Synthetic Traffic Generation for Streaming Video to Model IPTV

Abu (Sayeem) Reaz[1], Daisuke Murayama[2], Ken-Ichi Suzuki[2], Naoto Yoshimoto[2], Glen Kramer[3], Biswanath Mukherjee[1]
[1]University of California, Davis, CA 95616, USA
[2]NTT Access Network Service Systems Laboratories, Kanagawa, Japan
[3] Broadcom, USA
Email: asreaz@ucdavis.edu,gkramer@broadcom.com,
{murayama.daisuke, suzuki.kenichi, yoshimoto.naoto}@lab.ntt.co.jp

***Abstract*—With the rapid development of access networks, Internet Protocol Television (IPTV) has become a popular streaming video delivery technology. However, streaming IPTV over a network is challenging as the data rate of a video streaming varies with scene, time, frame-rate, and encoding. As a result, it is important to evaluate the performance of an access network for IPTV traffic to ensure better experience for users. Using video traces for such evaluation requires different videos traces to represent different scenarios and needs slower I/O-based operations, such as file reading. To overcome these limitations, in this paper, we explore several aspects of modeling IPTV streaming and present a synthetic video trace generator that represents the IPTV streaming. Our analysis shows that our synthetic video trace follows the statistical properties of empirical videos and its resultant data rate is representative of the IPTV video streaming.**

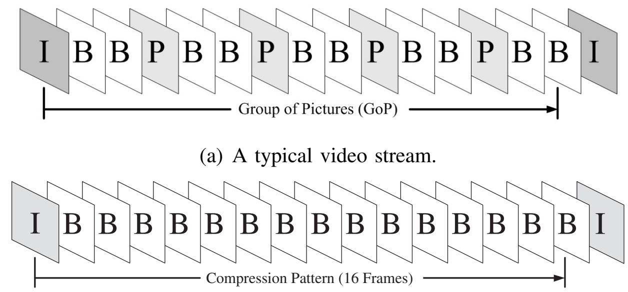
Keywords: IPTV, video modeling, synthetic video trace.

I. Introduction

Internet Protocol Television (IPTV) service includes streaming video over an IP network to be delivered to the end users. An end user typically receives the IPTV services over an access network, so it is important to study the performance of IPTV over an access network. Such a study can estimate important parameters such as number of concurrent video channels, network properties (e.g., buffer size), required degree of video compression, etc. Although there are studies that evaluate the performance of IPTV after a field deployment [1], it is important to study these parameters before deploying the service in order to evaluate alternative network architectures or to configure the network for better performance.

Typically, IPTV represents cable-like services (e.g., Comcast) with broadcast or multi-cast channels over IP. Such service provides a set of channels to users and each user selects a channel to view. However, there are several other video distribution architectures that are often termed as IPTV such as Peer-to-Peer (P2P) video applications and WebTV/ VoD where the server (may be more than one) holds contents and users select specific content to view [2]. In our work, we mostly focus on broadcast channel service.

Popular encoding methods for streaming video include MPEG (Moving Picture Experts Group) and H.264 SVC (Scalable Video Coding) [3]. The encoding usually divides the periodic video frames into three major categories: I, P, and B frames [4]. Figure 1(a) shows a typical video compression pattern. The I frame (Intra-coded frame) contains a complete image, like a conventional static image file. The P frame (Predicted frame) carries the changes in the image from the previous I or P frame. The B frame (Bi-predictive frame) uses differences between the current frame and both the preceding and following frames. Thus, P and B frames contain only part of the image information, so they provide video compression and lower the bit rate for streaming. The group of all frames from one I frame until and excluding the next I frame together is called a Group of Pictures (GoP) as shown in Fig. 1(a). GoP specifies the arrangement of I, B, and P frames.



(a) A typical video stream.

(b) A single-layer SVC coded video.

Fig. 1. The compression pattern of a video stream.

To match today's requirements, IPTV service needs to be High Definition (HD), bandwidth efficient (to accommodate the delivery of HD quality video over an access network), and adaptive (to the network condition). H.264 SVC can inherently deliver these properties as the average bit rate of an H.264 SVC encoding is typically half of the average bit rate of the corresponding MPEG-4 Part 2 encoding for the same video quality. These favorable properties render H.264 SVC a strong candidate for the encoding of IPTV services.

