Video Streaming in iOS using CSTREAMER

A.Tamizh Suwadhiga[1] R.Aktharunisa Begum[2]
[1]PG Student [2]Assistant Professor
[1,2]Department of Computer Applications
[1,2]IFET College of Engineering (Anna University)

Abstract— Transmission of period video generally has information measure, delay, and loss necessities. However, the present best effort net doesn't supply any quality of service (QoS) guarantees to streaming video. To introduce this special issue with the required background and supply Associate in Nursing integral read on this field have a tendency to cowl six key areas of streaming video especially, for the favored iOS primarily based mobile devices, accessing widespread net streaming services generally involves concerning 100% seventieth redundant traffic. Such a follow not solely over utilizes and wastes resources on the server facet and therefore the network (cellular or Internet), however additionally consumes extra battery power on users mobile devices and results in potential financial value. To alleviate such a state of affairs while not dynamical the server facet or the iOS have a tendency to style and implement a CStreamer image which will transparently work between existing iOS devices and media servers. It have a tendency to additionally build a CStreamer iOS App to alter finish users to access net streaming services via CStreamer.

Key words: CStreamer Image, Video Streaming

I. INTRODUCTION

Real-time transport of live video or hold on video is that the predominant a part of period of time transmission during this paper tend to square measure involved with video streaming, that refers to period of time transmission of hold on video. There square measure 2 modes for transmission of hold on video over the web, specifically the transfer mode and therefore the streaming mode (i.e., video streaming) within the transfer mode, a user downloads the complete video file and so plays back the video file. However, full file transfer in the transfer mode typically suffers long and maybe unacceptable transfer time. In contrast, within the streaming mode, the video content needn't be downloaded fully, but is being compete out whereas components of the content square measure being received and decoded. Compared to general internet water sport on the web, streaming applications usually involve bulk knowledge transmission in an exceedingly continuous fashion, which can spend the restricted battery power offer at a quick pace and incur further financial value for cellular knowledge arrange users. Therefore for mobile devices and mobile users, it's important that the streaming knowledge ought to be delivered in an exceedingly precise fashion while not excess traffic or further financial value. However, it discover that for the favored iOS primarily based mobile devices, accessing streaming services usually involves concerning 100% to seventieth excess redundant traffic if a user watches the requested video from the start to the top. That is, such redundant traffic isn't attributable to the first termination of the shopper access. Through experiments and analysis tend to more investigate why such a big quantity of redundant traffic is transmitted. Results enhance user's expertise of doubtless re-watching the video, the iOS Media Player perpetually re-down hundreds the start a part of the video once more when finishing downloading the complete file; Once the downloading speed is quick, the Media Player oftentimes aborts the hypertext transfer protocol association and so sends the request once more, inflicting knowledge in flow to be wasted; Once the downloading speed is slow, the Media Player unceasingly sends extra and overlapping requests to sleek the playback. Such a big quantity of redundant traffic not solely wastes network information measure, however conjointly over-utilizes server-side resources.A streaming server is commonly wanting information measure and process power nowadays attributable to the speedy increase of video files and requests. Moreover, notwithstanding such redundant traffic is for the sake of user's perceived streaming performance, it's harmful to the mobile device's interest (in terms of battery power consumption) and therefore the mobile user's interest (in terms of potential further financial cost) impelled by our measuring results, tend to examine the potential causes for such excess traffic in traditional mobile streaming accesses. we discover these issues square measure chiefly attributable to the restricted memory size of fashionable iOS devices and therefore the too fast/slow network connections. These findings inspire US to hunt effective solutions to alleviate and minimize such redundant traffic while not modifying the server facet or the shopper facet. For this purpose tend to style and implement CStreamer that may transparently work between the shopper and therefore the server. CStreamer partitions the video content into little segments. To eliminate the re-downloaded traffic, CStreamer synchronizes the downloading with the Media Player's playback progress. To refrain from causation too quick, it serves the segments sporadically, rather than all quickly. To traumatize slow connections, CStreamer permits the Media Player to seamlessly switch to a lower quality version of identical video provided by the server. to gauge the effectiveness of CStreamer in minimizing the redundant traffic, got enforced a example of CStreamer running on Amazon EC2. To modify clear user accesses, got conjointly engineered a corresponding CStreamer Application. Completely different iOS devices square measure taught to access varied streaming services via this example. Our experimental results show that CStreamer will utterly eliminate the redundant traffic while not poignant user perceived streaming expertise.

