JOURNAL OF ADVANCEMENT IN ENGINEERING AND TECHNOLOGY
Journal homepage: www.scienceq.org/JAET.php
ISSN: 2348-2931
V4I4.04
Research Article Open Access

An Efficient Streaming Service Adaptation On Mobile Devices

K. Sivarama gandhi[1*], I. Ravi[2]
[1,2]Department of computer Science, K.R.College of Arts and Science, Kovilpatti. India
*Corresponding author: E-mail: gandhisivarama@gmail.com

Received: February 1, 2017, Accepted: March 5, 2017, Published: March 5, 2017.

ABSTRACT

The Internet mobile streaming service at the server side particularly for the recent Internet mobile streaming delivery is investigated. H.264 is an industry standard for video compression, the process of converting digital video into a format that takes up less capacity when it is stored or transmitted. The H.264 standard provides advanced algorithms for motion estimation, inter-prediction, spatial intra prediction and transforms. The H.264 standard is detecting a user's bandwidth capability in real time and then adjusting the quality of the video stream accordingly. This results in less buffering, fast start times and overall better experience for both high-speed and low-speed connections. Perhaps the biggest advantage of H.264 over previous standards is its compression performance. Compared with standards such as MPEG-2 and MPEG-4 Visual, H.264 can deliver, Better image quality at the same compressed bit rate, or A lower compressed bitrates for the same image quality.

Keywords: *Streaming, codec, Compression, Buffer management.*

INTRODUCTION

Streaming or media streaming is a technique for transferring data so that it can be processed as a steady and continuous stream. Streaming technologies are becoming increasingly important with the growth of the Internet because most users do not have fast enough access to download large multimedia files quickly. With streaming, the client browser or plug-in can start displaying the data before the entire file has been transmitted.

For streaming to work, the client side receiving the data must be able to collect the data and send it as a steady stream to the application that is processing the data and converting it to sound or pictures. This means that if the streaming client receives the data more quickly than required, it needs to save the excess data in a buffer.

If the data doesn't come quickly enough, however, the presentation of the data will not be smooth. There are a number of competing streaming technologies emerging. For audio data on the Internet, the de facto standard is Progressive Network's RealAudio.

A client media player can begin playing the data (such as a movie) before the entire file has been transmitted. Distinguishing delivery method from the media distributed applies specifically to telecommunications networks, as most of the delivery systems are either inherently streaming (e.g., radio, television) or inherently nonstreaming (e.g., books, video cassettes, audio CDs). For example, in the 1930s, elevator music was among the earliest popularly available streaming media; nowadays Internet television is a common form of streamed media. The term "streaming media" can apply to media other than video and audio such as live closed captioning, ticker tape, and real-time text, which are all considered "streaming text". The term "streaming" was first used in the early 1990s as a better description for video on demand on IP networks; at the time such video was usually referred to as "store and forward video", which was misleading nomenclature.

Live streaming, which refers to content delivered live over the Internet, requires a form of source media (e.g. a video camera, an audio interface, screen capture software), an encoder to digitize the content, a media publisher, and a content delivery network to distribute and deliver the content.

An adaptive buffer power save mechanism for increasing the battery life of mobile devices during multimedia streaming. this will control how and when data will sent over a wireless lan. AB-PSM introduces additional buffer which hides data from the station it is intended for, allowing it to return to sleep and consequently saving power[1]. A simple model where the web accesses are independent and the reference probability of the documents follows zipflike distribution in [2]. System find that model yields asymptotic behavior that are consistent with the experimental observations, suggesting that various observed properties of hit ratios and temporal locality are indeed inherent to web accesses observed by proxies. The experimental results shows that network transmission using WCDMA consumes more energy than when using WLAN in [3]. In addition the power consumption is relevant with status network interfaces and impact on energy consumption based on the storge media used can be ignored. Convenience

The characterize of the performance of network applications on smartphones in a way that is relevant to end uses,cellular operator,smartphone vendors,applications developers as well as content providers in [4].

3 . Streaming and Compression

Streaming or media streaming is a technique for transferring data so that it can be processed as a steady and continuous stream. Streaming technologies are becoming increasingly important with the growth of the Internet because most users do not have fast enough access to download large multimedia files quickly. With streaming, the client browser or plug-in can start displaying the data before the entire file has been transmitted.

For streaming to work, the client side receiving the data must be able to collect the data and send it as a steady stream to the application that is processing the data and converting it to sound or pictures. This means that if the streaming client receives the data more quickly than required, it needs to save the excess data in a buffer. If the data doesn't come quickly enough, however, the presentation of the data will not be smooth.

