A QoS Model to Identify Required QoS for Guaranteeing Quality of Internet Video Streaming Services

Anan Sawabe*† and Takanori Iwai*
* System Platform Research Laboratories, NEC Corporation
1753 Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa, Japan
† Graduate School of Interdisciplinary Information Studies, the University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
Email: { a-sawabe, takanora1 }@nec.com

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500711

***Abstract*—Understanding the required quality of service (QoS) for guaranteeing the necessary video quality for Internet video streaming services is important for enabling network operators to provide high-quality networks. Recent trace-based approaches that infer video quality through machine-learning based traffic-feature analysis have a drawback in terms of the maintenance cost for updating their analysis model due to the massive number of videos uploaded daily. In this paper, we construct a QoS analysis model that calculates the required throughput for delivering video with a certain resolution by using encoding information instead of traffic tracing. For accurate modeling, we consider and formulate communication overhead, i.e. streaming behavior, retransmission, and headers. Through experiments, we demonstrate that our model can perform in two use cases: (1) estimating effective resolution for arbitrary QoS, and (2) calculating the required QoS for guaranteeing the necessary video quality.**

***Index Terms*—QoS, QoE, Adaptive streaming, Performance evaluation technique**

I. Introduction

Video streaming services are one of the major applications in the Internet market, with 500 hours of video uploaded to YouTube (one such service) every minute around the world [1]. These services will generate a vast amount of traffic (over 84% of consumer IP traffic) by 2022 [2]. For network operators such as internet service providers (ISPs), mobile network operators (MNOs), and mobile virtual network operators (MVNOs), monitoring video quality and understanding the required quality of service (QoS) for guaranteeing the necessary video quality are important in order to provide high quality networks. Recent trace based approaches [3], [4], which infer video quality such as bitrate with machine learning based traffic feature analysis, have a shortcoming in terms of high maintenance cost due to the massive number of videos uploaded daily. Specifically, operators must prepare a huge amount of storage for traffic and video quality traces over the total amount of analyzed video content and update the analysis model for uploaded videos so as to stay up to date. In addition, recent efforts to model video quality are standardized in ITU-T Recommendations, e.g., the P.120X series [5]–[7]. In these standards, the video quality function, constructed through subjective experiments, outputs a mean opinion score (MOS) ($\in [1, 5]$) by analyzing video quality, e.g., bitrate, resolution, and stalls, but does not analyze QoS. We therefore address maintenance cost reduction by constructing an analysis model between QoS and video quality without traffic tracing.

MPEG-DASH [8], one of the de facto standards for adaptive bitrate (ABR) streaming over HTTP, selects the highest video resolution in real time in achievable end-to-end (E2E) throughput. The scheme gives us the simplest analysis model: to regard bitrate as required throughput. However, the simplest model assumes a content server delivers only video content to a client. In practical applications, video delivery via networks often incurs communication overhead. For example, we can observe the simultaneous transmission of multiple chunks with different resolutions. In addition, lost packets are retransmitted for data integrity. Further, headers are added to communicated packets. These overheads should be considered in order to model the required throughput accurately.

In this paper, we propose a QoS analysis model for Internet video streaming services. The model enables network operators to estimate the required QoS for guaranteeing the necessary video quality while reducing the maintenance cost for stable network operation. The proposed model uses video encoding information, including bitrate for each resolution, and calculates the required throughput for delivering video with a certain resolution by considering three communication overheads (streaming behavior, retransmission, and headers) instead of traffic tracing. The proposed model has two main features. The first is that it uses video encoding information instead of playing videos, which reduces the maintenance cost associated with playing videos and storing traffic traces for each video, unlike conventional trace-based approaches. Second, the model is applicable in multiple use cases: (1) calculating effective resolution from arbitrary QoS (in Section IV-B), and (2) calculating the required throughput for guaranteeing the necessary resolution for multiple videos (in Section IV-C).