A streaming HD video using H.264 SVC single-layer coding can be constructed with a number of I, P, and B frames [3], [5] in a GoP. For H.264 SVC, the highest compression can be achieved through the video compression that have 16 frames in a GoP with one I frame and 15 B frames without any P frame as shown in Fig. 1(b). This video compression is recommended

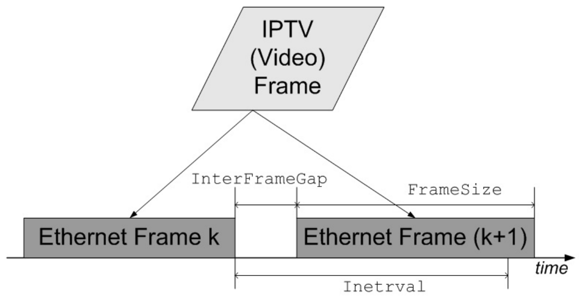


Fig. 2. Conversion of a video frame into Ethernet frames.

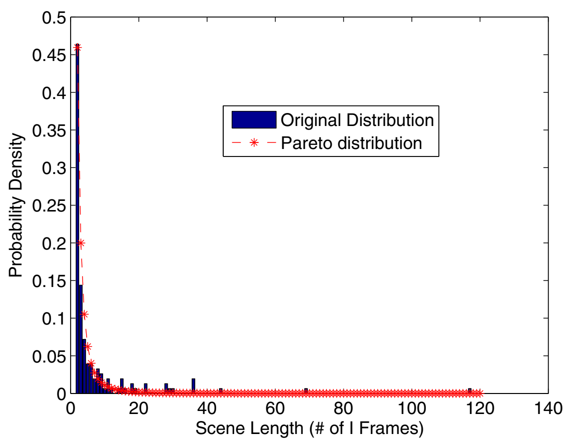


Fig. 3. Scene length distribution and its approximation.

for H.264 SVC [3]. Trace for such a video with its statistical and compression properties can be found in [6].

Synthetic video trace generation requires a proper model that represents an empirical video closely. Our work focuses on modeling the IPTV because, once we obtain such a model, it can be easily distributed, it can be used for performance evaluation, and the results can be reproduced anywhere.

II. Modeling IPTV Services

The modeling of IPTV services to evaluate the performance of an access network is a two-step process: 1) finding a model that accurately represents IPTV behavior and allows fast traffic generation, and 2) integrating the traffic model with an access network performance evaluator. We elaborate on each of these steps below.

A. Modeling

Modeling IPTV traffic is challenging because videos are unique and their statistical properties vary. The problem becomes more challenging because the same video stream has both temporal and spatial variations (randomness). For example, a video comprises of different scenes that vary with time and motion and one GoP is usually different from another GoP within the video. Moreover, within a GoP, there are numbers of different I, B, and P frames, each encoded differently and exhibits different statistical properties. But capturing these properties may not be sufficient for modeling because frame sizes within and across a scene are strongly correlated. So, finding the scenes and their correlation is important.

There are a number of modeling options. The first option is to create an independent model to represent IPTV. A number of such models are available in the literature [7], [8]. Usually, these models are multidimensional to capture the GoP-level representation of the variation of information with time and motion. They also have frame-level representation for each GoP. These models may differ based on encoding techniques. The benefits of such models are that they can generate synthetic video traffic algorithmically and are easily distributable. But accurate models are complex and computationally expensive [7] and not suitable for integration with a network performance evaluation platform. If the model is simpler [8], then it cannot capture the characteristics of video streams accurately.

The second option is to use video trace files [3], [5]. Many video traces are publicly available [5], they can be properly formatted to generate accurate video traffic, and they may fit any network-evaluation platform. However, using video traces for network performance evaluation has two major limitations: i) as it requires file I/O, it makes the evaluation process significantly slower, and ii) additional trace files make the evaluation tool large and distribution of the tool becomes cumbersome.