Compression is useful because it reduces resources required to store and transmit data. Computational resources are consumed in the compression process and, usually, in the reversal of the process (decompression). Lossless data compression algorithms usually exploit statistical redundancy to represent data without losing any information, so that the process is reversible. Lossless compression is possible because most real-world data exhibits statistical redundancy. For example, an image may have areas of color that do not change over several pixels; instead of coding "red pixel, red pixel, ..." the data may be encoded as "279 red pixels". This is a basic example of run-length encoding; there are many schemes to reduce file size by eliminating redundancy. Lossy data compression is the converse of lossless data compression. In these schemes, some loss of information is acceptable. Dropping nonessential detail from the data source can save storage space. Lossy data compression schemes are designed by research on how people perceive the data in question. For example, the human eye is more sensitive to subtle variations in luminance than it is to the variations in color.

3.1 STREAMING METHODOLOGY

Data stream: a sequence A = <a1, a2,..., am>, where the elements of the sequence (called tokens) are drawn from the universe [n]□ = {1, 2, ..., n}

Aim : compute a function over the stream, eg: median, number of distinct elements, longest increasing sequence, etc.

Target Space complexity

- ❖ Since m and n are "huge," so want to make s (bits of random access memory) much smaller than these
- ❖ Specifically, s to be sublinear in both m and n.

$$s = o(\min\{m, n\})$$

- The best would be to achieve

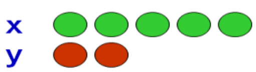
$$s = O(\log m + \log n)$$

Algorithm : Cash-Register Model:

Arrivals-Only Streams

c[x] is always > 0 Typically, c[x]=1

Example: <x, 3>, <y, 2>, <x, 2> encodes the arrival of



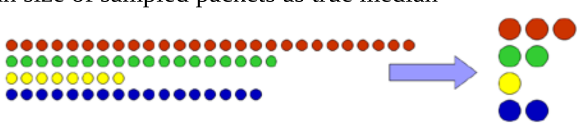
3 copies of item x,

2 copies of y,

copies of x.

Could represent, packets in a network, power usage

To compute median packet size, Sample some packets, Present median size of sampled packets as true median



Probability of any element to be included at round t

- ❖ Let us consider a time t > N.
- ❖ Let the number of elements that has arrived till now be Nt
- ❖ Since at each round, all the elements have equal probabilities, the probability of any element being included in the sample is N/ Nt
- ❖ For the last arriving element to be selected, the probability is N/ NT
- ❖ For the element before the last, the probability of selection = N/ (NT -1)
- ❖ The probability of the last element replacing the last but one element = (N/ NT) X (1/N) = 1/ NT
- ❖ The probability that the last but one element survives = 1-1/ NT = (NT -1)/ NT
- ❖ The probability that the last but one survives till the end = (N/(NT -1)) X (NT -1)/ NT = N/ NT

Memory Usage for Priority Sampling

- ❖ The Proposed Model only the eligible elements in the memory will be stored
- ❖ These elements can be made to form right spine of the data structure "treap"
- ❖ Therefore expected memory usage is O(log n), or O(k log n) for samples of size k

3.2 COMPRESSION METHODOLOGY

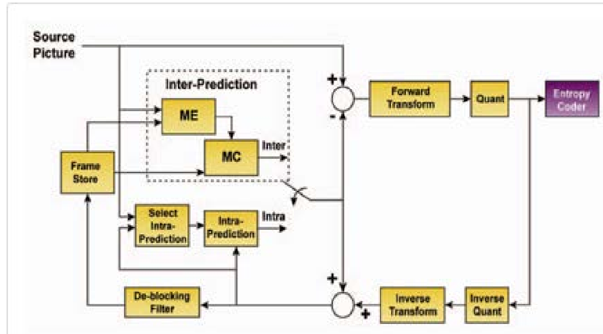


Figure 3.2.1 architecture of an H.264 encoder.

H.264 Algorithm:

1. Intra-prediction utilizing Forward and
2. Inverse Discrete Cosine Transforms (DCT), as well as Forward and Inverse Quantization
3. De-blocking filtering
4. Motion Estimation employing interframe comparisons to _ PEL and _ PEL accuracy.

As shown in the figure 3.2.1, each of these functions requires extensive processing and must be performed in real-time to be useful. The remainder of this paper will focus on how software-configurable processors provide a development methodology for creating cost-effective H.264 implementations that enable experimentation and refinement of algorithms.

