REAL-TIME AUDIO CAPTURE, COMPRESSION & STREAMING SERVICE ON A PDA

C. Garcia, F.J. Suarez
Universidad de Oviedo
Campus de Viesques, 33204 Gijon SPAIN

Keywords: Mobile Multimedia Service, Audio Compression, Audio Streaming, Performance Tunning.

Abstract: This paper shows how a PDA (Personal Digital Assstant)can be converted into an audio source within a private network or the web, providing capture, compression and streaming of audio as a real-time mobile service. Once the sound around the PDA has been captured and compressed to mp3 format, the service allows it to be broadcast to a Streaming Server. Once the audio reaches the Streaming Server, anyone with a network connection is able to receive and play it. The service provides different configuration parameters to control audio quality and broadcasting performance. For audio quality, different bitrate and frecuency values can be chosen. For broadcasting performance, different packet-length values can also be chosen, and the bitrate mode can be automatically controlled. The paper also includes the performance tunning of the compressor and experimental results using both wired and wireless networks.

1 SERVICE DESCRIPTION

PDAs provide now a wide variety of mobile connections, from standard modem connections to wireless connections. Taking advantage of their mobility and connection capability, a PDA can be converted into an audio source within a private network or the web, providing capture, compression and streaming of audio as a real-time service. For instance, this kind of service could be useful to broadcast live interviews, news reports or press conferences.

Once the sound around the PDA has been captured and compressed to mp3 format (Robinson and Hawksford,) (Branderburg and Popp,) (Rangachar, 2001), the service allows it to be broadcast to a streaming server. The service uses Real-time Transport Protocols (RTP/RTCP) and Session Description Protocol (SDP). These protocols allow the audio to reach the streaming server. The streaming server must be RTP and SDP compatible, which is the case of the Darwin Streaming Server (Apple) (Computer,) and the Helix Server (RealNetworks) (RealNetworks,). Once the audio reaches the streaming server, anyone with a network connection is able to receive and play it.

Instead of broadcasting to a streaming server, the audio can be saved in a file and broadcast later. An improved use of free memory and long-time recording capability are the profits of saving voice messages in this way.

The service provides different configuration parameters to control audio quality and broadcasting performance. For audio quality, different bitrate and frequency values can be chosen. For broadcasting performance, different packet-length values can also be chosen, and the auto-bitrate mode can be activated.

One of the main advantages of the service is the auto adaptable control of the bitrate (in the auto-bitrate mode), which provides the best bitrate compression at all times. This means the bitrate will lower or rise as the available bandwidth does.

The rest of the paper is structured in the following sections: service operation, audio compression (including compressor performance tuning), audio streaming, protocols (RTSP, RTP, RTCP & SDP), implementation details, experimental results (with both bluetooth and modem communication), concluding remarks and references.

2 SERVICE OPERATION

Figure 1 shows what can be achieved with the service. Obviously, the system is simpler for recording voice

J. Suarez F. and Garcia C. (2004).
REAL-TIME AUDIO CAPTURE, COMPRESSION & STREAMING SERVICE ON A PDA.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 345-350
DOI: 10.5220/0001391203450350

