

Fig. 3. Packets on forward and backward paths for the queued and unqueued link: (a) packets on path and (b) packets on queuing path.

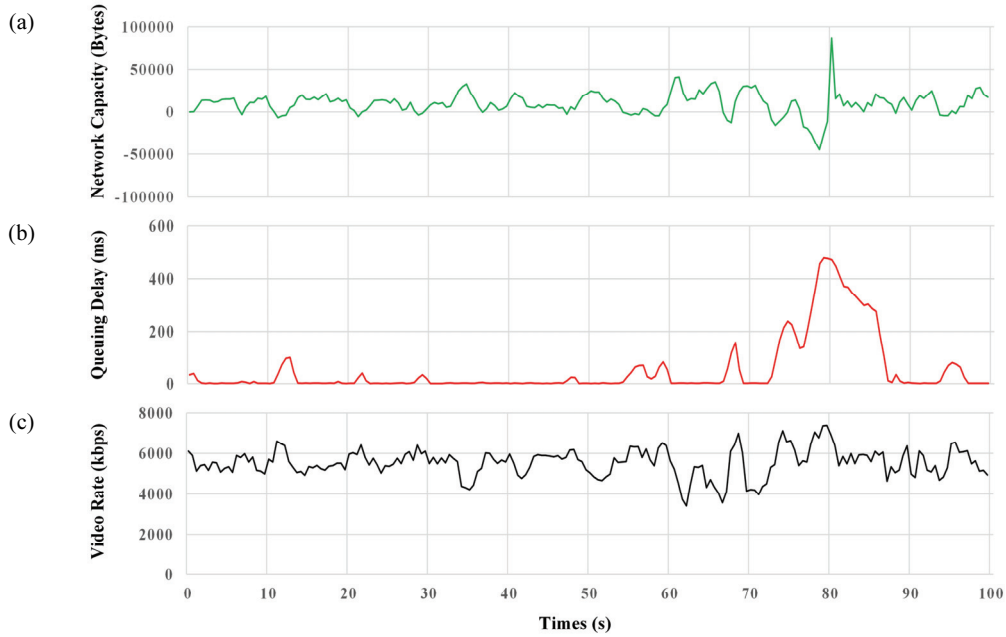

Fig. 4. The relationship of (a) remaining network capacity, (b) queuing delay, and (c) video rate.

In order to explore the significance of the remaining network capacity intuitively, the characteristic data of a video streaming, which is encoded at a certain average rate and transmitted over a network with a bandwidth of 6 Mbps, is shown in Fig. 4, including (i) the real-time video rate, (ii) the queuing delay, calculated by the RTT minus propagation delay, (iii) the remaining network capacity, calculated by (2). All the measured data have an average value of 0.5 seconds. As Fig. 4 shows, when the video rate exceeds the bandwidth (6 Mbps), which is obvious in the 11th, 55th, 67th, 73rd, and 93rd seconds, the queuing delay is increasing (i.e., the network is congesting) and the remaining network capacity is less than 0. Therefore, when the remaining network capacity is negative, the video rate should be decreased. On the contrary, when the video rate is lower than the bandwidth, the queuing delay is close to 0 most of the time (i.e., the network is smooth), and the remaining network capacity is more than 0. In particular, in a period of time after the 80th second, the network capacity is surplus. At the time, the network is still in congestion but in remission (i.e., the queuing delay is decreasing). Therefore, when the remaining network capacity is more than 0, the network state is smooth or tends to be smooth. We should not further congest the network when the queuing delay is reduced. We judge whether the network is in remission, as:

$$\Delta d < \delta, \tag{3}$$

where Δd is the difference of average RTT between two frames. δ is the delay threshold, which is 1 ms in our implementation. If (3) is satisfied, the queuing delay is not considered to be in remission.

3.3 Quality Adjustment Algorithm

In most schemes based on congestion control, the improvement of video quality is blind, such as GCC or TCP-like algorithms. In order to solve the problem of when the video quality can be increased, we propose a quality adjustment algorithm based on the remaining network capacity. The pseudo-code of the adaptive algorithm is shown in Algorithm 1.