II. ARCHITECTURE FOR VIDEO STREAMING

Video streaming usually has information measure, delay and loss needs. However, the present best-effort net doesn't supply any quality of service (QoS) guarantees to streaming video over the web additionally, for multicast, it's troublesome to expeditiously support multicast video whereas providing service flexibility to fulfill a good vary of QoS needs from the users. Thus, planning mechanisms and protocols for net streaming video poses several challenges to

handle these challenges, intensive analysis has been conducted. This special issue is geared toward dissemination of the contributions within the field of streaming video over the web. To introduce this special issue with the mandatory background and provides the reader a whole image of this field, tend to cowl six key areas of streaming videos are video compression, application-layer QoS management, continuous media distribution services, streaming servers, media synchronization and protocols for streaming media every of the six areas could be a basic building block, with that associate design for streaming video will be engineered. The relations among the six basic building blocks will be illustrated in Fig. 1 shows associate design for video streaming. In that, raw video and audio knowledge square measure pre-compressed by video compression and audio compression algorithms and so saved in storage devices. Upon the client's request, a streaming server retrieves compressed video/audio knowledge from storage devices and so the application-layer QoS management module adapts the video/audio bit-streams consistent with the network standing and QoS needs. when the difference, the transport protocols packet size the compressed bit-streams and send the video/audio packets to the web. Packets is also born or expertise excessive delay within the web attributable to congestion to enhance the standard of video/audio transmission, continuous media distribution services (e.g., caching) square measure deployed within the net for packets that square measure with success delivered to the receiver, initial labor in the transport layers and square measure and processed by applying layer before decoding the video/audio decoder to realize synchronization between video and audio shows, media synchronization mechanisms square measure needed. From it will be seen that the six square measure as square measure closely connected and that they are coherent constituents of the video streaming design tend to concisely describe the six areas as follows.

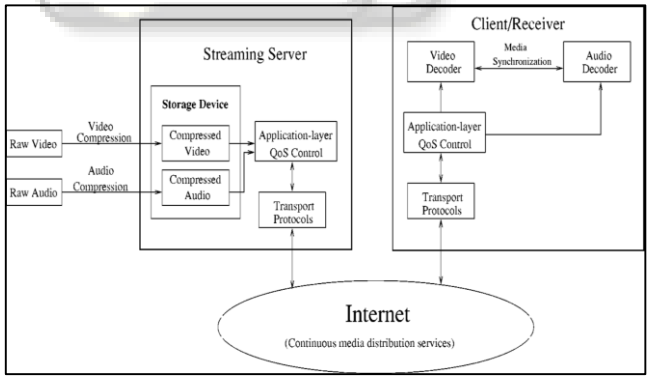

Fig. 1: Architecture for video Streaming

A. Video Compression:

Raw video should be compressed before transmission to attain potency. Video compression schemes are classified into 2 categories are scalable and non-scalable video secret writing. Since scalable video is capable of graciously handling the information measure fluctuations within the net tend to square measure primarily involved with scalable video secret writing techniques. will discuss the necessities obligatory by streaming applications on the video encoder and decoder.

B. Application-Layer QoS Control:

To address variable network conditions and totally different presentation quality requested by the users, varied application-layer QoS management techniques are planned . The application-layer techniques embrace congestion management and error management. Their several functions square measure as follows. Congestion management is utilized to forestall packet loss and cut back delay. Error management, on the opposite hand, is to enhance video presentation quality within the presence of packet loss. Error management mechanisms embrace forward error correction (FEC), re-transmission, error-resilient secret writing, and error concealment.

C. Continuous Media Distribution Services:

To supply quality transmission shows, adequate network support is crucial will be as a result of network support can cut back transport delay and packet ratio. designed on prime of the net (IP protocol), continuous media distribution services square measure ready to attain QoS and potency for streaming video/audio over the best-effort net. Continuous media distribution services embrace network filtering, application-level multicast, and content replication.

D. Streaming Servers:

Streaming servers is the key role in providing streaming services. To supply quality streaming services, streaming servers square measure needed to method transmission information below temporal arrangement constraints and support interactive management operations like pause/ resume, quick forward, and quick backward. What is more, streaming servers have to be compelled to retrieve media elements during a synchronous fashion. A streaming server generally consists of 3 subsystems, namely, a mortal (e.g., transport protocols), An package, and a storage system.