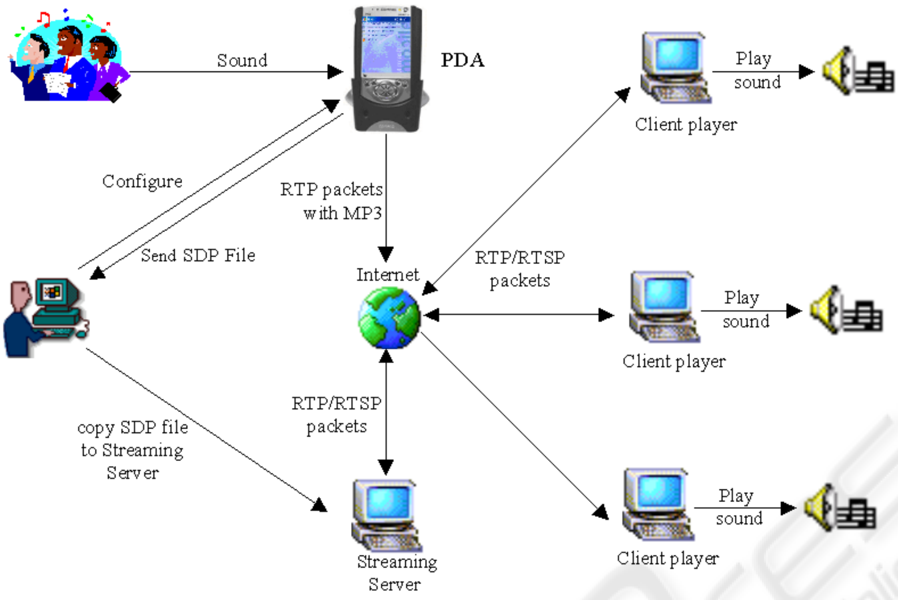

Figure 1: Service overview

messages only, but when an audio-conference takes place, the whole system comes into play. The SDP file must be copied to the streaming server public directory. It need not be copied again unless a different streaming server is used. Mp3 frames travel within RTP packets[1], first from the PDA to the streaming server, and then from the streaming server to the clients. The clients send a request to the streaming server using RTSP protocol in order to start receiving the RTP packets.

The Streaming Server and the client's players must be compatible with the protocols used. The Apple Quicktime or Darwin Streaming Server 3 work well, as does the latest version of the Real Server (Helix). The Winamp player with an RTP plug-in and the RealOne player also work well. RealOne is also available in Pocket PC.

So this service is capable of broadcasting the sound around the PDA to anyone with a PDA or a PC with an appropriate client player.

3 AUDIO COMPRESSION

The open source mp3 encoder used is Gogo-no-Coda 3.01, which is the highest speed mp3 encoder available. It has many optimizations and is derived from Lame 3.88. In the context of this work, the source code has been modified, changing some parts of the assembler code to C code, Linux code to Windows and Pocket PC code, and deleting some unnecessary code, such as stereo, joint stereo and VBR sections.

The performance of the encoder has been tested on an AMD K6-2 450MHz and also on the slower Pentium 60MHz. using the AMD the encoding speed is 5X. However, using the Pentium, the encoding speed reaches only 0.8X. Unfortunately, although the PDA processor is an StrongARM 206 MHz, it's encoding speed is below 0.5X. To explain these poor results, some integer and floating point operations have been tested (see Figures 2 and 3). Using integer operations, the StrongARM processor is slower than the Pentium although it has three times more MHz. But the difference is greatly exaggerated when dealing with complex floating point operations. The absence of a FPU and a memory cache of only 16 KBytes explain the bad performance. This could also explain the current lack of mp3 encoders for PDAs.

3.1 Compressor performance tuning

Two optimizations were carried out to create a Real-Time mp3 encoder.

- The most complex function is the fast Fourier Transform (FFT). This is executed once per frame, and analyzes every block of samples to be encoded. The improvement consists of executing it with only the first block of samples, the remaining blocks using the results of the first block analysis.
- X raised to 3/4 is the next most time consuming function, because it is executed once per sam-

[1] RealOne is only compatible with one mp3 frame per RTP packet.