In order to determine whether the remaining network capacity can accommodate the video frames with improved quality, the sender estimates the cost of improving the video frames quality C_n, as:

$$C_n = l_p \cdot \beta \cdot f \cdot d, \tag{4}$$

where l_p is the length of the last frame. $\beta \in [0.1,0.2]$, is the incremental coefficient that the increment of frame length caused by improving frame quality. As described in [20], in H.264, an increase of 6 in the quantization parameter approximately halves the bitrate. The value of β is 0.2 in our implementation. f is the video framerate.

$l_p \cdot \beta$ means the length increment of one frame when the quality of frames is improved. $f \cdot d$ indicates the number of frames in flight. Intra-coded frames are encoded independently so successive frames are similar in length because they are similar in content, so lp is used to estimate the length of next few frames. Therefore, the needed size is $l_p \cdot \beta \cdot f \cdot d$ for the improved video streaming. Based on the analysis in Section 3.2, the sender will improve quality when the needed network capacity does not exceed the remaining network capacity and the queuing delay is not decreased (Line 4 in Algorithm 1). The QP is decreased by 1 in our implementation. When the number of packets in flight exceeds the capacity of the network, the remaining network capacity is less than 0, the sender will reduce the quality of the video

(Line 5 in Algorithm 1). The QP is increased by 1 in our implementation. In other cases, maintain the existing quality (Line 6 in Algorithm 1).

$$(d/\tau_i - N_i) \times 1500 > l_p \cdot \beta \cdot f \cdot d, \tag{5}$$

$$(d/\tau_i - N_i) \times 1500 < 0. \tag{6}$$

Besides, when the queue has occurred, the calculated values of d and N_i will be less than the actual value in the same proportion γ, $\gamma \in (0,1)$, as shown in Fig. 3(b). However, it will not disturb the judgment, because the factor γ can be eliminated. The judgment condition, (5) and (6) changes when a queue has occurred, as:

$$\begin{gathered}\big(d(1/\gamma)/\tau_i - N_i(1/\gamma)\big) \times 1500 > l_p \cdot \beta \cdot f \cdot d(1/\gamma), \\ \Big(\big(d(1/\gamma)\big)/\tau_i - N_i(1/\gamma)\Big) * 1500 < 0,\end{gathered} \tag{7}$$

where $d(1/\gamma)$ is the approximate actual value of OWD and $N_i(1/\gamma)$ is the approximate actual number of packets on the forward path. The factor $1/\gamma$ can be eliminated by multiplying γ on both sides of the inequality.

Algorithm 1. Adjust quality

1: set to value:
  τ_i ←mean_inter_arrival_time
  d ←RTT/2
  Ni←(last_recv_seq-last_send_seq)/2
2: remaining_capacity ← $(d/\ \tau_i - N_i) \times$ 1500bytes
3: needed_capacity ← last_frame_length × 0.2 × framerate × d
4: **if** remaining_capacity > needed_capacity (i.e. Formula (5)) and the delay is not decreasing (i.e. Formula (3)) are met
  Improve Quality
5: **else if** remaining_capacity < 0 (i.e., Formula (6)) are met
  Reduce Quality
6: **else**
  Hold
7: **endif**

4. Experimental Results

4.1 Experimental Setup

To evaluate the performance of the scheme proposed in this paper, we design a video transmission system on two Ubuntu 20.04 virtual machines connected by Ethernet. The Traffic Control, which is the traffic control tool in the Linux kernel, is used to set network parameters such as available bandwidth and delay to emulate the wireless network. The one-way propagation delay is set to 50 ms both on the forward path and backward path. To eliminate the impact of video content differences, we capture a 3000-frame

YUV video, and use the same video for every emulation. The ×264 is used to encode raw video frames into intra-coded frames, and the framerate is 30 fps. Finally, we record each value of RTT and the average video rate per 500 ms.