E. Media Synchronization Mechanisms:

Media synchronization could be a major feature that distinguishes transmission applications from different ancient information applications. With media synchronization, the applying at the receiver facet will gift varied media streams within the same approach as they were originally captured. An example of media synchronization is that the movements of a speaker's lips match the played-out audio.

F. Protocols for Streaming Media:

Protocols square measure designed and standardized for communication between shoppers and streaming servers. Protocols for streaming media offer such services as network addressing, transport, and session management. per their functionalists, the protocols is classified into 3 categories are network-layer protocol (net protocol IP), transport protocol (user data gram protocol UDP) and session management protocol (period streaming protocol RTSP).

III. STREAMING SERVICES TO IOS MOBILE

Different iOS devices square measure taught to access varied streaming services via this example. Our experimental results show that CStreamer will fully eliminate the redundant traffic while not poignant user

perceived streaming expertise. In summary, it makes the subsequent contributions are

- Discover that the present streaming services to iOS mobile devices usually generate 100 percent to seventieth redundant traffic that's damaging to the server (for delivery), the network (for transmission), the mobile device (for battery consumption), and also the mobile user (for money).
- Conducting client-side experiments,to investigate the potential reasons of such redundant traffic. Discover it's chiefly attributed to the restricted memory size of mobile devices, and too quick or too slow network connections.
- Impelled by our findings,to style and implement CStreamer that transparently works between the shopper and also the server. To measure our CStreamer example with varied well-liked net streaming services, and show that CStreamer will fully eliminate redundant traffic while not degrading the user's QoS.

IV. HYPERTEXT TRANSFER PROTOCOL VARY REQUEST AND STREAMING TO IOS

Among the popular mobile devices, iOS primarily based devices square measure leading the market in line with Freewheel, eightieth of wireless video views occur on iOS devices iOS supports 2 streaming protocols are Pseudo streaming and hypertext transfer protocol Live Streaming (HLS) . Pseudo streaming nowadays carries a lot of mobile traffic than HLS, because it is commonly utilized by video streaming services like YouTube and Daily Motion , and YouTube alone contributes twenty seventh of mobile traffic in North America. With Pseudo streaming, the shopper will transfer the media content from associate degree hypertext transfer protocol server. The playback will begin before the complete file is downloaded. The shopper uses hypertext transfer protocol vary requests to request a part of the video file. Associate degree hypertext transfer protocol vary request, or vary request in brief, is associate degree hypertext transfer protocol request with ranges laid out in the header of the request, indicating the required information vary of the requested file. The server solely must respond thereupon a part of information rather than the complete file. However, the complete file will be requested with the vary mere from zero to file size-1. The iOS Media Player identifies itself with the user agents (e.g., AppleCoreMedia/1.0.0). it might 1st raise the server for meta-data info regarding the video file, as well as file size, last changed time. This can be achieved by causing out associate degree hypertext transfer protocol GET request specifying a variety of 0-1. Then, the Media Player would send multiple hypertext transfer protocol requests for the video file, and specifies a variety to transfer in every request.

V. STYLE AND IMPLEMENTATION OF CSTREAMER

The desktop in operation systems, mobile in operation systems nowadays don't use swap/virtual memory to increase memory size. Although the physical memory size is exaggerated from 128 M Bytes in iPhone 3G to 512 M Bytes in iPhone 4S, the matter persists. This can be probably attributable to the exaggerated screen resolution of iPhone 4S that uses a lot of memory for show, and also the exaggerated degree of multitasking on iPhone 4S because the quality level of mobile videos additionally keeps increasing, the restricted memory size is probably going to continue as a bottleneck for net mobile streaming. Further a lot of, iOS could be a closed system, that makes it tough for users to change the iOS Media Player. One could argue that such a tangle is attributable to style pitfall or a software package bug, and may be fastened by software package updates. However, such a tangle is seen in several iOS versions from 3.1.2 to 5.1 with various devices put in change existing software package might not be straightforward and fast. With these concerns in mind, designed a middle ware system, tend to decision CStreamer. With CStreamer, redundant traffic will be eliminated while not dynamical either the iOS Media Player or the numerous media sites that serve videos via Pseudo Streaming.

