Video Streaming Over Wireless Network by CABAC Arithmetic Coding with Generalized FNT Transform

Ayad A. Abdulsalam

Computer Department, College of Education for Women, University of Baghdad, Baghdad, Iraq

Abstract: ***Multimedia Streaming is a very interesting addition to enrich media transmission experience. The recent trend in Internet traffic enhancement is a multimedia streaming services over wireless networks. In this paper we presented the combination of video compression method Context-based Adaptive Binary Arithmetic Coding CABAC with the video filtering scheme Generalized Fermat Number Transform GFNT which depends on advance mathematic to enrich the encourage throughput, our results show that the proposed scheme improves the streaming quality by keeping high PSNR values comparing with the traditional schemes.***

Keywords: Multimedia, Video Streaming, CABAC, FNT, Peack Signal to Noise Ratio, transmission protocol

1. Introduction

Multimedia streaming technology refers to transmitting data such as audio, video and other media over network connection directly from the source in real-time [1]. In streaming, the end user does not need to wait for video download to finish, "streaming" video will start after few seconds upon receiving the video frames. Technically, the video frames are still "downloading" but the end user does not need to wait before starting to watch. Media streaming is a continuous process, without any intermediate storage. While webcasting refers to both streaming and file downloading. Streaming might be considered a subset of web-casting. But streaming does not require to use the Web; data streams can be delivered through wired, wireless networks or over private intranets [1].

2. Data Streaming Methods

Media streaming can be used in two different methods: live streaming and on-demand streaming. Live streaming is when the transmission of the media is pushed to the client viewer as the content is created [2]. While the alternative is the on-demand, where the viewer requests the media from a content server (library), see Figure 1 [1]. Live media transmission especially video transmission is a real time transmission of video frames through a local area network or through the Internet so that it will be seen on personal computers, smart phones and mobile devices. The Live stream is called "real time" however the frames are filtered, compressed and then transmitted to the client where decoding process is performed, therefore some delay is introduced [2].

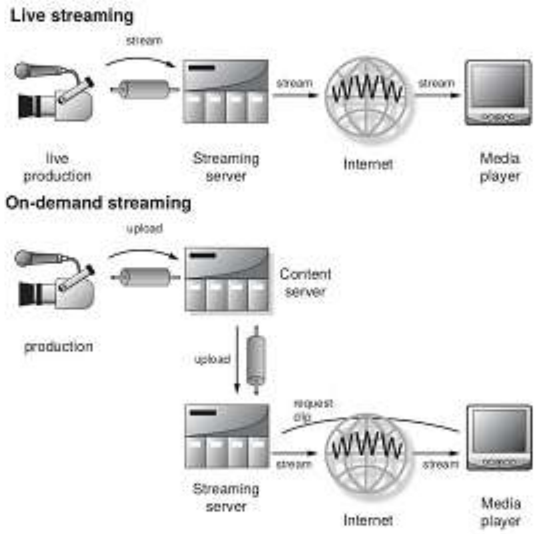

Figure 1: Live and On-demand Streaming

Many applications such as TV broadcasting and home entertainment in Video Home System VHS have been developed during the last decades. While those systems have served for many years, the end users still feel that there are limitations in scene reconstruction: scene is on a two dimension plane, video systems can offer immersive viewing experience, sense of presentation realism, and higher interaction [3] Delivering a smooth and steady media is not an easy task, but it is very hard to communicate when the stream is breaking down. There are a lot of issues and challenges that can go wrong with a live stream, and getting the end users to understand this obstacle is difficult [4]. Data streaming requires some combination of bandwidth sufficiency, for real-time human perception of the data, and the ability to make sure that enough data is being continuously received without any noticeable flicker.

Streaming data requires sending content in compressed form over the network to be displayed by the viewer in real time. With streaming, the client side can start displaying the data before the entire file has been transmitted. Therefore, the client side needs to receive the data more quickly than required to process, and convert it to visual or audible data. If the data is not received quickly enough, the presentation of the data will not be smooth.

Streaming video systems sometimes requires some kind of security especially when the streaming application is a highly confidential data, while on the other hand, sometimes the client only needs to verify that the streaming server is authentic. Therefore, some applications require securing the data stream itself especially when the video content is critical, while on the other hand, authentication can be sufficient. Authentication is based on certificates and Public Key Infrastructure (PKI). In summary, live and dynamic streaming technology with multimedia files are going on and will be going on for many years, since the development of new multimedia devices, protocols and network technologies are continuously and extensively [5].