There have been many studies aimed at understanding the relationship between video quality and traffic patterns. Some

978-1-7281-7122-7/21/$31.00 ©2021 IEEE

recent approaches are dedicated to estimating video quality (such as bitrate and stalls) from traffic features with machine learning techniques [3], [4]. These approaches, however, have shortcomings in terms of their maintenance cost, e.g., the required storage and traffic collection while playing videos. The closest work to our research is provided in the ITU-T Recommendations P.120X series standards [5]–[7], which calculates users' perceived quality as MOS by analyzing video playback quality such as bitrate and stalls. However, this work differs from ours in the following ways. First, our proposed model outputs the required throughput from video quality. Second, the prior work does not consider communication overhead.

In summary, our contributions are as follows.

- Constructing a QoS analysis model to calculate the required throughput from video encoding information considering communication overhead.
- Evaluating the performance of the proposed model through traffic shaping experiments and achieving 76.0% accuracy in estimating effective video resolution.
- Demonstrating our model in two use cases: (1) calculating effective resolution from arbitrary QoS, and (2) calculating the required throughput for guaranteeing the necessary resolution by analyzing 672 YouTube videos.

The structure of this paper is as follows. We first provide background for ABR streaming and related work in Section II. We then construct the proposed QoS analysis model in Section III and evaluate our model's performance in two use cases in Section IV. After that, we discuss an extension of our model for calculating quality of experience (QoE) by combining it with ITU-T standards in Section V. Finally, Section VI concludes this paper.

II. Related Work

In this section, we give a brief overview of ABR streaming and then discuss related work.

A. ABR streaming

In ABR streaming, video is encoded in the available resolution set. The value of video size [b] divided by its duration [s] for each resolution is called the average bitrate [bps]. In addition, encoded video is divided into several parts, called chunks. From the viewpoint of communication, a client requests a video encoded with the highest video quality available under E2E throughput from the content server of an over-the-top (OTT) service provider, and the server then delivers the video chunks with the requested quality. For example, YouTube videos are delivered over HTTPS sessions, and the client communicates a request including the video ID and its quality by specifying the request information in the URL field in the HTTP request header. Finally, the client stores the received chunks and then decodes and plays back the chunks with the maximum resolution in the playback buffer.

B. Related work

Many studies have been dedicated to understanding Internet video streaming services. In particular, the major OTT service provider YouTube has garnered attention from various research fields, such as traffic characteristics and modeling [9], [10] and encrypted traffic analysis [3], [4], [11]. Some researchers have proposed estimating video-playback quality from communication-traffic features (e.g., [3], [4]). Pan et al. [3] have proposed the extraction of traffic features such as statistics of throughput and packet size from observed traffic traces during video playback, followed by the identification of video quality, e.g., bitrate and resolution, with machine learning. Gutterman et al. [4] have proposed the categorization of traffic into chunks before extracting the traffic features for accurate identification. These methods are dedicated to identifying video playback quality at the time the video is played. In terms of network operation for controlling video traffic, however, network operators must attain the required QoS for guaranteeing the necessary video quality. Thus, our target is to calculate the required QoS from arbitrary video quality. In addition, past studies do not discuss the maintenance cost for traffic tracing and video playing. The cost of preparing storage for traces is prohibitive due to the massive number of videos being uploaded daily, so we aim to construct a QoS analysis model without tracing.

The closest work to our research is an opinion model in the ITU-T Recommendations P.120X series [5]–[7]. According to P.1201 [5] and P.1202 [6], an opinion model is proposed to estimate MOS of users' perceived quality from video playback information (e.g., bitrate, resolution, and stalls) for constant bitrate (CBR) streaming services. In addition, an opinion model in P.1203 [7] analyzes time-series video-playback quality to calculate MOS for ABR streaming services by considering QoE degradation caused by time-series quality changes. The P.120X standards differ from our work in relation to network operation in the following two ways. First, our proposed model outputs the required throughput from video quality for network operation. Second, the communicated traffic includes not only video content but also communication overhead, so we can consider and formulate the overhead to construct an accurate model.

III. Proposed Method

In this section, we construct a QoS analysis model for calculating the required QoS for guaranteeing the necessary video quality. We first describe the problem formulation (in Section III-A) and then discuss each component of our proposed model.