The third option is a hybrid approach: generating synthetic video trace that follows the properties of a empirical video trace closely [9]. This approach requires comparison of known distributions (e.g., Lognormal) with empirical data. The distribution that provides the best fit can be used to generate video traffic. The hybrid approach is easy to code and it follows the statistical properties of a video, so it overcomes the limitations of the other two options. However, synthetic video generation has several challenges including modeling the I and B frame size distribution, capturing the correlation within a GoP and among the GoPs, and capturing the overall statistical properties of a video since each video is different and has its own statistical properties. For example, a video for an action movie will be different from a video of a news broadcast. So videos should be modeled differently for different categories. Some of the general categories can be sports, news, action movie, slow movie, etc.

Our objective is to model video streaming as accurately as possible while keeping the model simple and easy to code. So, we choose the third option: using a hybrid approach to generate synthetic video trace. In our video trace generation, we use the compression pattern shown in Fig. 1(b). We consider the video given in [6] because it is a HD quality video, has the highest compression, and can be used to approximate HD IPTV.

B. Integration

As Ethernet is the most popular access network technology, in this section, we focus on how our proposed IPTV traffic model can be converted to Ethernet frames for the performance evaluation of an access network. The synthetic video trace

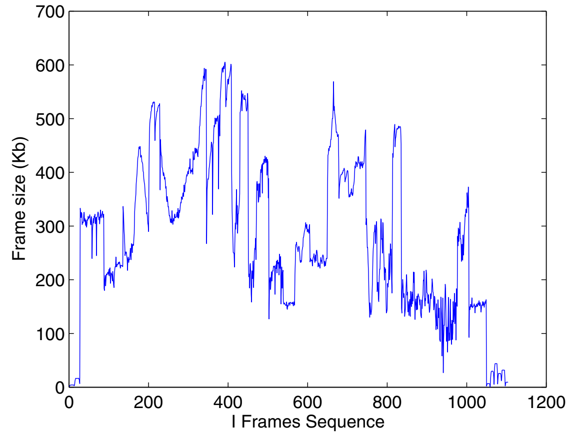


Fig. 4. I frame sequence of the trace.

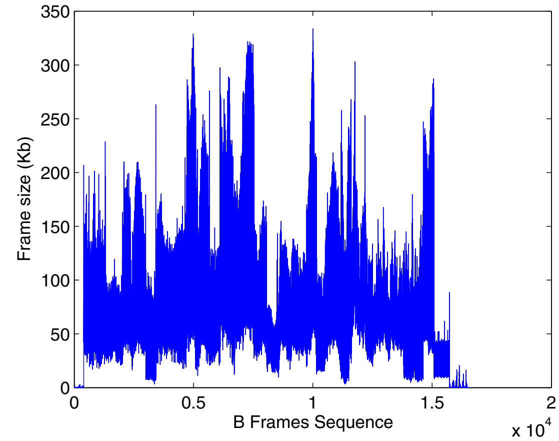


Fig. 5. B frame sequence of the trace.

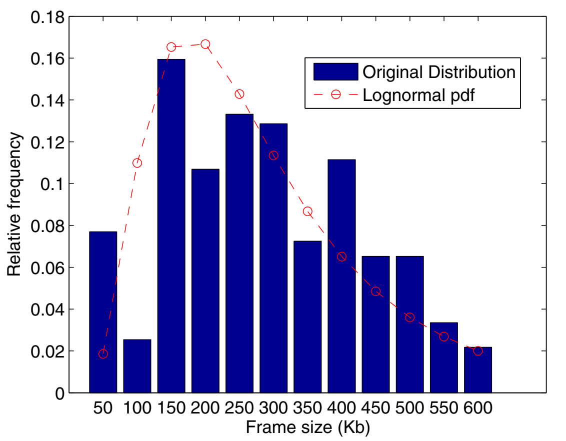


Fig. 6. Histogram of I frames and fitting Lognormal distribution.

