Audio Streaming on Mobile Phones

Ajinkya Patil, Apurva Mayekar, Shruti Gurye, Varun Karandikar, Pramila Chavan

Abstract— The development of digital communication with increased bandwidths and support of various protocols that allow streaming of multimedia data on mobile devices have created the need to manage and access the multimedia content from mobile devices. A few mobile applications have been developed which readily enable streaming of multimedia files on mobile phones. There are several issues that need to be handled in the process. The paper talks about the critical issues and presents essential concepts with respect to application development for streaming content to a mobile device. It provides a comparison between protocols and audio formats. A system for streaming audio has been proposed exploring various aspects of streaming.

Index Terms— Audio Formats, Interleaving, Mobile, Multimedia, Progressive Streaming, Protocols, Streaming Server.

1 Introduction

Today, the mobile devices do not depend only on network service providers for wireless network access. With support for secondary wireless channels such as IEEE 802.11 or Bluetooth, the devices are able to exchange data with a larger bandwidth and speed. The availability of the secondary channels reduces the cost the user may have to pay compared to the cost when the device is connected to a network over 3G or GPRS. Not only that but the processing power, the battery life and the memory capacities have been improved to a great extent, making the mobile device and PDAs capable of handling continuous streams.

Understanding Downloading and Streaming
In general, audio content can be delivered over a network in two ways:

1. The audio file can be downloaded and then played from the local hard disk of the client.
2. The audio content can be streamed from the server to the client who decodes the received packets in real time, displays the content immediately.

We will discuss the advantages and disadvantages of both the methods in brief.

Advantages of downloading:

- Downloading works with any data rate and allows any audio quality one wants to offer.
- The file is transmitted error-free thus no quality reduction occurs during transmission.
- The download results in permanent storage of the data. This can be reused as per the demand.

Disadvantages of downloading:

- Download times are extremely long if high audio quality is provided over a low-bandwidth network. For example, a five-minute-long music clip encoded with 128kbps may take more than 20 minutes to download over a typical private Internet access with an effective long-term bandwidth of 20kbps. To reduce download times, audio quality must be reduced.
- The user has to load the complete file before he can listen to any part of it. He cannot preview the file to decide whether he is interested in the content.
- There is no possibility to provide a “live” service as rebroadcasting a radio program into the network simultaneously with e.g. the terrestrial broadcast.

Advantages of streaming

- The user can listen to the content immediately after he has demanded for it.
- He can seek to any time frame and listen to that part of the audio content without waiting for the whole file to download.
- It is possible to provide ‘‘live’’ services by encoding the audio signal in real time and sending the resulting audio data stream immediately to the client.

Disadvantages of streaming:

- To stream real time audio data, the transmission line has to provide the full bandwidth of the stream during the whole transmission period. This imposes a limit on the bandwidth and hence the quality of the audio stream.
- The transmitted stream is very sensitive to network load which may cause lost or delayed data packets. This leads to drop-outs in the client audio output.
- To process the required flow control, additional server software (a “streaming server”) is needed.

In particular, if the client on the network has limited memory, eg a mobile device, then the choosing one of the above two methods make a lot of difference with respect to cost and user experience. For such devices, due to their limited memory downloading the entire file may not be feasible. Streaming on the other hand, forms a better solution, which can be implemented by flushing the unused memory time to time. It has to be considered that the main advantage of downloading, the possible high audio quality even over low bandwidth networks, is compensated in practical use by the required high download times.

The drawbacks of streaming, however, can be reduced by dynamically adapting the amount of transmitted data to the actual capacity of the network connection. To do this, the audio data must either be stored in a format that allows dropping parts of the data resulting in a “graceful degradation” or they must be stored in different formats that the server can choose the format best suited for the bandwidth of a given connection.

As far as audio applications on mobile phones are concerned downloading is not a viable option because mobile phones do not have enough memory to download audio data and then play it. Therefore streaming is the only option for mobile phones.

2 Comparison of Protocols

2.1 HTTP (Windows Media Services HTTP Streaming Protocol Extensions)

HTTP Streaming Protocol Extensions are Microsoft Media Server (MMS) streaming which is tunneled through HTTP. HTTP has following features:

HTTP supports TCP because it uses HTTP for all data transmission. It never loses packets. If the network is unreliable, your client will stop and start buffering all over again and fall behind. This means a video being streamed will suffer unwanted delays if it is streamed using HTTP. There will be no loss of frames but this losslessness inot intended in streaming.