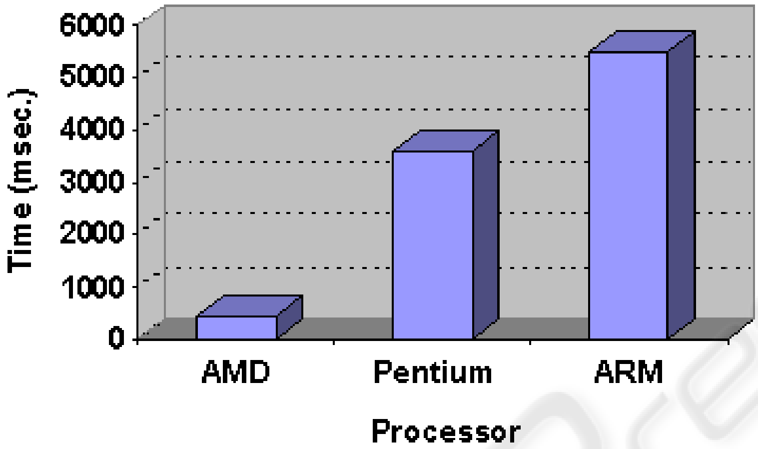

Figure 2: Integer performance

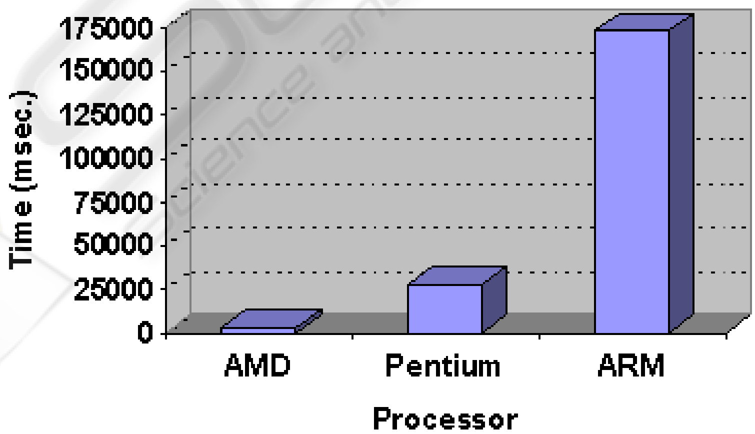

Figure 3: Floating point performance

ple. Thus, if the frequency of capture is 11,025 Hz, the function is executed 11,025 times per second. Furthermore, this function is implemented with two SQRT calls and a multiplication using floating point arithmetic. By joining different lineal functions, a new, much less complex function replaces the original one with minimum error.

4 AUDIO STREAMING

Audio streaming allows users to play audio without previously downloading an audio file. They simply listen to the audio data as they receive it.

A user can start a live streaming session by making a RTSP[2] (Schulzrinne et al., 2002) request to the streaming server for the required SDP file[3]. As soon as the streaming server receives the request, the user can begin to receive RTP packets sent by the PDA application and listen to the audio.

4.1 Real-Time Transport Protocol (RTP)

RTP (Schulzrinne et al., 1996) (Schulzrinne, 1996) is suitable for real time transmissions, like audio or video. It usually runs on top of UDP protocol, so it does not guarantee quality delivery service. This means packets can be lost or disordered, however it does provide a way to detect these events.

RTP uses an auxiliary control protocol (RTCP), which provides feedback about service quality, and basic user information.

The application developed has a basic unicast implementation, in order to consume the lowest cpu power. As mentioned in Section 3, StrongARM cpu power is poor. RTCP has also been implemented, but seems to be unnecessary. RTCP allows the streaming server to know whether a user is active; if a user doesn't send RTCP packets to the streaming server, the server stops sending him RTP packets. However, in this service the streaming server doesn't actually need to receive RTCP packets to know the encoder application is active, because the streaming server is already receiving the RTP packets. As a result, the service also works without sending RTCP packets. Furthermore, RTCP packets sent by the streaming server are also unnecessary (see (Schulzrinne et al., 1996) for more details).

[2]Real-Time Streaming Protocol

[3]SDP files are used for live streaming sessions in Quick Time Streaming Server and they are also compatible with Real Helix Server.

4.2 Session Description Protocol (SDP)

SDP (Handley and Jacobson, 9968) is necessary for the streaming server to receive the audio packets and reflect them to clients. It is also necessary in order for the clients to be able to request from the streaming server live streaming audio packets.

The application developed generates the SDP file needed to start a live streaming session.

5 IMPLEMENTATION DETAILS

Fig. 4 shows how the service implementation works. It has been simplified for better understanding. For instance, it doesn't show how it treats RTCP packets or mp3 files.

The application is multithreaded and uses synchronization between threads. First of all, there is a thread that captures audio samples using API functions. It saves blocks of samples in a buffer and signals another thread which compresses the audio samples of the shared buffer. When the audio compression finishes, it stores mp3 frames in another buffer, and signals a final thread whose task is to generate RTP packets containing the mp3 frames and send them to the streaming server.