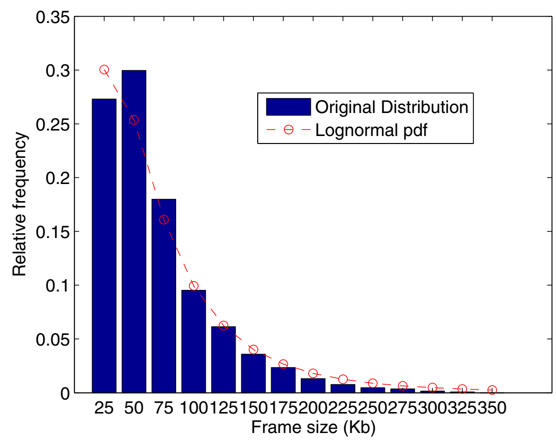


Fig. 7. Histogram of B frames and fitting Lognormal distribution.

generator creates video frames based on a deterministic frame rate given in frames per second (FPS), but the video frames can have different sizes. As video frames are large, each video frame, depending on its size, is packed over a number of consecutive Ethernet frames with minimum inter-frame gap (12 bytes) between them. So the synthetic video trace generator can be converted to a video traffic generator. Each video frame is generated in a regular time interval based on its FPS and for each video frame all the corresponding Ethernet frames will arrive consecutively. Thus, the video traffic generator will have a periodic burst of Ethernet frames, depending on the corresponding video frame size. Figure 2 shows how video frames are converted to Ethernet frames.

III. Scene Length and Frame Size Distribution

For our hybrid approach, we analyze the video frame distribution of a video trace [6] and extract important properties of the video such as scene length and video frame size distribution.

A. Scene Length Distribution

A video usually consists of a number of scenes. Scene changes in a video largely determine the statistical properties of the video. How I frames (and B frames) of similar sizes are grouped together in a video is determined by the scene length distribution. Scene length distribution thereby determines how I-frames and B-frame sizes are related to the characteristics of the video.

As I fames contain detailed scene information, if there are significant differences between two consecutive I frames, we can consider such a situation to be a scene change [9]. So, the number of consecutive I frames can be considered as the scene length. As scene lengths are independent, they can be approximated using independent identically distributed (*iid*) variables.

We extract the statistical distribution of scene length from an actual video trace in [6]. We observe that the scene length distribution shows similar behavior as a generalized Pareto distribution with tail index parameter $K = 0.55$, scale parameter $\sigma = 0.5$, and threshold parameter $\theta = 0$, as shown in Fig. 3. This Pareto distribution can be used to generate the scene lengths of the synthetic video trace.

B. Frame Size Distribution

A video stream shows a complex correlation structure resulting from two types of frames in a single stream [9]. In

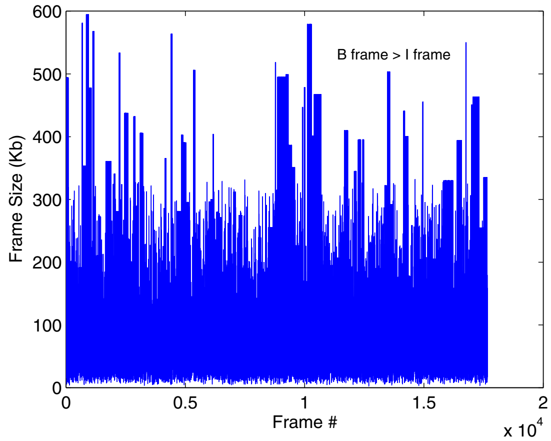


Fig. 8. Resultant synthetic trace using Lognormal distribution.

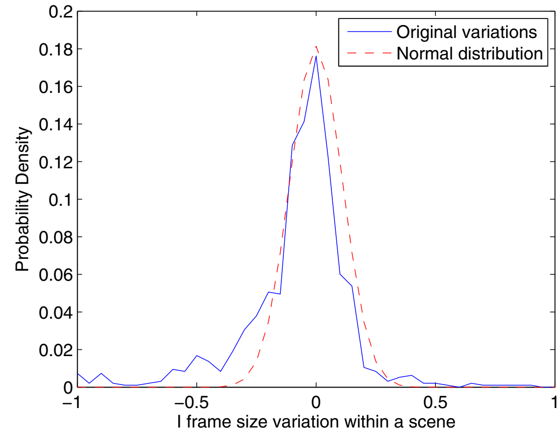


Fig. 9. Variation of I frame size within a scene.