Live video streaming such as video conferencing have a certain delay (encoding, transmission and decoding process) where the video is not actually being watched live, instead it is being displayed with a 15 to 90 seconds of delay [2]. Using different protocols will lead to different live streaming latencies; for Real Time Messaging Protocol RTMP the live streaming can vary from 4 to 9 seconds, however for HTTP live video streaming can vary from 35 seconds to 1 minute [6]. In live video streaming applications "latency" is the time between capturing the frame at the server to the time of displaying the frame at the client. The amount of total delay which is also called "End-to-End Latency" is the summation of: processing time, the time required for the transmission and rendering time. Low latency is a vital requirement in live streaming applications and can be achieved using different methods [2].

Dynamic streaming is required since the streaming depends on the available bandwidth between the client and the server. Additionally the system is designed to stream a live video which is a huge amount of data. Therefore, the system should have an adaptive feature, where the live video size will be adjusted according to the available network bandwidth.

3. Related Work

Cloud streaming service [7] focus on the comprehensive analysis of various cloud based streaming methods with underlying architecture and compare the performance of cloud based streaming with conventional streaming model. The experiment results shows that cloud based streaming delivery outperforms in terms of throughput, packet delivery and effective use of the network bandwidth under similar network conditions. Authors stated a comparisons with other methods (RTP, HTTP, HLS, DASH), we depend on the results that they reached as a reference to evaluate our proposed scheme.

4. Proposed Scheme

Presented system focuses on a video streaming over a WLAN, the streaming action will start only after the server certificate is being authenticated by the client. The authentication process is done through Transport Layer Security TLS handshake protocol. Certificate generation is done by creating a private/public key pair for the server and then a certificate request signing is created to be issued and signed by a Certification Authority CA. In this paper, a self-signed certificate is also generated in case of running the system at different network. If the client checks that the server certificate is authentic, the streaming phase will begin.

The server measures the existed network bandwidth with the client. The generated video is represented where filtering and compression is a most accomplished stage before streaming.

A. Video Compression Scheme

Context-based Adaptive Binary Arithmetic Coding (CABAC) as a normative part of the new standard for video compression is applied. By combining an adaptive binary arithmetic coding technique with context modeling, a high degree of adaptation and redundancy reduction is achieved. The CABAC framework also includes a novel low-complexity method for binary arithmetic coding and probability estimation that is well suited for efficient hardware and software implementations. CABAC significantly outperforms the baseline entropy coding method of H.264/AVC for the typical area of envisaged target applications. For a set of test sequences representing typical material used in broadcast applications and for a range of acceptable video quality of about 30 to 38 dB, average bit-rate savings of 9%-14% are achieved. [8].

B. Video Filtering Schemes

Video convolution is investigated based on Fermat Number Transform (FNT) modulo q=2M+1 where M is an integer power of two. These transforms are found to be ideal for image convolutions, except that the choices for the word length, restricted by the transform modulus, are rather limited. Two methods are discussed to overcome this limitation. First, allow M to be an arbitrary integer. This gives much wider variety in possible moduli, at the cost of decreased transform length of 16 or 32 points for M<32. Nevertheless, the transform length appears still to be useful especially with block-based video filtering applications. These transforms are called the generalized FNT (GFNT). The second solution is to use a Residue Number System (RNS) to enlarge the effective modulus, while performing actual number theoretic transforms with smaller moduli. This approach appears to be particularly useful with moduli q1=216+1 and q2=28+1, which allow transforms up to 256 points with a dynamic range of about 24 bits. An efficient reconstruction method is designed based on mixed radix conversion for converting the result from diminished-1 RNS into normal binary code [9].

5. Experimental Setup

In our experiments we used the following dataset for testing: bit rates from 100Kbps to 6000 Kbps with the quality level ranging from 265p up to 1024P resolution with 16 different levels .To maintain the testing consistency, each time the video streamed from the server in internet environment and streamed over the duration of 300 seconds. There are enough debug messages added in receiving streaming player components to capture the session logs along with wire shark packet analysis.

The streaming evaluation has been carried out network simulation with different internet bandwidth configuration using dummy net tool. Dummy net is a network emulation tool, which serves to access and limit link bottleneck. Internet experiment use the same segmented video files for testing. The bandwidth is configured from 4 Mbps in order to study the system behavior under the same network availability. Segment duration 10 seconds for HLS and 4 seconds for all other streaming methods are taken for experiment. HLS allows only fixed 10 seconds duration segments. The experiment is aimed to measure and compare the following metrics useful to analyze the performance of the rate adaptation.

1) Average throughput: The average received output of the streaming media is calculated for the duration of 300 seconds. Average throughput is one of the metric to measure the effectiveness of the stream switching algorithms for the available bandwidth.
2) Average PSNR: Average value of peak signal to noise ratio of the received video. PSNR gives the efficiency of the received throughput versus the target bit rate and it defines the quality of the stream.
3) Average Delta delay: It is the average delay between each received consecutive packet.
4) Average Bandwidth utilization: The bandwidth utilization measures the actual use of available link capacity. It is important performance metric of the client rate adaptation.
5) Packet loss: The number of packets arrives later than play out latency due to network/retransmission delay or retransmitted due to loss or completely lost in transport.
6) Packet loss/retransmitted count: The number of packets lost during transmission or packets discarded due to late arrival.