A. Problem formulation

We start by considering the required throughput when delivering a video with a certain resolution. Given a video with T-second duration, content length L_i [b], and resolution ξ_i, average bitrate β_i [bps] is calculated by L_i/T. The range of i depends on the space size of the available resolution. For example, the available resolution set in some YouTube videos

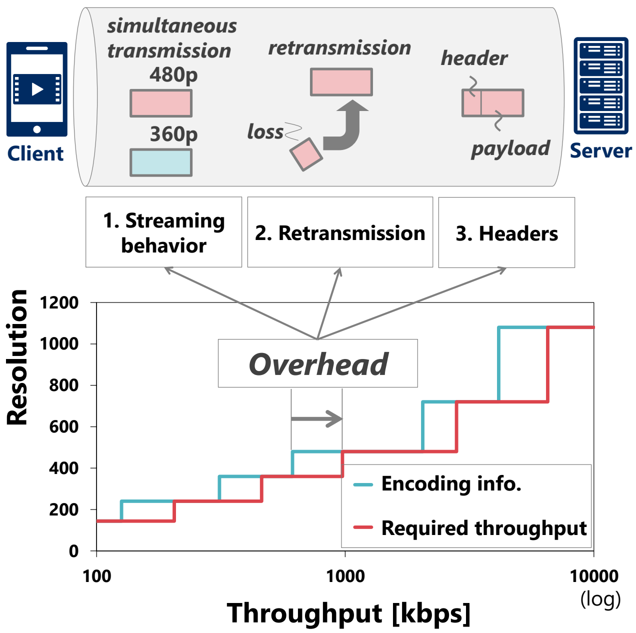

Fig. 1. Required throughput with communication overhead.

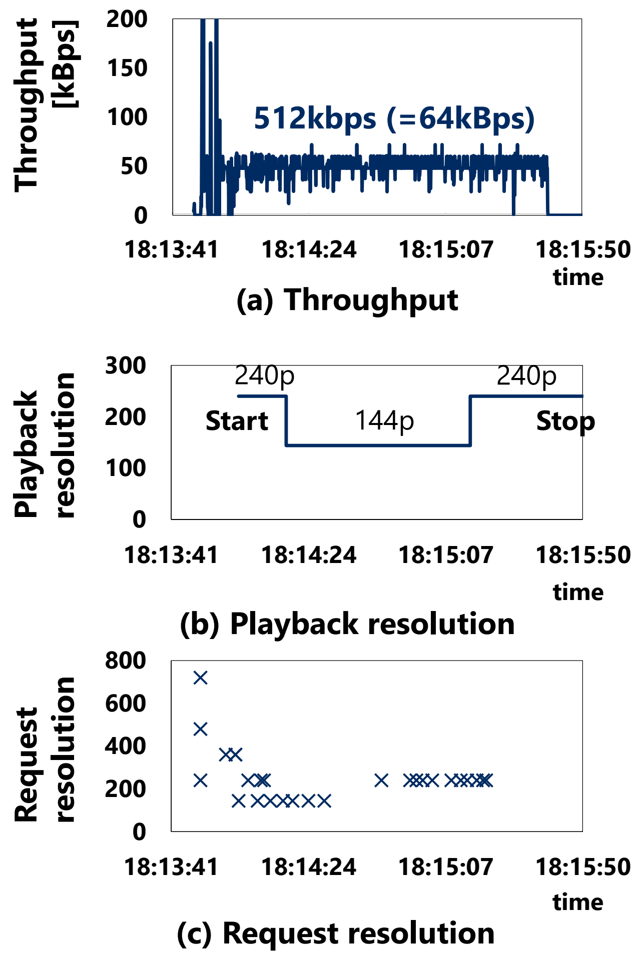

Fig. 2. Streaming behavior.

is defined as {144p, 240p, 360p, 480p, 720p, 1080p}, that is, $i = 1, 2, \cdots, 6$.