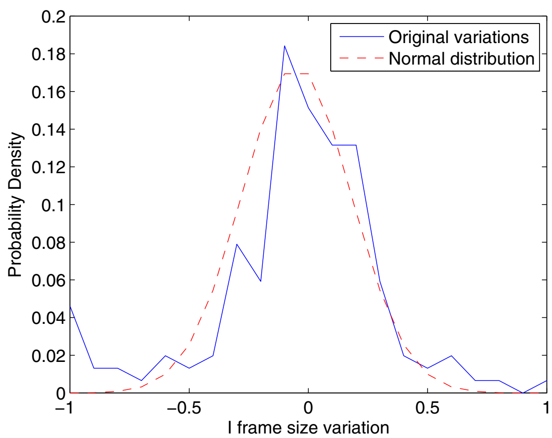


Fig. 10. Relative size distribution of the first I frames in a scene.

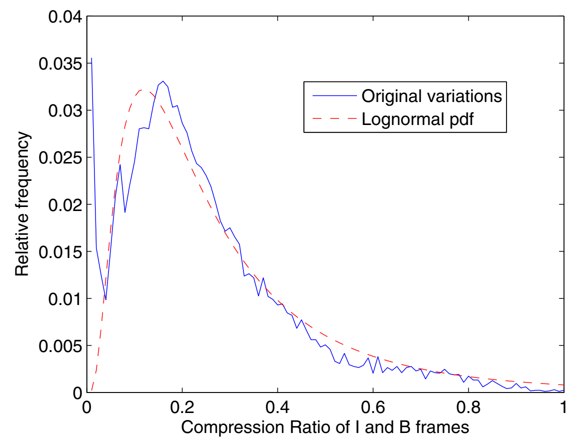


Fig. 11. Relative B frame sizes with respect to the I frame in a GoP.

order to visualize the impact, we decompose the video stream into two sub-streams, each containing a single type of frame. Figures 4 and 5 show the sequences of frame sizes for I and B frames, respectively. They exhibit different traffic patterns and statistical properties even though they are from the same video stream. Therefore, the sequences of I frames and B frames are modeled separately before being combined into an aggregated synthetic stream.

The frame-size distribution of the video stream [6] is created to analyze the behavior of I and B frames. Figures 6 and 7 show the distribution of I and B frames in the video trace, respectively. They show that I and B frame sizes from the same video have different distributions. They also show that Lognormal distribution with different shaping parameters, σ and μ, can be used to closely approximate the frame size distributions. Following the strategy described in [9], we discovered that, for the given video, the frame size distribution follows closely the Lognormal distribution with $\sigma = 0.595$ and $\mu = 5.5188$ for I frames, and $\sigma = 0.791$ and $\mu = 4.038$ for B frames.

IV. Synthetic Trace Generation

In this section, we discuss the challenges of synthetic video trace generation and how these challenges can be addressed to design an algorithm that generates synthetic video trace that follows actual video trace closely.

A. Using Basic Statistical Modeling

1) Trace Generation: Using the statistical distributions described in Sections III-A and III-B, we can model a synthetic video trace generator. As the I frame sizes in each scene should remain close to each other, we keep the I frame size in each scene the same. First, we obtain the scene length in terms of number of I frames from Pareto distribution. Then, we obtain an I frame size from Lognormal distribution, and for entire the scene length, we keep the I frame size same. For each I frame, we generate 15 B frames using Lognormal distribution.

2) Limitations: Such basic statistical modeling is not sufficient to represent IPTV traffic. The I frame distribution and B frame distribution of the generated synthetic trace are shown in Fig. 8 where we observe the impact of scene length. As we keep the same I frame size for each scene, we see "flat-tops" for the I frame sequences in Fig. 8 for each scene, as expected. We also observe that I and B frame sizes are not correlated within a GoP because they are generated from two different distributions. As a result, for a small I frame, there are large B frames, which should not be the case. Moreover, the generated synthetic traces are burstier than the original traces shown in Figs. 4 and 5.