The performance is compared with three different schemes:

CC-AIMD: we design a delay-based controller and a loss-based controller in our video transmission system [14]. The delay-based controller changes the state according to the finite state machine (FSM) in [14], which computes the video rate by additive increase and multiplicative decrease algorithm (AIMD), And the delay static threshold, the increasing step and the decreasing rate are set to 20 ms, 20 kbps and 0.85, respectively. The loss-based controller computes the video rate according to two packet loss ratio thresholds. The thresholds of packet loss ratio are set to 0 and 0.02, respectively.

CC-MIMD: Based on CC-AIMD, we use the multiplicative increase and multiplicative decrease algorithm (MIMD) to compute the rate instead of AIMD [21]. the increasing rate and decreasing rate are set to 1.05 and 0.85, respectively.

BE: We use the method of sending probe packet pair described in [10] to estimate the wireless bandwidth, and set the estimated value to the output bitrate of the encoder.

4.2 Variable Link Capacity

In this experiment, the link capacity starts from 6 Mbps, and is changed to four different rates: 8 Mpbs, 10 Mpbs, 14 Mpbs, and 12 Mpbs. Each rate lasts 20 seconds (Fig. 5). Besides, the packet loss rate is set to 0. We consider the following metrics to evaluate the performance of schemes: (i) bandwidth utilization $U=R/b$, where b is the known link capacity and R is the average video rate; (ii) queuing delay Dq, estimated by the difference between the value of the RTT and the propagation delay. We compute the values of the 5th, 50th, 95th percentiles (i.e., the smaller values of 150th, 1500th, and 2850th, respectively), and the average value.

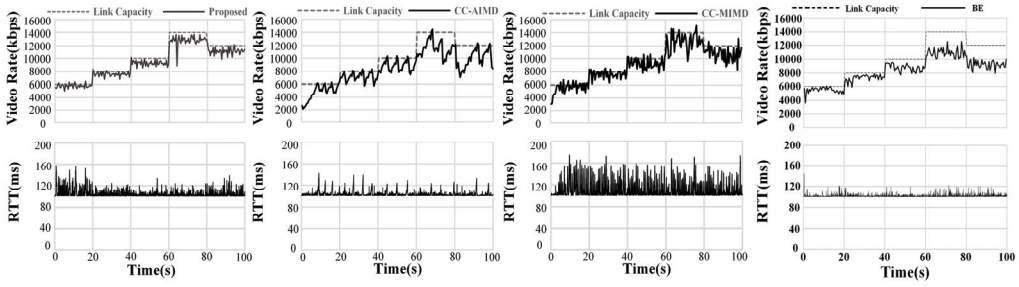

Fig. 5. The dynamic bitrate and RTT of different schemes: (a) proposed scheme, (b) CC-AIMD, (c) CC-MIMD, and (d) BE.

Table 1 displays the bandwidth utilization and the queuing delay for all schemes. Fig. 5 shows the dynamic bitrate and RTT of all schemes. Table 1 and Fig. 5 show that, in the face of varying link capacity, the video bitrate of our scheme is more stable than the other three schemes, and the bitrate is the closest to the link capacity. The bandwidth utilization of our scheme is 92.6%, which is higher than the other three schemes. Because of the conservative detection of available bandwidth, CC-AIMD has a lower

queuing delay at the cost of lower bandwidth utilization. Our scheme achieves more than 8.4% bandwidth utilization at the cost of only 1.5 ms average queuing delay than CC-AIMD. CC-MIMD detects available bandwidth more quickly than CC-AIMD, which leads to more frequent queuing, and its average queuing delay is more than 2.3 times of our scheme. Besides, the value of the 95th percentile of the queuing delay is equal to 18.96 ms for our scheme whereas 43.99 ms for CC-MIMD. BE can achieve relatively stable bitrate and delay, but the bandwidth utilization is not high. Its efficiency depends on the accuracy of estimation and video control scheme. Our proposed scheme judges the improvement or reduction of quality through the network capacity, rather than responding after the congestion signal appears, or simply estimating the bandwidth.