3.3 How video compression works

Video compression is about reducing and removing redundant video data so that a digital video file can be effectively sent and stored. The process involves applying an algorithm to the source video to create a compressed file that is ready for transmission or storage. To play the compressed file, an inverse algorithm is applied to produce a video that shows virtually the same content as the original source video. The time it takes to compress, send, decompress and display a file is called latency. The more advanced the compression algorithm, the higher the latency, given the same processing power. A pair of algorithms that works together is called a video codec (encoder/decoder). Video codecs that implement different standards are normally not compatible with each other; that is, video content that is compressed using one standard cannot be decompressed with a different standard. For instance, an MPEG-4 Part 2 decoder will not work with an H.264 encoder. This is simply because one algorithm cannot correctly decode the output from another algorithm but it is possible to implement many different algorithms in the same software or hardware, which would then enable multiple formats to be compressed. Different video compression standards utilize different methods of reducing data, and hence, results differ in bit rate, quality and latency.

Results from encoders that use the same compression standard may also vary because the designer of an encoder can choose to implement different sets of tools defined by a standard. As long as the output of an encoder conforms to a standard's format and decoder, it is possible to make different implementations. This is advantageous because different implementations have different goals and budget. Professional non-real-time software encoders for mastering optical media should have the option of being able to deliver better encoded video than a real-time hardware encoder for video conferencing that is integrated in a hand-held device. A given standard, therefore, cannot guarantee a given bit rate or quality.

Furthermore, the performance of a standard cannot be properly compared with other standards, or even other implementations of the same standard, without first defining how it is implemented. A decoder, unlike an encoder, must implement all the required parts of a standard in order to decode a compliant bit stream. This is because a standard specifies exactly how a decompression algorithm should restore every bit of a compressed video.

3.4 H.264 profiles and levels

The joint group involved in defining H.264 focused on creating a simple and clean solution, limiting options and features to a minimum. An important aspect of the standard, as with other video standards, is providing the capabilities in profiles (sets of algorithmic features) and levels (performance classes)that optimally support popular productions and common formats.

H.264 has seven profiles, each targeting a specific class of applications. Each profile defines what feature set the encoder may use and limits the decoder implementation complexity.

Network cameras and video encoders will most likely use a profile called the baseline profile, which isintended primarily for applications with limited computing resources. The baseline profile is the most suitable given the available performance in a real-time encoder that is embedded in a network video product. The profile also enables low latency, which is an important requirement of surveillance video and also particularly important in enabling real-time, pan/tilt/zoom (PTZ) control in PTZ network cameras.

H.264 has 11 levels or degree of capability to limit performance, bandwidth and memory requirements. Each level defines the bit rate and the encoding rate in macroblock per second for resolutions ranging from QCIF to HDTV and beyond. The higher the resolution, the higher the level required.

3.5 Frames

Depending on the H.264 profile, different types of frames such as I-frames, P-frames and B-frames, may be used by an encoder.An I-frame, or intra frame, is a self-contained frame that can be independently decoded without any reference to other images. The first image in a video sequence is always an I-frame. I-frames are needed as starting points for new viewers or resynchronization points if the transmitted bit stream is damaged. I-frames can be used to implement fast-forward, rewind and other random access functions. An encoder will automatically insert I-frames at regular intervals or on demand if new clients are expected to join in viewing a stream. The drawback of I-frames is that they consume much more bits, but on the other hand, they do not generate many artifacts.

A P-frame, which stands for predictive inter frame, makes references to parts of earlier I and/or P frame(s) to code the frame. P-frames usually require fewer bits than I-frames, but a drawback is that they are very sensitive to transmission errors because of the complex dependency on earlier P and I reference frames.A B-frame, or bi-predictive inter frame, is a frame that makes references to both an earlier reference frame and a future frame.

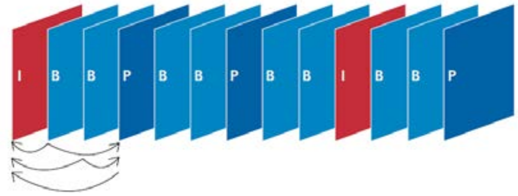


Figure 3.2.2 A typical sequence with I-, B- and P-frames. A P-frame may only reference preceding I- or P-frames, while a B-frame may reference both preceding and succeeding I- or P-frames.

When a video decoder restores a video by decoding the bit stream frame by frame, decoding must always start with an I-frame. P-frames and B-frames, if used, must be decoded together with the reference frame(s).In the H.264 baseline profile, only I- and P-frames are used. This profile is ideal for network cameras and video encoders since low latency is achieved because B-frames are not used.

3.6 Basic methods of reducing data

A variety of methods can be used to reduce video data, both within an image frame and between a series of frames. Within an image frame, data can be reduced simply by removing unnecessary information, which will have an impact on the image resolution.