B. Using Extensive Statistical Modeling

The purpose of developing our synthetic trace generator is to evaluate the performance of IPTV over of a network, and from the discussion in Section IV-B, it is apparent that using basic statistical properties of a video trace is insufficient to create an accurate synthetic trace generation. So, an improved version of the generator is required to address these limitations. We introduce a variation of I frame size within each scene to avoid flat-tops. Another reason for this problem is the lack of proper correlation across scenes. In a real video, successive scenes may have some correlation. We try to establish better correlation across different scenes by using relative I frame sizes for each scene. We also introduce the relative sizes of B frames with respect to the I frame of the GoP to create B frame sizes to create better correlation within a GoP.

1) I Frame Variation: The number of I frames in a scene is too small to obtain statistically significant data on the variation of I frame size within each scene. However, the overall variation of I frames within each scene with respect to the first I frame of the scene gives representative distribution as shown in Fig. 9. We observe that most I frames in a scene are almost of the same size but a few of them may vary quite a lot. This variation closely follows a Normal distribution with $\sigma = 0.11$ and $\mu = 0$. So, we can vary the I frame sizes in a scene with respect to the first I frame of the scene.

2) Relative I Frame Sizes Across Different Scenes: We analyze the relative sizes of the first I frame in every scene from the real video trace. We observe that a Normal distribution with $\sigma = 0.23$ and $\mu = -0.05$ as shown in Fig. 10 follows the relative I frame sizes across different scenes. We consider the first I frame because the synthetic video trace generation is a sequential process and the variation of I frame sizes within a scene is modeled based on the first I frame. So, we use the relative sizes of the first I frame of each scene to generate the I frame sizes in synthetic video trace and establish a better correlation across different scenes.

3) Relative B Frame Sizes within a GoP: We analyze the relative B frame size within a GoP with respect to its I frame. We observe from Fig. 11 that a Lognormal distribution with $\sigma = 0.7833$ and $\mu = -1.5115$ follows the relative B frame size closely. This improves correlation of the I and B frame sizes within the GoP.

C. Synthetic Video Trace Generation Algorithm

In our video trace generation algorithm, we introduce a variation to the I frames shown within a scene in Fig. 9 to remove flat-tops. Moreover, we generate the relative B frame sizes for all B frames in a GoP using the Lognormal distribution in Fig. 11 with respect to the I frame of the GoP. Then, we use these relative B frame sizes to generate the B frame sizes. We also generate relative sizes of the first I frames of each scene (Fig. 10) and generate the I frame size. The improved algorithm is shown in Fig. 12.

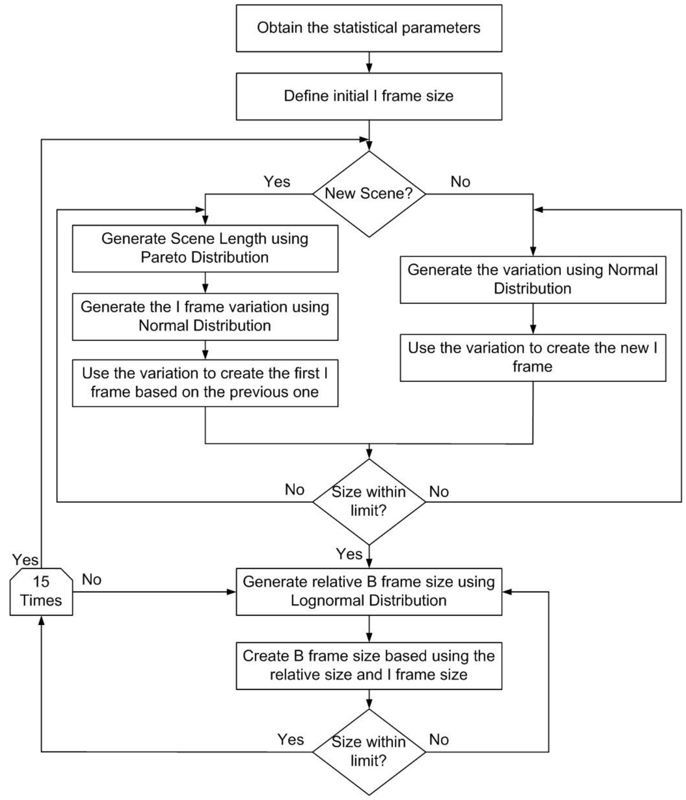


Fig. 12. Synthetic video trace generation algorithm.