When the video is transmitted from a content server to a client, networks need to guarantee available E2E throughput over not only β_i but also communication overhead, which can be broken down into the following three components. The first component is streaming policy overhead. We observe the behavior of the client downloading multiple chunks with different resolutions at the same time, which results in increased traffic γ_{policy} [bps] due to simultaneous transmission (in Section III-B). The second component is retransmission overhead. When packets are lost, they are retransmitted to guarantee data integrity, and the traffic of γ_{loss} [bps] increases (in Section III-C). The third component is header overhead. For example, YouTube videos are transmitted via TCP/HTTPS [12] or UDP/QUIC [13] sessions. Packet headers are added in each communication layer so that the traffic of γ_{header} [bps] increases (in Section III-D). We illustrate this concept in Fig. 1.

In summary, required throughput γ_i for transmitting the video with ξ_i can be calculated by the following equation:

$$\gamma_i = \beta_i + \gamma_{\text{policy}} + \gamma_{\text{loss}} + \gamma_{\text{header}}. \tag{1}$$

We discuss each of the three components in the following sections.

B. Streaming behavior overhead (γ_{policy})

To consider streaming behavior overhead, we first show the results of chunk-request behavior through traffic shaping with a data rate of 512 kbps while watching a YouTube video in Fig. 2. As the results show, the client requests three chunks with different resolutions (240p, 480p, and 720p) at the start time of playback. After that, 240p and 144p chunks are requested in parallel. Finally, only 240p chunks are requested. Although some requests are simultaneously transmitted, a media player on the client device plays back the video with only the maximum resolution stored in the playback buffer. Thus, the data volume of the unused chunks is overhead.

However, accurate overhead calculation requires GET request tracing, which contradicts our original goal of maintenance cost reduction. We therefore roughly approximate the behavior from the viewpoint of streaming policy. We again review the high-level ABR streaming policy, that is, delivering maximum video quality under available E2E bandwidth to provide higher-quality videos. On the basis of the policy and practical behavior discussed above, we assume that a video-streaming server transmits not only video chunks with playback resolution ξ_i but also chunks with one-level-lower resolution ξ_{i-1} at least. Therefore, we formulate the overhead as follows:

$$\gamma_{\text{policy}} = \beta_{i-1}. \tag{2}$$

C. Retransmission overhead (γ_{loss})

Packet loss occurs because of low communication quality, e.g., bit error and buffer overflow, so lost packets are retransmitted to guarantee data integrity. Here, we formulate the retransmission overhead.

When the packet-loss rate is ρ, we calculate the retransmission overhead with the following equation:

$$\gamma_{\text{loss}} = \beta p_{\text{nloss}} + \sum_{k=1}^{n}(k+1)\beta p_{\text{loss}}^{k} - \beta, \tag{3}$$

where n is the maximum amount of loss for the same packets, p_{nloss} is the probability of packet receipt without loss, and p_{loss} is the loss probability, the same as ρ. p_{nloss} can be calculated as the following equation:

$$\begin{aligned} p_{\text{nloss}} &= 1 - (\rho + \rho^2 + \cdots + \rho^n) \\ &= 1 - \rho\frac{1-\rho^n}{1-\rho} \\ &= \frac{\rho - 2\rho + \rho^{n+1}}{1-\rho}. \end{aligned} \tag{4}$$

The second term of Eq. (3) is calculated as follows.

$$\sum_{k=1}^{n}(k+1)\beta p_{\text{loss}}^{k} = \beta\left(\frac{\rho-\rho^{n+1}(n+1)}{1-\rho} + \frac{\rho(1-\rho^n)}{(1-\rho)^2}\right) \tag{5}$$

Finally, Eq. (3) is calculated by replacing p_{nloss} and p_{loss} in Eq. (4) with those in Eq. (5).

$$\begin{aligned} \gamma_{\text{loss}} &= \beta\frac{1-\rho+\rho^2-(n+1)\rho^{n+1}+n\rho^{n+2}}{(1-\rho)^2} - \beta \\ &= \beta\left(\frac{1-\rho+\rho^2}{(1-\rho)^2} - 1\right)(n\to\infty) \end{aligned} \tag{6}$$