This behaviour corresponds to ideal conditions. If conditions are degraded, such as a degradation of the available bandwidth or CPU, then the buffers will fill. At this point the auto-bitrate mode will start to operate, if it is activated, changing the bitrate to a lower value if a degradation appears. This has two advantages, the mp3 compression is faster and the bandwidth requirement descends. On the contrary, if the buffers are nearly always empty, the auto-bitrate mode will change the bitrate to a higher value in order to get more audio quality. So, in fact, this option provides the best audio quality for the available bandwidth at all times.

Development for Pocket PC (Microsoft O.S. for PDAs) (Grattan and Brain,) is slower than for Windows O.S. This is mainly because the source code is written in the Embedded Visual Tools environment and then the binaries are generated and downloaded to the PDA. This process is necessary each time the source code is changed in order to test the execution in the PDA. There is also a Pocket PC emulator for Windows, but is unable to emulate all the PDA functions, for example some communication functions.

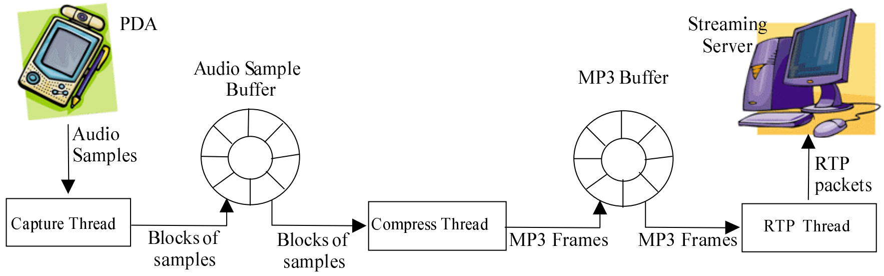

Figure 4: Service implementation

6 USER INTERFACE

The application interface consists of a main menu that presents various options and help. From this menu, users can start and stop a recording, start a mp3 file transmission instead of sending live audio, or generate an SDP file. Many options can be configured, as can be seen in Fig. 5.

The streaming server IP must be placed in the textbox. A name can be chosen for the mp3 file in which audio is to be saved, and for the mp3 audio source file. There are three checkboxes, which can be checked for live transmissions, file recordings and to activate the auto-bitrate mode. There are also three sliders, with which users can choose the compression bitrate, the audio frequency and the number of frames within an RTP packet. This number goes from one frame per packet, to the number of frames that fit in 1.5 Kbytes. If only one frame per packet is chosen, many packets per second will be sent. Each packet has an RTP header, so more data is sent and more resources are required.
There is also a status bar which shows important information during the recording process, such as the bitrate status, the audio frequency and the buffer status.

7 EXPERIMENTAL RESULTS

The service has been tested with two Compaq iPAQ PDAs: one 3800 (206MHz CPU) and one 3950 (400MHz CPU). A bluetooth and modem connections have been tested in order to check the audio streaming. In every experiment, three computers were used: one with the Darwin Streaming Server installed, and the other two with client players; a Winamp player and a RealOne player.

First a network connection is established. Then, the IP of the PC where the Darwin Streaming Server is running is introduced into the PDA application. The auto-bitrate mode option is always activated and the bitrate value is always set to the maximum before starting recording. The best RTP packet length is the maximum, as was said in Section 5, but for RealOne compatibility it has to be set to the minimum, that is, one frame per packet.

The auto-bitrate mode is unnecessary for file recordings only. In addition, StrongARM 206MHz has enough performance for recording audio to a mp3 file with the maximum quality provided.

7.1 Bluetooth communication

The bluetooth connection provides bandwidth enough for the maximum audio quality. Under normal circumstances the application maintains maximum quality until it is stopped. The auto-bitrate rarely needs to change the bitrate in the iPAQ 3950. However, with the iPAQ 3800, for maximum audio quality, there should be a higher RTP packet length value than only one frame per packet.