Table 1. Bandwidth utilization and the values of the 5th, 50th, 95th percentiles of queuing delay with variable link capacity

	U (%)	D_q (ms)			
		5th	50th	95th	Average
Proposed scheme	92.6	0.57	1.66	18.96	4.7
CC-AIMD	84.2	0.57	1.38	13.66	3.2
CC-MIMD	91.6	0.59	3.17	43.99	11.12
BE	83.3	0.53	1.48	13.13	2.47

Table 2 shows the results when the three coefficients α, δ, and β take different values under the same experimental conditions. Compared with the previous values (α = 0.1, δ = 1, β = 0.2), it can be seen that the larger α, the greater the impact of current delay, which will improve the delay jitter and result in large fluctuation of the estimated value and video quality, and the accuracy of the algorithm is reduced. The smaller the value of α, the lower the delay sensitivity and adaptive efficiency of the algorithm. The smaller δ, the range of remission state is wider. Since the quality cannot be improved in the remission state, the conditions for quality improvement are more stringent, therefore the U and D_q are reduced. On the contrary, the larger δ makes more opportunities for quality improvement and network congestion, the U and D_q are increased. The smaller β makes the judgment that the quality can be increased more optimistic or frequent, which increases the probability that the bitrate exceeds the network, this leads to the increase of the U and D_q. The larger β makes the quality improvement more conservative and the lower U and D_q.

Table 2. The results when the α, δ, and β takes different values

α	δ	β	U (%)	Average D_q (ms)
0.05	1	0.2	92.5	6.04
0.5	1	0.2	90.5	8.28
1	1	0,2	88.7	34.23
0.1	0.5	0.2	90.8	4.57
0.1	5	0.2	92.9	6.03
0.1	1	0.1	92.9	6.59
0.1	1	0.3	90.5	3.52

Fig. 6 shows the bitrate variation of our proposed scheme in the case of bandwidth random transformation. In this experiment, the bandwidth is changed randomly every 10 seconds, and the

probability of increasing 1 Mbps, decreasing 1 Mbps and keeping unchanged is equal. The experiment lasted 500 seconds. The result shows that the bandwidth utilization is 92.7%, and the average delay is 6.92 ms. It can well achieve the purpose of adaptation.

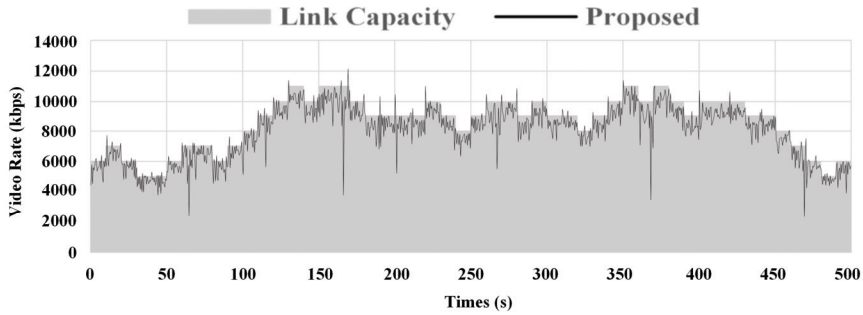

Fig. 6. The bitrate variation of our proposed scheme in the case of bandwidth random transformation.

4.3 Network Interruption and Recovery

In this experiment, we set the link capacity to 6 Mbps for the first 20 seconds, then we decreased the capacity to 0 for 5 seconds, finally the capacity is restored to 6 Mbps for 20 seconds. We consider the time it takes for the video rate to return to normal level. Fig. 7 displays the video bitrate in this scenario. The result shows that the video bitrate of our scheme returns to the normal level after about 1.5 seconds, while CC-AIMD takes about 10.5 seconds, CC-MIMD takes about 7 seconds and BE takes about 3 seconds, which means our scheme achieves faster bandwidth change response.