V. Analysis of the Algorithm

We analyze how our algorithm follows the desired statistical properties. We observe from Figs. 13 and 14 that, even though we did not directly use the Lognormal distribution to generate the I or B frames in our algorithm in Fig. 12, the synthetic trace follows the targeted distribution closely. So, the frame sizes generated in this algorithm are quite accurate.

To analyze the effectiveness of our synthetic video trace generator, we convert the video frames of the synthetic trace from the algorithm in Fig. 12 into Ethernet frames representing IPTV streaming, as shown in Fig. 2. We observe the resultant data rate and compare it with the data rate of the actual video trace from [6]. From Fig. 15, we observe that the data rate of both the actual video trace and synthetic video trace are gradual. For example, we observe from both traces that, when the data rate is high, it remains in that range over a period of time, reflecting a scene.

Figure 16 shows I frame sizes of the synthetic trace over time that are similar to the actual trace shown in Fig. 4. We also observe from Figs. 15 and 4 that the video frames and the resultant data rate of the synthetic video trace closely follow the actual video. Note that observing the frame size distribution of B frames is not necessary in this case as they are generated from the relative sizes with respect to the I frame within a GoP.

The statistical parameters used to represent the video trace in [6] is not sufficient to generate synthetic video traces representative of all video types. However, by extracting the

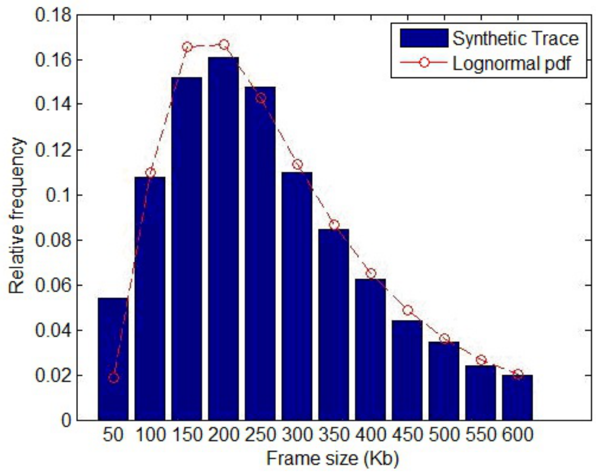


Fig. 13. I frame sizes of synthetic trace and targeted Lognormal distribution.

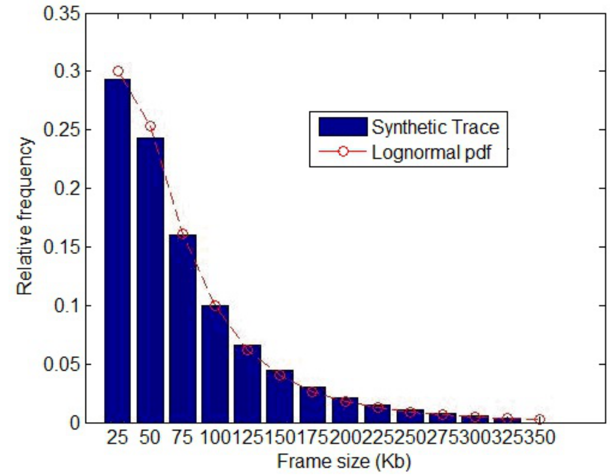


Fig. 14. . B frame sizes of synthetic trace and targeted Lognormal distribution.