The advantage is that error correction is already taken care of. So, there is no additional overhead. Also, the firewalls do not block HTTP. Again, the number of messages that are sent by HTTP are lesser than that of RTSP. E.g. HTTP streaming can be started with two GET requests whereas, RTSP will require at least six messages to do the same task.

RTSP requests are sent over the single TCP connection but HTTP requests require different TCP connections. This means more three way handshakes.

Modes:

Simple mode:

It requires two HTTP connections when the streaming is started. Again, whenever the client pauses, resumes or seeks, one more connection is needed. Load balancing in such a scenario is a problem as each of these connections might be sent to a different server. Streaming protocol is considered to be stateful. It must have a consistent connection to one server. This mode can cause erros while streaming when content is received through a load balancer.

Pipelined mode:

This mode uses the HTTP 1.1 pipelining feature. It send all requests on a single connection. This can reduce the problem of load balancer that occurs with simple mode but cannot totally eliminate it. But, most of the HTTP proxy servers do not support this kind of pipelining, in which case, the simple mode is used by the application that does not involve pipelining.

2.2 Real Time Streaming Protocol

RTSP has the following features:

Packets in RTSP have a choice between TCP and UDP. If packet loss is not an issue, UDP can be used for streaming as it has less overhead compared to TCP.

The encapsulation of Advanced Streaming Format (ASF) packets in RTP is proprietary. The description of the ASF file, called ASF encapsulated in SDP, is proprietary.

If an RTP packet sent over UDP is lost it can be retransmitted in WMS. Thus, the client will not wait for expired RTP packets.As a result, it will not fall behind in streaming if it loses some packets.

A forward error correction (FEC) scheme for RTP packets is also supported for WMS.

A firewall can block ports and protocols that are used by RTSP. This will result into failure of streaming if the firewall exists between the client and server. This is genetally found in home Internet gateways. In case of presence of a built-in RTSP NAT, streaming is ought to fail.

RTSP needs to send multiple requests before the streamed content can actually start playing. But, the client can aggregate together these requests and pipeline them over a single TCP connection.

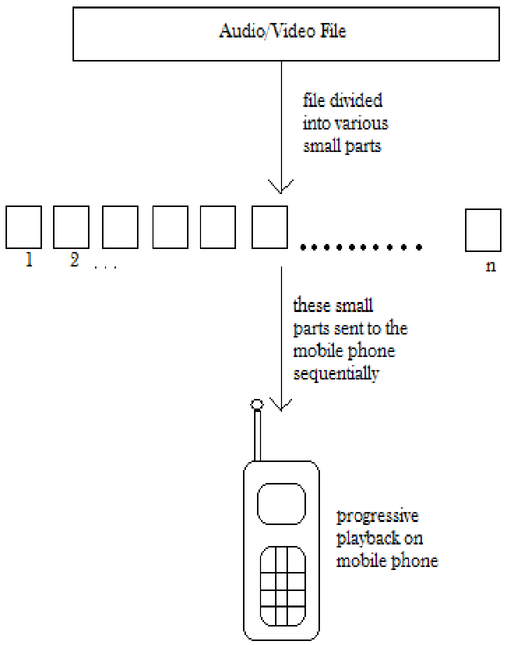
2.3 HTTP v/s RTSP

If the content to be streamed is pre-encoded and end-to-end delay is not a concern HTTP protocol can be used.

If the client runs a desktop OS and the stream is a live broadcast, the end-to-end delay must be reduced. Here, RTSP should be preferred over HTTP.

3 Progressive Streaming

The main problem with the mobile phones is that it does not support progressive streaming. The player will wait until the entire audio/video is streamed and then only it starts playing. Normally mobile phones do not have enough memory to save audio/video therefore the audio/video that the mobile phone can stream and play depends on the memory of the phone.


Audio/Video File
file divided into various small parts
1
2 . . .
n
these small parts sent to the mobile phone sequentially
progressive playback on mobile phone

Overview of Progressive Playback

One way to work around it is to divide the audio/video into many small parts such that they can be easily saved and then played on the mobile phones. This will also allow progressive streaming by sending parts of the audio/video sequentially. For example consider an audio file which is divided into n parts. The server will send these parts sequentially to the mobile client. When the mobile phone receives the first part it will play it then when it receives the second part it will play only after the first part is done playing. Therefore the player will not wait for the entire audio/video to be streamed.

The size of each part can be decided depending on the mobile client, the memory available and the bandwidth of the network. The file may be divided into more number of parts, if serving to a client with low memory. The system can be designed such that to download the initial part, it will spend some time and start playing as soon as done. While it is playing the first part, the consecutive part is downloaded too and played at the right time.