D. Packet header overhead (γ_{header})

Finally, we formulate the overhead caused by packet headers. In ABR streaming services, a video is divided into multiple chunks, and each chunk is then also fragmented into a maximum transmission unit (MTU) for communication from a content server to a client. In addition, each packet has a header in each communication layer. Therefore, when the header size is η [b], we calculate the header overhead as the following equation:

$$\gamma_{\text{header}} = \beta\frac{\eta}{MTU}. \tag{7}$$

IV. Evaluation

In this section, we give the experimental results of our proposed model from three perspectives. First, we evaluate the model's accuracy through traffic shaping experiments. Second, we show results for a use case of estimating effective video quality from arbitrary QoS, which can be used as a quality index of network performance. Third, we show results for another use case of estimating the required QoS from arbitrary video quality, which can be used as a baseline QoS for network operation such as traffic shaping.

In this paper, we evaluate our proposed model with YouTube videos for the following two reasons. First, YouTube is one of the largest OTT service providers in the market of Internet video streaming services. Second, YouTube continuously updates their streaming techniques, so we can evaluate if our model is applicable to operational state-of-the-art streaming techniques.

A. Model accuracy for effective video quality

We first evaluate the accuracy of our model through experiments to estimate effective video quality under traffic shaping. We collect the effective resolution by playing N (=10) YouTube videos while shaping video traffic with the predefined data rates r_k, $k = 1, \cdots, 5, K = 5$, i.e. 256 kbps, 512 kbps, 1024 kbps, 2048 kbps, and 4096 kbps, while playing a video. As a comparative method, we use the simplest approach to select the effective resolution with the maximum *bitrate* lower than shaping rate r_k.

Our model requires the bitrate for each resolution, so we collect video encoding information for the ten videos using the get_video_info[1] file that describes video encoding information, e.g., bitrate for each available resolution. On the basis of the collected encoding information, we estimate effective resolution $\hat{\xi_{n,k}}$ for video n by selecting resolution ξ_i with maximum required throughput γ_i lower than shaping rate r_k.

We calculate accuracy as a performance measure. Accuracy is calculated by

$$\text{Accuracy} = \frac{1}{NK}\sum_{k=1}^{K}\sum_{n=1}^{N}(1 - \text{Sign}(|\hat{\xi_{n,k}} - \xi_{n,k}|)). \tag{8}$$

For the ten videos, the estimation accuracy of the compared method and our model are 0.18 and 0.76, respectively. The reason that our model improves the accuracy of effective resolution estimation is evident in the results given in Fig. 3. As the results show, our model estimates a lower resolution than the estimated resolution of the compared method. As described in Section III, the communicated traffic between the client and server assessed in our method includes not only video content but also communication overhead, which results in our method performing more accurate estimation than the compared method.

B. Case 1: Mapping QoS to effective resolution

Usually, customers who form a contract with an ISP or MNO/MVNO check the network speed, i.e. throughput, as a performance measure. However, throughput does not indicate quality in the application layer, so we believe effective video quality in operators' networks will also be used as a performance measure for consumers' decisions in the future. In addition, for network operators, understanding effective video quality in their networks is important for network monitoring. Therefore, we next show the results of experiments to estimate effective resolution from arbitrary QoS.

Here, we calculate the effective resolution when throughput x and loss rate ρ are given. First, required throughput γ_i for each resolution ξ_i is calculated from collected video encoding