A. CStreamer Design:

While Pseudo Streaming to iOS generates redundant traffic discover that such development doesn't happen once videos square measure delivered with hypertext transfer protocol Live Streaming (HLS). This can be as a result of in HLS, a giant video file is metameric into tiny segments, every containing generally ten seconds of streaming content. Given the little size of those segments, the iOS MediaPlayer will transfer every phase in one hypertext transfer protocol request and store the downloaded phase totally in memory this can be completely different from Pseudo Streaming, wherever the massive video file being downloaded can not be keep fully given the restricted accessible memory. This implies a simple answer for mitigating the redundant traffic in Pseudo Streaming is convert Pseudo Streaming into HLS. The challenge here, however, such conversions will be drained a clear approach. Fig 2 shows the design of the CStreamer. CStreamer combines associate degree iOS App with a proxy-like CStreamer server. The iOS App works with the CStreamer server to rewrite Pseudo Streaming video links so the Media Player requests streaming information victimization HLS from the CStreamer server. Once the CStreamer server receives such a video request with the rewritten computer address, it downloads the required video from the video server employing a single hypertext transfer protocol GET request. Then it segments the video in line with HLS, and transmits the segments to the iOS devices for playback changing Pseudo Streaming to HLS with CStreamer brings the subsequent benefits are

1) Once Downloading Speed is Fast:

The MediaPlayer requests the media file sharply along with Pseudo Streaming. In CStreamer, however, the MediaPlayer requests file segments consecutive and sporadically. That is, a subsequent request isn't sent out at once following the present one. Rather, it waits for its flip till the playback progress has reached its regular time. this permits the Media Player to require into thought the playback progress once supplying requests betting on the memory accessible at the shopper aspect and also the connection speed, the Media Player requests a minimum of one, at the most five segments previous the present playback. On one hand, this reduces the UN-watched information if the user stops observation within the middle. On the opposite hand, once the accessible memory size is tiny, the request rate isn't as aggressive as in

Pseudo Streaming, and,therefore hypertext transfer protocol requests wouldn't be aborted. Moreover, the Media Player doesn't re-download the start portion of the video once finishing downloading the complete video.

B. CStreamer Implementation:

Our CStreamer example consists of 4 major components are

1) Request Handler:

The Request Handler processes 2 varieties of requests sent by the mobile device are meta-info requests and video requests. For meta-info request (e.g., requesting a file containing video name, duration, and video link), the Request Handler requests the required content from the video server. However, before it delivers the response, it rewrites the Pseudo Streaming link within the response to a brand new computer address are CStreamer URL. This computer address is associate degree HLS computer address that points to a brand new play list file on the CStreamer Media Server. Once the mobile device receives the response containing the CStreamer computer address, if the user decides to observe the video, the Media Player sends out a video request directive for the CStreamer computer address. Once the Request Handler receives such a video request, it calls the Media Downloader.

2) Media Down loader:

The Media Down loader receives the request from the Request Handler. It extracts the first Pseudo Streaming link from the CStreamer computer address, and starts at once to transfer the requested video at the best speed. because the video is being downloaded, the Media Down loader pipelines content to the Media Segment-er, that segments video while not anticipating the transfer to finish. This pipe-lining procedure ends up in a tokenism user perceived start-up delay.

3) Media Segmenter:

The Media Segmenter consists of 2 parts are instrumentation Changer, and Segmenter. Videos deliverable to iOS devices via Pseudo Streaming nowadays, square measure usually place into either MP4 or 3GP format aside from MPEG2-TS utilized by HLS. The video file should be place into MPEG2-TS instrumentation format to be metameric. However, in contrast to video transcoding that is processor intensive and slow, dynamical solely the instrumentation format doesn't need dynamical the audio/video cryptography and is quick enough to be conducted at period of time. The Media Segment er receives pipe-lined output from the Media Down-loader, feeds the info into the instrumentation Changer to alter the instrumentation format. The instrumentation Changer more pipelines its output to the Segmenter that segments the video into segments in line with the HLS specification. The pipelined execution of the Media Downloader and also the Media Segmenter makes CStreamer in no time to arrange the video content once the requested video has been processed, the Media Down-loader and also the Media Segmenter will pass on to method another version of identical video, either in higher quality or lower quality.

4) Media Server:

whereas the Media Downloader and Media Segmenter square measure still process, the Media Server permits the user to transfer and watch the primary phase while not associate degree EXT-X-ENDLIST tag within the play list file. Media Player waits and retrieves the play list once more later from the Media Server that contains updated playback meta-information.