4 Audio Formats

There are a number of audio formats available. The choice of audio format depends upon two factors: Quality of service and Bandwidth required for sending audio.

These two factors are inversely proportional. Thus, there is a trade off involved. For streaming over internet, to avoid unwanted delays, the bandwidth required should be kept minimum. But, reducing the bandwidth will lower the quality of audio. Thus, the aim is to obtain a good quality stream that will take minimum bandwidth.

AMR codec:

AMR, or Adaptive Multi-Rate, is an audio data compression scheme used in speech coding. This open standard format involves compressing of audio data to allow more storage on voice files. Originally developed for GSM (Global System for Mobile), a circuit-switched telecommunication system, it has now been adopted by most cellular companies all over the world.

AMR operates on narrow-band signals (200-3400 Hz) at eight different bit rates - 12.2, 10.2, 7.95, 7.40, 6.70, 5.90, 5.15 and 4.75 kb/s – which are based on frames containing 160 samples and are 20 milliseconds long. AMR is optimized for link adaptation, which enables it to select the best coding technique for better reproduction. The coding techniques that AMR uses are ACELP (Algebraic Code Excited Linear Prediction), DTX (Discontinuous Transmission), VAD (Voice Activity Detection), and CNG (Comfort Noise Generation). Hence the name Adaptive Multi-Rate.

As AMR offers eight different bit rates, it can help during network congestion improving the Quality of Service. It is known for its robustness to loss of packets, bit errors and is more immune to background noise

Advantages of AMR:

-AMR provides the best end-to-end solution to various applications and its flexible bit rate enables error correction

-AMR can be used in multi-media applications after being encapsulated in 3GP or MPEG-4 file formats

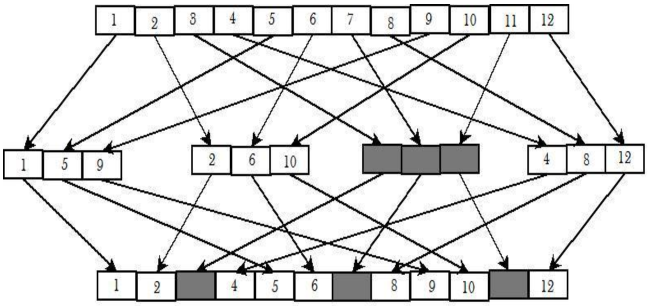
-AMR can easily be applied in low bit rate speech coding

-AMR has the most consistent performance across all major languages

5 Issues in Progressive Streaming

Following are the issues that are encounter by using the progressive streaming algorithm that we suggested:
1. Loss of data during transmission
2. Size of the chunk in which the audio is cut
3. Static slicing of the audio vs. dynamic slicing of the audio

First we will discuss the issue of loss of data during transmission. Suppose the audio is cut into 10 parts and is transmitted and while transmission some of the parts are lost. Since this is progressive streaming we cannot request for the lost part because then we will have to wait for the audio part because that will take a lot of time and will create a gap when the audio is played. Now we need to compensate in some way for the last part or else there will be an abrupt gap when the audio is played. We can use interleaving along with an error concealment algorithm.



What happens in interleaving is that the audio is divided into very small chunks and then they are interleaved as follows:

Therefore if one of the parts is lost there will not be a long gap in the audio that is played. These small gaps can be compensated by using an error concealment algorithm.

Another issue of upmost importance is the size of the chunk in which the audio is cut. If the size is too small it will create a lot of network traffic by sending an individual part per packet. Alternatively if the chunk size is too large and if a part is lost during transmission then there will be a long abrupt gap in when the audio is played. Also the mobile phone may not have that much memory to temporarily save an individual part of the audio file. Therefore the size of the chunk should not be very small neither should it be very long.

The last topic of concern is whether to adapt static slicing of audio or dynamic slicing of audio. In static slicing the audio will be cut beforehand and then it will be saved in the database. Hence in static slicing when a request comes in the chunks will be retrieved from the database and then it will be streamed. Disadvantage of static slicing is that the time required to retrieve each part may exceed when the audio file is very large and is divided into many small chunks. Another disadvantage is that later if we want to change the chunk size then we will have to retrieve all the chunks from the database and then merge it and then again cut it according to the new size and save it in the database. This will be a very time consuming task if the database contains a large amount of audios saved in it. In dynamic slicing the audio file is saved as a whole unit. So when a request for an audio comes in the cutting of the audio is done dynamically.