[1] https://www.youtube.com/get_video_info?video_id=VIDEO_ID

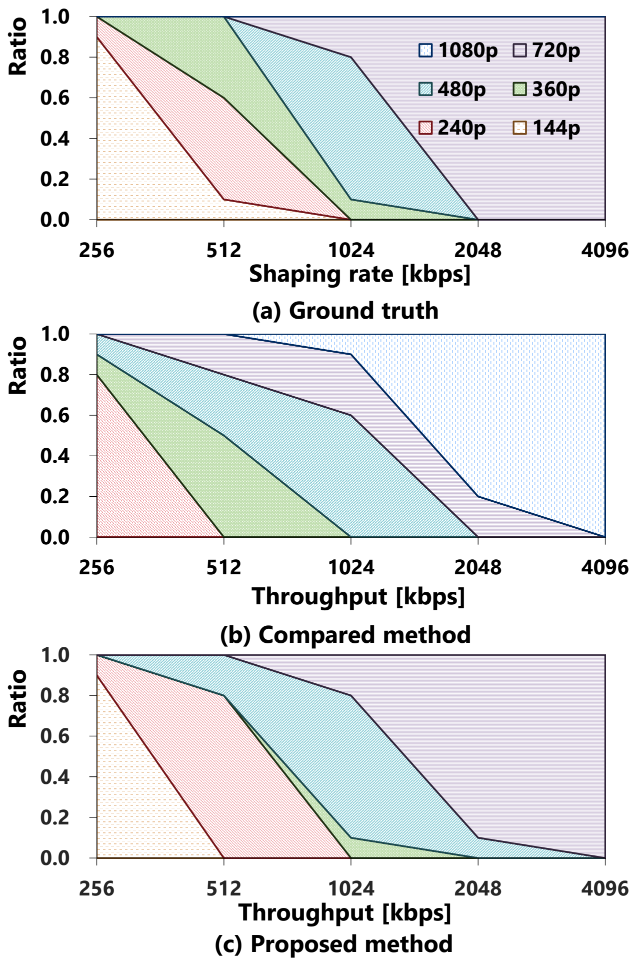

Fig. 3. Results of effective resolution ratio for each shaping rate. Each area shows the ratio of videos with its resolution. For example, for the shaping rate 1024 kbps, 10% of videos are 360p, 70% of videos are 480p, and 20% of videos are 720p.

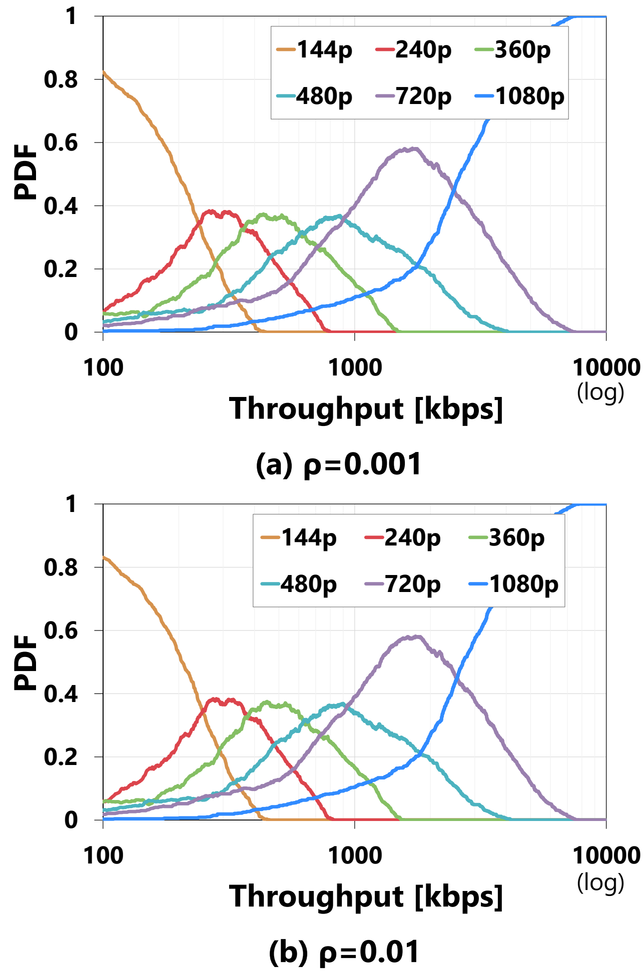

Fig. 4. Results of effective resolution distribution.

information and ρ with Eq. (1). Note that we collect 672 get_video_info files in this experiment. From the results, we can calculate the effective resolution of each video by selecting resolution ξ_i with maximum required throughput γ_i lower than x in a similar manner to that in Section IV-A. After that, we calculate the probability density function (PDF) of effective resolution. The results are shown in Fig. 4. For example, we can see that a network with x = 1 Mbps and ρ = 0.001 can provide about 40% of videos with 720p resolution.