7.2 Modem communication

The modem connection does not assure the minimum bandwidth at every moment; sometimes the real bandwidth may be 24 kbps, going down as low as 0 kbps for some seconds, then raising to 16 kbps, and so on. For RealOne compatibility, 8 KHz is advisable in order to balance the higher resources consumed for the minimum value of the RTP packet length. If RealOne compatibility is not needed, the maximum value of the RTP packet length is more appropriate. Modem connections truly benefit from the auto-bitrate mode, which ensures the audio leaving the PDA in real-time in the event of the bandwidth going down, avoiding the bitrate being higher that the available bandwidth. It is preferable for audio to reach clients with lower quality than not at all or with many breaks.

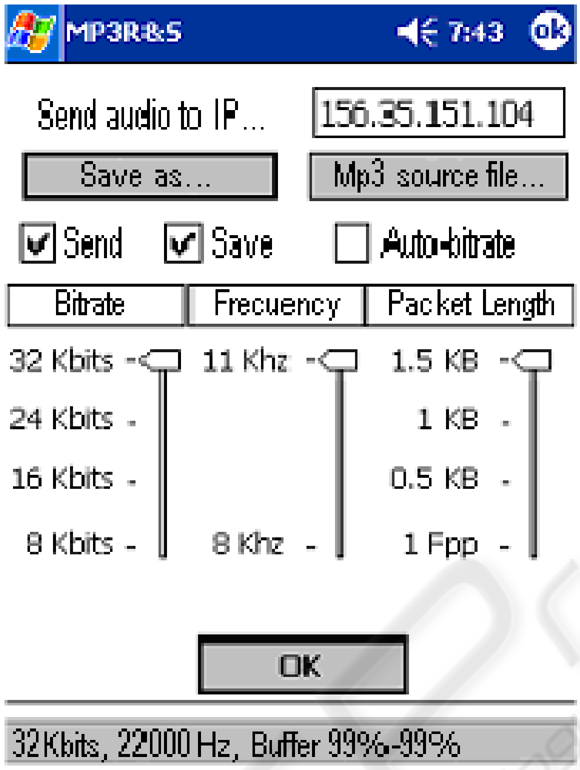

Figure 5: Preferences window

8 CONCLUDING REMARKS

This service converts a PDA to an audio source within a network, and also provides a stand-alone mp3 recorder. A PDA could be placed, for example, in a conference room and it would capture the audio, compress it to mp3 and send it to a streaming server to be delivered to users with access to the server. In summary, the service includes:

- A stand-alone mp3 audio recorder.
- A mp3 live audio source for all users connected to Internet.
- A deferred audio transmission of mp3 audio files previously recorded.
- An auto-bitrate mode which guarantees the maximum quality of live audio transmission for the resources available at any given moment.

REFERENCES

Branderburg, K. and Popp, H. An introduction to mpeg layer-3. Technical report, Fraunhofer Institut für Integrierte Schaltungen (IIS).

Computer, A. Quicktime streaming. Technical report, Apple Computer, Inc.

Grattan, N. and Brain, M. *Windows CE 3.0 Applicattion Programming*. Prentice Hall.

Handley, M. and Jacobson, V. (19968). Sdp: Session description protocol. Technical Report RFC 2327.

Rangachar, R. (2001). *Analysis and improvement of the MPEG-I audio layer III algorithm at low bit-rates*. PhD thesis, Arizona State University.

RealNetworks. Media delivery, the helix server. Technical report, RealNetworks, Inc.

Robinson, D. J. and Hawksford, M. O. Psychoacustic models and non-linear human hearing. Technical report, Centre for Audio Research and Engineering, Department of Electronic Systems Engineering, The University of Essex, The University of Essex, Wivenhoe Park, Colchester CO4 3SQ, United Kingdom.

Schulzrinne, H. (1996). Rtp profile for audio and video conferences with minimal control. Technical Report RFC 1990, Columbia University.

Schulzrinne, H., Casner, Frederick, and Jacobson (1996). Rtp: A transport protocol for real-time applications. Technical Report RFC 1889, Columbia University.

Schulzrinne, H., Rao, A., and Lanphier, R. (2002). Rtsp: Real time streaming protocol. Technical Report RFC 2326-bis2, Columbia University.