In a series of frames, video data can be reduced by such methods as difference coding, which is used by most video compression standards including H.264. In difference coding, a frame is compared with a reference frame (i.e. earlier I- or P-frame) and only pixels that have changed with respect to the reference frame are coded. In this way, the number of pixel values that are coded and sent is reduced.

Figure 3.2.3 With Motion JPEG format, the three images in the above sequence are coded and sent as separate unique images (I-frames) with no dependencies on each other.

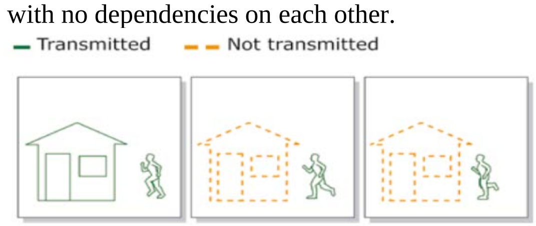


Figure 3.2.4 With difference coding (used in most video compression standards including H.264), only the first image (I-frame) is coded in its entirety. In the two following images (P-frames), references are made to the first picture for the static elements, i.e. the house, and only the moving parts, i.e. the running man, is coded using motion vectors, thus reducing the amount of information that is sent and stored.

The amount of encoding can be further reduced if detection and encoding of differences is based on blocks of pixels (macro blocks) rather than individual pixels; therefore, bigger areas are compared and only blocks that are significantly different are coded. The overhead associated with indicating the location of areas to be changed is also reduced. Difference coding, however, would not significantly reduce data if there was a lot of motion in a video. Here, techniques such as block-based motion compensation can be used. Block-based motion compensation takes into account that much of what makes up a new frame in a video sequence can be found in an earlier frame, but perhaps in a different location. This technique divides a frame into a series of macro blocks. Block by block, a new frame—for instance, a P-frame—can be composed or 'predicted' by looking for a matching block in a reference frame. If a match is found, the encoder simply codes the position where the matching block is to be found in the reference frame. Coding the motion vector, as it is called, takes up fewer bits than if the actual content of a block were to be coded.

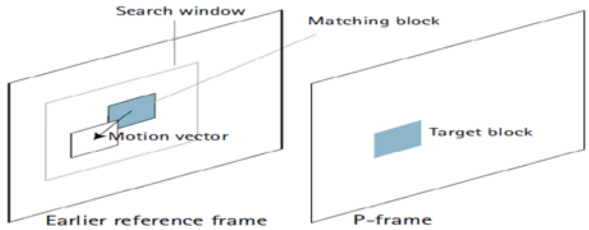


Figure 3.2.4 Illustration of block-based motion compensation

4. EXPERIMENTATION RESULT

H.264 takes video compression technology to a new level. With H.264, a new and advanced intra prediction scheme is introduced for encoding I-frames. This scheme can greatly reduce the bit size of an I-frame and maintain a high quality by enabling the successive prediction of smaller blocks of pixels within each macroblock in a frame. This is done by trying to find matching pixels among the earlier encoded pixels that border a new 4x4 pixel block to be intra-coded. By reusing pixel values that have already been encoded, the bit size can be drastically reduced. The new intra prediction is a key part of the H.264 technology that has proven to be very efficient. For comparison, if only I-frames were used in an H.264 stream, it would have a much smaller file size than a Motion JPEG stream, which uses only I-frames.

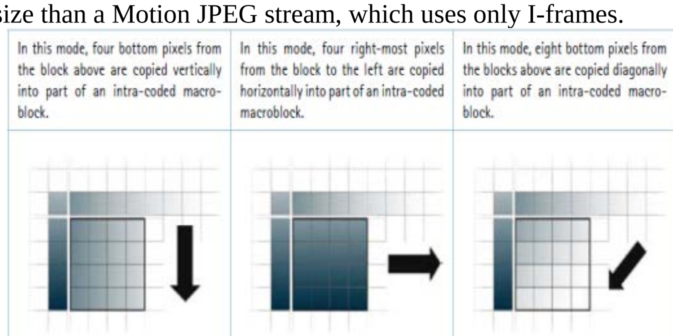


Figure 4.1.1 Illustrations of some of the modes that intra prediction can take in coding 4x4 pixels within one of the 16 blocks that make up amacroblock. Each of the 16 blocks in a macroblock may be coded using different modes.

The graph below provides a bit rate comparison, given the same level of image quality, among the following video standards: Motion JPEG, MPEG-4 Part 2 (no motion compensation), MPEG-4 Part 2 (with motion compensation) and H.264 (baseline profile).