C. Case 2: Mapping required video quality to required QoS

Finally, we show results for an experiment to estimate required QoS from required video quality. This use case is quite important for network operators to control video traffic while guaranteeing QoE for their customers.

The approach is simple. We use the PDF constructed in the previous section IV-B. When the required resolution is ξ^*, we categorize videos with resolution ξ_i lower than ξ^* into a *dissatisfying* class and the other videos into a *satisfying* class. We show the results of the ratio of satisfying required video quality in Fig. 5. For example, when the requirement is that over 95% of videos have a resolution of over 360p, the required throughput can be calculated to 806 kbps.

In summary, we have shown the results of the model's accuracy in Section IV-A and have demonstrated that our model performs in two important use cases: calculating effective resolution in given QoS in Section IV-B, and calculating the required QoS for guaranteeing the necessary resolution in Section IV-C. Our model could help network operators understand the relationship between QoS and video quality.

V. Discussion

We have evaluated how to calculate effective video resolution from QoS with our proposed model in Section IV-B. QoE is also important as a performance metric as standardized in ITU-T, so in this section, we describe a method for calculating effective QoE by combining it with the ITU-T Recommendations P.120X series.

When we apply our model to calculate effective resolution with a constant throughput and loss for each video, as discussed in Section IV-B, we can regard the results as

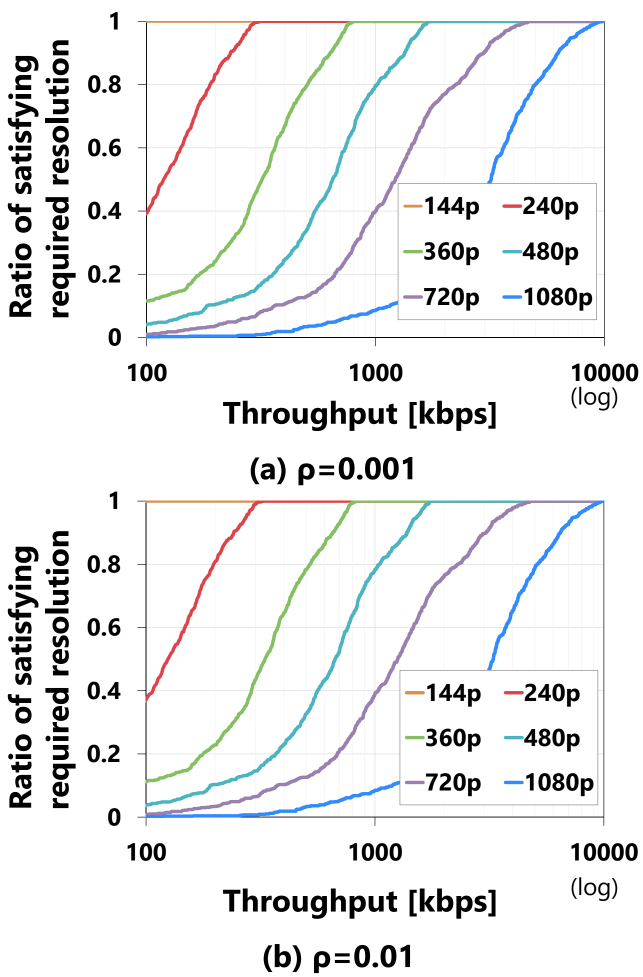

Fig. 5. Results of ratio of satisfying required resolution. Each line shows the ratio of videos with a resolution higher than the required resolution.

the effective resolution for *CBR* streaming. Therefore, we can calculate MOS from P.1201 and P.1202 by inputting the effective resolution and its bitrate.