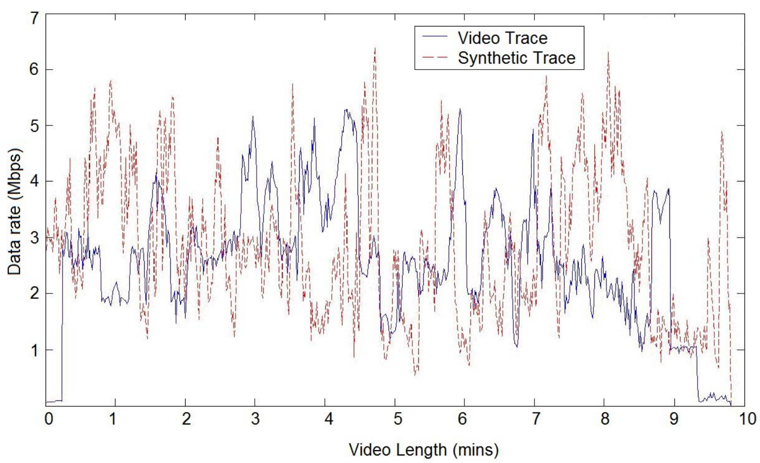


Fig. 15. Data rate over EPON for actual and synthetic video traces.

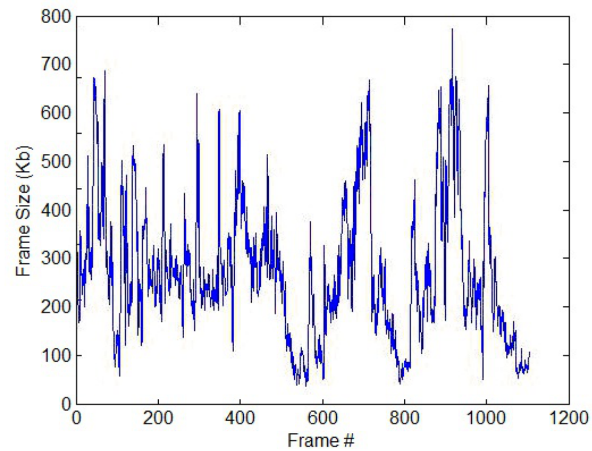


Fig. 16. I frame sequence of the synthetic trace.

parameters described in Section IV-B from different other video traces, the synthetic trace generator algorithm in Fig. 12 can generate synthetic traces representative of different video types.

VI. Conclusion

IPTV has become a popular streaming video delivery technology with the development of high-capacity access networks. It is essential to evaluate the performance of IPTV streaming over an access network before the deployment. To assist in such evaluation, in this paper, we explored different aspects of video streaming and designed a synthetic video trace generator that represents the IPTV streaming. Our analysis shows that our synthetic video trace has the desired statistical properties and the resultant data rate represents IPTV traffic over an access network.

Acknowledgment

We thank Professor Martin Reisslein of Arizona State University for providing us with the video traces and for his helpful suggestions.

References

[1] M. Mellia and M. Meo, "Communication networks measurement of IPTV traffic from an operative network," *European Transactions on Telecommunications*, 2009.

[2] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, "A measurement study of a large-scale P2P IPTV system," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1672–1687, Dec. 2007.

[3] P. Seeling and M. Reisslein, "Video transport evaluation with H.264 video traces," *IEEE Communications Surveys & Tutorials*, no. 99, Sep. 2011.

[4] L. Surhone (Ed.), M. Tennoe (Ed.), and S. Henssonow (Ed.), *Video Compression Picture Types*, Betascript Publishing, 2010.

[5] M. Reisslein, L. Karam, and P. Seeling, "H.264/AVC and SVC video trace library:a quick reference guide." http://trace.eas.asu.edu/publications/H264_QuRefMay09.pdf, May 2009.

[6] Video Traces for Network Performance Evaluation, "Video traces 2 for sony_1080." http://trace.eas.asu.edu/videotraces2/svc_single/1080/sony/sony1080_G16B15/, Sep. 2009.

[7] M. Dai, Y. Zhang, and D. Loguinov, "A unified traffic model for MPEG-4 and H.264 video traces," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 1010–1023, Aug. 2009.

[8] F. Wan, L. Cai, and T. Gulliver, "A simple, two-level markovian traffic model for IPTV video sources," in *Proc. IEEE Globecom*, New Orleans, LA, pp. 1474–1478, Nov.-Dec. 2008.

[9] M. Krunz and H. Hughes, "A traffic model for MPEG-coded VBR streams," in *Proc. ACM SIGMETRICS*, Ottawa, Canada, pp. 47–55, May 1995.