6 Media Streaming Protocol for Reducing Frame Errors

The following three methods are used in media streaming protocol to reduce frame errors:

1. Link-Layer Retransmission

In EGPRS there is an option of using Automatic Repeat request (ARP). This ARP is used at the data link layer. Therefore it enables us to create a reliable pipeline for transmission of audio data over radio-link. The network has the capability of choosing the maximum number of retransmissions for each radio block. This number in many practical applications is set to 3.

2. Dynamic Packet Assignment

EGPRS wanted to efficiently support bursty data therefore it provides a facility of dynamic packet assignment (DPA). In DPA a link is assigned different channel dynamically which depends on the channel quality at the receiver. Initially each mobile is assigned to a channel. It continues to use the same channel till M consecutive blocks error occurs. Here M is a constant which is assigned to 2 in most of the cases. When M consecutive block errors occur the mobile is reassigned to another channel. DPA increases the capacity of the system therefore it offers better coverage.

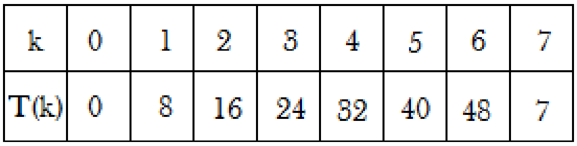
3. Packet Shuffling

Shuffling or interleaving the frame and then transmitting increases the effectiveness of the error concealment algorithm. Shuffling and interleaving is done at the server side or it can be done at some other intermediate node. Reassembling of the frames takes place at the receiver and after reassembling the frames are placed in the play out buffer. This method helps in dispersing the effect of frame burst error. The frame number should be included in the frame so that we can hide the shuffling and interleaving from the mobile device. This hiding is required so that packet shuffling feature can be activated without changing the mobile. The shuffling technique is based on convolution interleaver which is defined by two interger parameters, N and B.

Let us take an example where N = 7 and B = 1. If T (k) denotes the position at which frame k is transmitted, then we have:

T (k) = k + (k mod N) NB

The order of transmission of the frames is 0, 8, 16, 24, 32, 40, 48, 7, 15, 23, 31, 39 etc .Using this technique two consecutive frames are separated by at least NB frames after the shuffling.



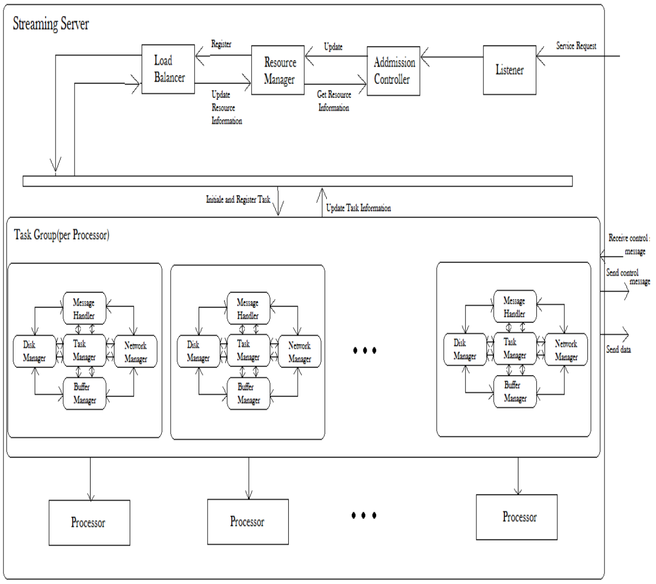
k	0	1	2	3	4	5	6	7
T(k)	0	8	16	24	32	40	48	7

Convolutional interleaving for N = 7, B = 1.

This technique is very effective in shortening gaps in the received frame sequence that result from burst transmission errors. This technique also introduces a delay in the transmission of some frames. The choice of interleaver should provide a suitable trade-off between the frame error spreading and the reduced buffer length.

7 STREAMING SERVER

The process of streaming can also be achieved by the use of server side processing. Streaming servers such as Darwin Streaming Server, IceCast, Ampache process the audio data available on the server to create a stream of data.



On request, this data can be made available to a mobile client via different protocols including rtsp and http. The client hardware needs to support rtsp protocol suite in order to get served by an RTSP server.

8 STREAMING SERVER ARCHITECTURE

A streaming server is basically structured for a multiple CPU platform that has the capability to provide support to many concurrent media streams.
The design of the server is based on:
1. Efficiency
2. Reliability
3. Scalability

Figure above illustrates the server structure. The main parts of the server are as follows:
1. Admission controller
2. Resource manager
3. Load balancer
4. Task pool
There exists a task pool for each processor. Actual media streaming is carried out by the task which is allocated to the processor.