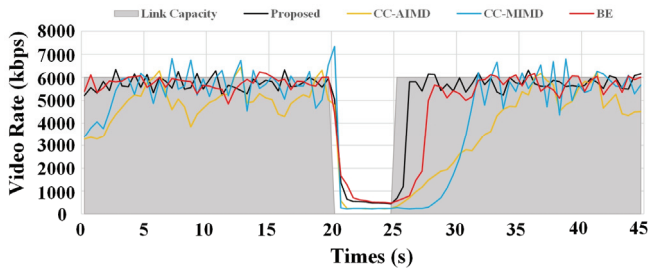

Fig. 7. The network is interrupted at the 20th second, and then is recovered at the 25th second.

5. Conclusion

In this paper, we analyze the significance of remaining network capacity for judging the current network state, and propose a scheme to transmit real-time video with intra-only coding over a wireless network. It adjusts the quality of frames through the estimation of remaining network capacity. We propose two formulas to estimate the remaining network capacity and the capacity needed to improve the video frame quality. Through these two formulas, the sender changes the quality of the video frames to achieve video adaptation. We compare our scheme with three different schemes, and the results show that our scheme can achieve better bandwidth utilization at the cost of less delay and faster response when

the bandwidth changes significantly. However, the proposed scheme uses the size of the last frame to estimate future frames. But there is a limitation to the video which uses inter-prediction. Because there is a large difference in the size of I, P, and B frames. In the future work, the adaptive transmission of inter-prediction video can be studied. For example, the size of last group of pictures may be used as information instead of the last frame, but this will reduce the response speed of adaptation. More suitable schemes can be researched in the future work.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61601070) and the Special General Program of Technology Innovation and Application Development of Chongqing (Grant No. cstc2020jscx-msxm1505).

References

[1] C. Bachhuber, E. Steinbach, M. Freundl, and M. Reisslein, "On the minimization of glass-to-glass and glass-to-algorithm delay in video communication," *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 238-252, 2018. https://doi.org/10.1109/TMM.2017.2726189

[2] S. Lei and Y. Jun, "Research on adaptive transmission mechanism of video data," in *Proceedings of 2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, Jabalpur, India, 2015, pp. 570-573. https://doi.org/10.1109/CICN.2015.117

[3] A. A. B. Ifit, O. Alaoui-Fdili, P. Corlay, F. X. Coudoux, and M. El Hassouni, "Transmission energy analysis and modeling of a video sensor node in the context of next generation WVSN," in *Proceedings of 2018 IEEE Conference on Antenna Measurements & Applications (CAMA)*, Vasteras, Sweden, 2018, pp. 1-4. https://doi.org/10.1109/CAMA.2018.8530585

[4] M. A. Labiod, M. Gharbi, F. X. Coudoux, P. Corlay, and N. Doghmane, "Cross-layer scheme for low latency multiple description video streaming over vehicular ad-hoc networks (VANETs)," *AEU-International Journal of Electronics and Communications*, vol. 104, pp. 23-34, 2019. https://doi.org/10.1016/j.aeue.2019.03.001

[5] Apple Inc., "HTTP Live Streaming (HLS)," c2023 [Online], Available: https://developer.apple.com/streaming/.

[6] "Information technology — Dynamic Adaptive Streaming over HTTP (DASH)," 2019 [Online], Available: https://www.iso.org/standard/79329.html.

[7] J. Yoon and S. Banerjee, "Hardware-assisted, low-cost video transcoding solution in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 3, pp. 581-597, 2020. https://doi.org/10.1109/TMC.2019.2898834

[8] Y. Li, S. Wang, X. Zhang, C. Zhou, and S. Ma, "High efficiency live video streaming with frame dropping," in *Proceedings of 2020 IEEE International Conference on Image Processing (ICIP)*, Abu Dhabi, United Arab Emirates, 2020, pp. 1226-1230. https://doi.org/10.1109/ICIP40778.2020.9190683