6. Results and Discussion

The key streaming performance metrics are measured in the experiment for the duration of 300 Seconds and the same are listed in Table 1.

Table 1: Stremaing Performance Metric Comparison

Method	Throughput KB/s	PSNR dB	Delay ms	Band utilization %	Lost packets %
HLS	225.86	18.87	5.133	46.08	0.17
DASH	164.209	19.76	4.67	33.63	0.99
Cloud Streaming	342.861	22.19	4.37	70.03	0.58
CABAC/GFNT	325.717	31.23	3.91	67.81	0.22

Table1 shows that the CABAC/GFNT adaptive streaming outperforms in terms of PSNR, and bandwidth utilization. Since the cloud streaming evaluation uses amazon streaming environment where the amazon cloud front is located near to the end user location which enable fast data transfer and efficiently uses the bandwidth by transferring good quality data. CABAC/GFNT streaming performs average and efficient than traditional streaming .Though adaptive streaming uses same bandwidth and transfer various quality data based on the user need so that the end user experience is exceptional compare to the conventional streaming.

The number of packets arrives later than play out latency in due to network/retransmission delay or retransmitted due to loss or completely lost in transport impacts the application quality. Considerable amount of packets are retransmitted/ lost in MPEG DASH and cloud adaptive streaming compare to conventional streaming, while it enhanced in CABAC/GFNT streaming method. This occurs due to more network congestion and incompetent rate selection for some duration. Figure 2 represents the captured throughput for HLS as a sample of throughput of other services.

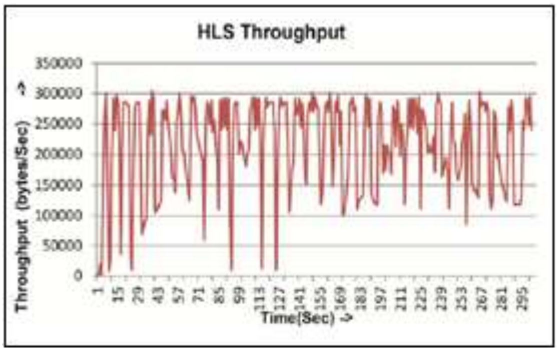

Figure 2: HLS throughput

7. Future Work and Conclusion

In this paper, we have discussed evolution of CABAC/GFNT service delivery with underlying technology and. The performance metrics of streaming methods and efficiency are measured, which gives the behavior understanding of the streaming methods in the bottleneck environment and provides more research possibilities in the area of improvement required thereby paving way for further research and enhancement. The outcome of the performance result shows that CABAC/GFNT based delivery excels in efficient use of resources and overall quality of experience. It also helps to understand issues and challenges of the current implementations of the streaming models. Extension to this work will involve analyze the different filters and compression methods under different network conditions and propose solution to address all the performance issues.

References

[1] D. Austerberry. “The Technology of Video and Audio Streaming”. Elsevier, 2005.
[2] TKO Video Conferencing. Live Video Streaming. English. 2015. url: http://www.video-conferencing.com/definition/live-video-streaming.html.
[3] G. Petrovic. “Efficient 3D Video Streaming”. PhD thesis. Eindhoven, University of Technology, 2013
[4] T. Dreier. Livestream Explains the Challenges of Streaming Live Video. 2013. url: http://www.streamingmediaglobal. Com /Articles /Editorial /Featured -Articles/Livestream- Explains - the - Challenges - of - StreamingLive - Video - 88010.aspx
[5] Sultani, Zainab. “Dynamic and Secure Live Data Streaming Protocol”. Phd theises. University of Technology, Baghdad, 2016.
[6] Streaming Media UX Specialist Supreet Singh and Founder DUX Media. What is a typical live video streaming latency distribution? - Quora. English. 2014.

url: https : / / www . quora.com/What-is-a-typical-live-video-streaminglatency- distribution.

[7] Selvaraj, Kesavan. "Network Performance Analysis of Cloud Based Multimedia Streaming Service" IEEE Transactions on Circuits and Systems for Video Technology (Volume:16 , Issue: 1).

[8] D. Marpe. " Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard". IEEE Transactions on Circuits and Systems for Video Technology (Volume:13 , Issue: 7), July 2003

[9] T. Toivonen. " Video filtering with Fermat number theoretic transforms using residue number system". IEEE Transactions on Circuits and Systems for Video Technology (Volume:16 , Issue: 1), Jan. 2006.