The main components of a task pool are as follows:
1. Disk manager
2. Network manager
3. Buffer manager
4. Message handler
5. Task manager

A service request by any user is handled as follows:
1. A listener basically listens to new service request. So when it receives a new service request it passes it on to the admission controller.
2. The job of the admission controller is to decide whether to service the request or reject it. The decision is based on the availability of system resources. A request will be serviced only when there is enough resources available to service it. The server has to base its decision on resource availability because it has to guarantee quality of service (QoS) to the existing user sessions.
3. The admission controller gets the status of the system resources from the resource manager. The resource manager maintains various data related to CPU usage, disk bandwidth, network bandwidth and memory size. It is the job of the resource manager to maintain latest information for various system resources.
4. If the service request is admitted resource usage is updated and the load balancer selects a processor to execute the newly admitted request. The main job of the load balancer is to intelligently balance the load between different processors.
5. The job of the task manager is to schedule the stream request so that it meets its timing constraints.
6. When the server gets a request from the client it transfers a fixed size media from the disk to the buffer and then analyses the stream and transfers the stream to the network depending on the media rate.
7. Each packet in the server is associated by a timestamp that denotes the time before which it must be transmitted to the network. The local timestamp which is there in each packet is added to the system which then it is added to the time-lined job queue where requests which are sorted according to the final timestamps are kept.
8. The job of the scheduler is to examine the queue constantly after a specific small time interval and to send the media packets whose timestamps are within the current scheduling period.

When the system capacity is exceeded by allowing excessive number of system requests then the quality of service may not be guaranteed. The admission control mechanism in the streaming server should be very strict so that it does not allow users in exceeding numbers which the server cannot handle. The decision should be based on the current resource availability at the server side. The admission decision should be such that it does not deteriorate the QoS requirements of the clients which are already being serviced by the server. The admission control mechanism does not only depend on the current system

resource availability but it also depends on the resource of the new request. The processor usage and disk access bandwidth vary dynamically depending on the way that the client behaves. It also depends on the underlying operating system. But the resource requirements of a new stream are known beforehand. Since the resource usage is dynamic and depends on the operating system and the user behaviors the resource usage table must also be dynamically updated. The updating of the resource tables is done by the server. The information from this table is used by the admission controller to admit new requests to the system.
The criteria that are used for the resource availability are:
1. Processor usage
2. Memory capacity
3. Disk bandwidth
4. Network bandwidth

4 CONCLUSION

Playing audio files on mobile device should be achieved through streaming as it will reduce the user waiting time involved in downloading. The effect of streaming can be achieved through progressive streaming that does not involve a streaming server, which can also be used alternatively. AMR turns out to be the most suitable audio format for streaming as it will require minimum bandwidth with a good audio quality compared to other audio formats. HTTP protocol can be preferred over RTSP as it incurs less overhead. In case a streaming server is used, various methods like link layer retransmission, dynamic packet assignment, packet shuffling etc. should be used.

References

[1] Anna Kyriakidou, N. K. (n.d.). Video Streaming for fast moving users in 3G mobile networks. *Univ. of Athens, Greece*, ACM.

[2]Christopher E.Hess, R. H. (n.d.). Media Streaming Protocol: An Adaptive Protocol For The Delivery Of Audio and Video Over the Internet. *Department of Computer Science, University of Illinois at Urbana-Champaign,Urbana, IL 61801*, IEEE.

[3]HOJUNG CHA, J. L. (2005). A Video Streaming System for Mobile Phones: Practice and Experience. *Department of Computer Science, Yonsei University*, ACM.

[4]Kapil Chawla, P. F. (n.d.). Transmission of Streaming Data over an EGPRS Wireless Network. *AT&T Laboratories - Research*, ACM.

[5]Karl Jonas, P. K. (n.d.). Audio Streaming on the Internet Experiences with Real-Time Streaming of Audio Streams. *German National Research Center for Information Technology*, ACM.

[6]Samjani, A. (2002). General Packet Radio Service, IEEE.

[7]Samjani, A. (2001). Mobile Internet Protocol, IEEE.

[8]Sumit Roy, M. C. (n.d.). A System Architecture for Managing Mobile Streaming Media Services. *Hewlett-Packard Laboratories*, IEEE.

[9]VAZQUEZ, M. (n.d.). A Mobile Audio Messages Streaming System. *National Institut of Telecommunications*, ACM.

[10]YE WANG, W. H. (N.D.). A FRAMEWORK FOR ROBUST AND SCALABLE AUDIO STREAMING. *SCHOOL OF COMPUTING, NATIONAL UNIVERSITY OF SINGAPORE*, IEEE