[9] B. Wang, F. Ren, and C. Zhou, "Hybrid control-based ABR: Towards low-delay live streaming," in *Proceedings of 2019 IEEE International Conference on Multimedia and Expo (ICME)*, Shanghai, China, 2019, pp. 754-759. https://doi.org/10.1109/ICME.2019.00135

[10] Y. F. Ou, Y. Xue, and Y. Wang, "Q-STAR: a perceptual video quality model considering impact of spatial, temporal, and amplitude resolutions," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2473-2486, 2014. https://doi.org/10.1109/TIP.2014.2303636

[11] C. He, Z. Xie, and C. Tian, "An adaptive QP adjustment of multimedia over heterogeneous wireless networks," *IOP Conference Series: Materials Science and Engineering*, vol. 719, no. 1, article no. 012028, 2020. https://doi.org/10.1088/1757-899X/719/1/012028

[12] T. Arsan, "Review of bandwidth estimation tools and application to bandwidth adaptive video streaming," in *Proceedings of International Conference on High Capacity Optical Networks and Emerging/Enabling Technologies (HONET)*, Istanbul, Turkey, 2012, pp. 152-156. https://doi.org/10.1109/HONET.2012.6421453

[13] X. Zhu and R. Pan, "NADA: a unified congestion control scheme for low-latency interactive video," in *Proceedings of 2013 20th International Packet Video Workshop*, San Jose, CA, USA, 2013, pp. 1-8. https://doi.org/10.1109/PV.2013.6691448

[14] G. Carlucci, L. De Cicco, S. Holmer, and S. Mascolo, "Congestion control for web real-time communication," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2629-2642, 2017. https://doi.org/10.1109/TNET.2017.2703615

[15] T. Dai, X. Zhang, and Z. Guo, "Learning-based congestion control for internet video communication over wireless networks," in *Proceedings of 2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence, Italy, 2018, pp. 1-5. https://doi.org/10.1109/ISCAS.2018.8351530

[16] Y. Hao and W. Fang, "The research on algorithm for congestion control based on video transmission," in *Proceedings of 2013 5th International Conference on Intelligent Networking and Collaborative Systems*, Xi'an, China, 2013, pp. 344-347. https://doi.org/10.1109/INCoS.2013.64

[17] 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. https://doi.org/10.1109/TCSVT.2007.905532

[18] C. Al Hasrouty, M. L. Lamali, V. Autefage, C. Olariu, D. Magoni, and J. Murphy, "Adaptive multicast streaming for videoconferences on software-defined networks," *Computer Communications*, vol. 132, pp. 42-55, 2018. https://doi.org/10.1016/j.comcom.2018.09.009

[19] 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 *Proceedings of 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Reston, WA, USA, 2018, pp. 267-282.

[20] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, 2003. https://doi.org/10.1109/TCSVT.2003.815165

[21] L. Wu, A. Zhou, X. Chen, L. Liu, and H. Ma, "GCC-beta: improving interactive live video streaming via an adaptive low-latency congestion control," in *Proceedings of 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6. https://doi.org/10.1109/ICC.2019.8761256

Shu Tang **https://orcid.org/0000-0001-7517-7992**

He received the M.E. degree in computer science from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2007, and the Ph.D. degree in Chongqing university, China, in 2013. He is currently an associate professor at the College of Computer Science and Technology at Chongqing University of Posts and Telecommunications, China. His research interests include signal processing, image processing, and computer vision.

Yuanhong Deng **https://orcid.org/0000-0001-6169-2133**

He received a B.S. degree at Chongqing University of Posts and Telecommunications, Chongqing, China, in 2019. Currently, he is studying for M.S. degree in the School of Computer Science and Technology at Chongqing University of Posts and Telecommunications. His research interests include wireless video transmission.

Peng Yang **https://orcid.org/0000-0002-0237-2938**

He received his bachelor's degree in engineering from Chongqing University of Posts and Telecommunications in 2020. He is currently studying for the M.S. degree in the School of Computer Science and Technology at Chongqing University of Posts and Telecommunications. His research interests are wireless video transmission.