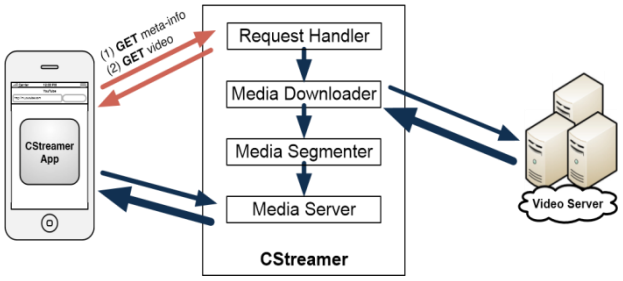

Fig. 1: Diagram

C. CStreamer iOS App:

For associate iOS device to use CStreamer, the tip user will set the CStreamer server as associate protocol proxy to handle the requests. However, manually configuring the iOS device is inconvenient for finish users, and proxying all traffic through CStreamer puts lots of burden on the CStreamer server. To mitigate such drawbacks,got additionally enforced a CStreamer App to finish users, the CStreamer App may be a applications programme. However, it monitors all requests, and identifies meta-info requests. for instance, the request computer address for a YouTube video meta-info .The response to the current request contains a json file with video's Pseudo streaming link in it. The CStreamer App redirects such video meta-info requests to the CStreamer server, wherever the response is rewritten by the Request Handler. CStreamer, to examine the amount between once the video request is shipped and once the primary phase was downloaded to form the comparison a lot of purposeful, to compare a try of experiments that square measure conducted consecutive. the results. For YouTube,discover that the video server is near our testing location thus with Pseudo streaming, it took just one 78 seconds to transfer the initial ten seconds of playback information. With CStreamer, despite the communication between our shopper and CStreamer server yet because the process delay, it took 1.75 seconds to transfer the primary 10-second phase. the same as YouTube,discover that Daily motion doesn't expertise a lot of extra delay either this means the start-up delay, that is vital to user perceived QoS, isn't full of CStreamer. exploitation CStreamer additionally brings another profit. It permits the wireless network interface card (WNIC) on the mobile device to pay longer in low-power sleep mode, and therefore saves battery power consumption. The battery saving comes from 2 aspects are the reduced total traffic quantity and also the bur-sty traffic delivery. for instance, within the YouTube experiment, the WNIC is ready to sleep eighty six.8% (416 seconds) of your time during the 480 second playback; whereas it solely sleeps 85.0% of your time in our succeeding check once look a similar video via Pseudo streaming. For Daily-motion, exploitation CStreamer permits the WNIC to sleep ninety 1.7% (439 seconds) of the time over 478 seconds, whereas it will solely sleep eighty 3.6% time once exploitation Pseudo stream.

VI. CONCLUSION

Through activity and analysis, we discover that there's non-trivial redundant traffic delivered once existing mobile

streaming services square measure accessed on iOS devices driven by the analysis results, style a middle-ware that may transparently cut back such redundant traffic. Net mobile streaming traffic has began to dominate the web mobile information traffic, and it continues to extend with wider adoption of all types of mobile devices exactly delivering streaming traffic to mobile devices isn't solely necessary to the service suppliers and also the net, however additionally necessary to mobile devices (battery power wise) and mobile.

REFERENCES

[1] 1.Effectively Minimizing Redundant Internet Streaming Traffic to iOS Devices Yao Liu1 Fei Li1 Lei Guo2 Bo Shen3 Songqing Chen1 1George Mason University 2Ohio State University 3Vuclip {yliud, lifei, sqchen}@cs.gmu.edulguo@cse.ohio-state.edu bshen@ vuclip.com
[2] Streaming Video over the Internet: Approaches and Directions Dapeng Wu, Student Member, IEEE, Yiwei Thomas Hou, Member, IEEE, Wenwu Zhu, Member, IEEE, Ya-Qin Zhang, Fellow, IEEE, and Jon M. Peha, Senior Member, IEEE
[3] Investigating Redundant Internet Video Streaming Traffic on iOS Devices: Causes and Solutions Yao Liu, Qi Wei, Lei Guo, Bo Shen, Songqing Chen, and Yingjie Lan
[4] "Effectively Minimizing Redundant Internet Streaming Traffic to iOS Devices," http://cs.gmu.edu/_sqchen/open-access/range-tr.pdf.
[5] Y. Liu, L. Guo, F. Li, and S. Chen, "An Empirical Evaluation of Battery Power Consumption for Streaming Data Transmission to Mobile Devices," in Proc. of ACM Multimedia, 2011.