Although we can assume CBR-like behavior when video traffic is shaped with a constant rate, video quality changes in real time with dynamic QoS changes in cases without traffic shaping. In such cases, MOS can be calculated as follows. First, arbitrary time-series QoS changes including throughput and loss rate are prepared. Then, the time-series effective resolution replaced from ξ_i to $\xi_{i,t}$ is calculated using the prepared time-series QoS changes. Finally, MOS is calculated from P.1203 using $\xi_{i,t}$ and its bitrate $\beta_{i,t}$.

VI. Conclusion

In this paper, we propose a QoS analysis model for Internet video streaming services. The model enables network operators to estimate the required QoS for guaranteeing the necessary video quality while reducing the maintenance cost for stable network operation. The proposed model uses video encoding information and then calculates required throughput for delivering video with a certain resolution by considering communication overhead (i.e. streaming policy, retransmission, and headers) instead of traffic tracing. In experiments, our model estimates effective resolution under traffic shaping without tracing and achieves an accuracy of 0.76. In addition, we demonstrate that our model can perform in two use cases: (1) estimating effective resolution for arbitrary QoS, and (2) calculating the required QoS for guaranteeing the necessary video quality.

References

[1] Statista, “Hours of video uploaded to youtube every minute as of may 2019,” 2019. [Online]. Available: https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/

[2] Cisco, “Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022 white paper,” 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html

[3] W. Pan, G. Cheng, H. Wu, and Y. Tang, “Towards qoe assessment of encrypted youtube adaptive video streaming in mobile networks,” in *Proceedings of the IEEE/ACM 24th International Symposium on Quality of Service*, ser. IWQoS ’16, 2016, pp. 1–6.

[4] C. Gutterman, K. Guo, S. Arora, X. Wang, L. Wu, E. Katz-Bassett, and G. Zussman, “Requet: Real-time qoe detection for encrypted youtube traffic,” in *Proceedings of the 10th ACM Multimedia Systems Conference*, ser. MMSys ’19, 2019, pp. 48–59.

[5] ITU-T, “P.1201: Parametric non-intrusive assessment of audiovisual media streaming quality,” 2012. [Online]. Available: https://www.itu.int/rec/T-REC-P.1201/en

[6] ——, “P.1202: Parametric non-intrusive bitstream assessment of video media streaming quality,” 2013. [Online]. Available: https://www.itu.int/rec/T-REC-P.1202/en

[7] ——, “P.1203: Parametric bitstream-based quality assesment of progressive download and adaptive audiovisual streaming services over reliable transport,” 2017. [Online]. Available: https://www.itu.int/rec/T-REC-P.1203/en

[8] “MPEG-DASH,” 2012, https://mpeg.chiariglione.org/standards/mpeg-dash.

[9] J. J. Ramos-munoz, J. Prados-Garzon, P. Ameigeiras, J. Navarro-Ortiz, and J. M. Lopez-soler, “Characteristics of mobile youtube traffic,” *IEEE Wireless Communications*, vol. 21, no. 1, pp. 18–25, February 2014.

[10] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, and J. M. Lopez-Soler, “Analysis and modelling of youtube traffic,” *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 4, pp. 360–377, 2012.

[11] F. Li, J. W. Chung, and M. Claypool, “Silhouette: Identifying youtube video flows from encrypted traffic,” in *Proceedings of the 28th ACM SIGMM Workshop on Network and Operating Systems Support for Digital Audio and Video*, ser. NOSSDAV ’18, 2018, pp. 19–24.

[12] Z. Durumeric, Z. Ma, D. Springall, R. Barnes, N. Sullivan, E. Bursztein, M. Bailey, J. A. Halderman, and V. Paxson, “The security impact of https interception,” in *Proc of the NDSS ’17*, 2017, pp. 1–14.

[13] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, J. Bailey, J. Dorfman, J. Roskind, J. Kulik, P. Westin, R. Tenneti, R. Shade, R. Hamilton, V. Vasiliev, W.-T. Chang, and Z. Shi, “The quic transport protocol: Design and internet-scale deployment,” in *Proc. of the ACM SIGCOMM ’17*, 2017, pp. 183–196.