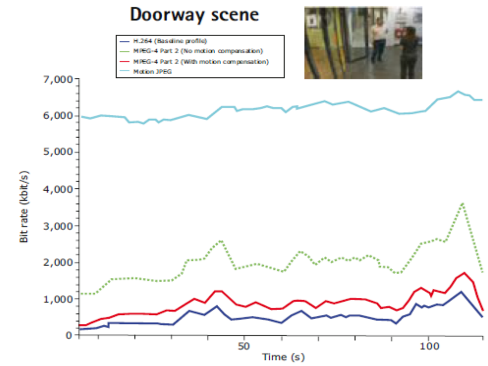


Figure 4.1.2 An H.264 encoder generated up to 50% fewer bits per second for a sample video sequence than an MPEG-4 encoder with motion compensation. The H.264 encoder was at least three times more efficient than an MPEG-4 encoder with no motion compensation and at least six times more efficient than Motion JPEG.

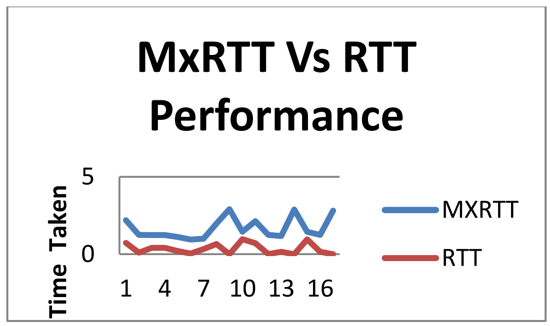


Figure 4.1.3 the MaxRTT vs RTT performance

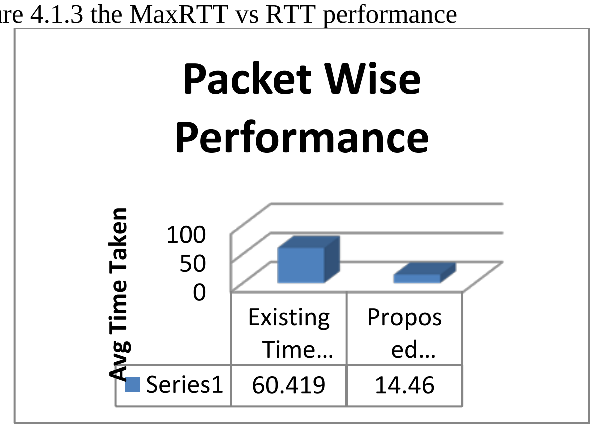


Figure 4.1.4 the packet wise performance

5. CONCLUSION AND FUTURE ENHANCEMENT

The H.264 standard provides advanced algorithms for motion estimation, inter-prediction, spatial intra prediction and transforms. The H.264 standard to detecting user's bandwidth capabilities in real time and then adjusting the quality of the video stream accordingly.

- ❖ Encoding and decoding is more complex than some other codec's (mpeg4...)
- ❖ Compression performance, the difference between mpeg 4 part 2 and H264 is really not high
- ❖ H264 uses lossy compression like Jpeg, during the encoding preocess during the encoding processes images are blurred
- ❖ latency is higher(time delay)
- ❖ H264 takes more processing power than earlier codecs
- ❖ Error resilliency

The H.264 has now been widely accepted and the focus is move to H.265. Streaming is very important in data transfer from one node to another node through internet. Lot of compression techniques are available in the market. But all are used in different type of application. The future enhancement of this paper is to make one efficient compression technique for all type of usage.

REFERENCES

1. G. Georgoulas, D. Stylios, P. Groumpos, Predicting the risk of metabolic acidosis for new borns based on fetal heart rate signal classification using support vector machines, IEEE Trans. Biomed. Eng. 53 (2006) 875–884.
2. S.L. Salzberg, On comparing classifiers: pitfalls to avoid and a recommended approach, Data Min. Knowl. Discov. (2007) 317–328.
3. M. Cesarelli, M. Romano, P. Bifulco, Comparison of short term variability indexes in cardiotocographic fetal monitoring, Comput. Biol. Med. 39 (2009) 106–118.
4. K. Bache, M. Lichman, Cardiotocography data set, in: UCI Machine Learning Repository, 2010. http://archive.ics.uci.edu/ml/datasets/cardiotocography
5. N. Krupa, M. Ali, E. Zahedi, S. Ahmed, F.M. Hassan, Antepartum fetal heart rate feature extraction and classification using empirical mode decomposition and support vector machine, Biomed. Eng. Online 10 (2011).

Citation: K. Sivarama gandhi *et al.* (2017). An Efficient Streaming Service Adaptation On Mobile Devices, J. of Advancement in Engineering and Technology, V4I4.04. DOI: 10.5281/zenodo.998907.