MP3 streaming over Bluetooth to multiple users

Mritunjay Singh, Tata Consultancy Services, TelecomIP (PAN Group), Advanced Technology Center, Hyderabad, India mritunjay.singh@tcs.com
Trish Messiter, Clarinox Technologies, Melbourne, Australia, trish@clarinox.com
Gokhan Tanyeri, Clarinox Technologies, Melbourne, Australia, gokhan@clarinox.com

ABSTRACT

Wireless has been well received ever since the first wireless radio transmission over the Atlantic Ocean by Marconi almost a century ago. Since that time the technology has been harnessed to suit a myriad of applications. The appeal is in the freedom given to the user to be mobile. A relative newcomer, Bluetooth, has made real time audio/video transmission in mobile and pervasive environments possible. In comparison with many other wireless standards, Bluetooth is more cost- and power-effective, making it ideal for small and light mobile devices.

However, the use of Bluetooth is not without challenges such as limited bandwidth, high degree of error rates, and the time-varying nature of the radio link. This paper discusses the implementation of streaming of MP3 Audio data via Bluetooth, and the issues pertaining to it, initially as a theoretical exercise and also through an example implementation.

In this paper we propose the implementation of a Generic audio distribution transport profile for streaming MP3 data over Bluetooth to multiple users. The profile designed is based on the Audio video Transport Protocol (AVDTP) [2] specifications of the Bluetooth Special Interest Group. The structure of the Media Transport and Reporting packets used in AVDTP is based on the Real Time Transport Protocol (RTP) [7] and Real Time Control Protocol (RTCP) [7].

Keywords

Audio, ACL, AVDTP, Bluetooth, L2CAP, MP3, QoS, RTCP, RTP, SEP, SEID, SCO

1. INTRODUCTION

Bluetooth technology is ideal for networking of all sorts of electronic devices within 100m of each other. In fact several classes of Bluetooth device exist to provide for different coverage areas; these are class 1, 2 and 3 providing approximately 100m, 30m and 10m respectively.

A distinguishing feature of Bluetooth is the sophisticated service discovery mechanism that allows for devices to establish connection, transfer data and disconnect without requiring user intervention. User approval may however be required for authentication and security purposes. Bluetooth has elaborate security mechanisms to enforce authorization, authentication and encryption built in.

Bluetooth also caters for many devices to co-exist in the same space. These devices may be part of the same piconet, they may be connected in scatternet topology, or may have no connection to each other at all. Whichever is the case they can co-exist without interfering with the operation of each other.

The standard defines all layers of the OSI protocol stack layer, which translates to a focus on application development inherent in the specification design.

The Bluetooth low power rates of around 1mW Class 3 (100mW Class 1) during transmission make it suitable for use for handheld devices. Added to this is various low power modes utilized during idle periods which can be under the control of the application.

Bluetooth wireless specification includes a physical layer, link layer and an application layer, which support data, voice and content-centric applications. The Bluetooth link-layer supports two types of link modes between the master and slave devices, namely:

- Synchronous Connection Oriented (SCO) [9] links
- Asynchronous Connection-Less (ACL) [9] links

SCO is a symmetrical point-to-point circuit-switched connection with up to 64Kbps bandwidth, typically used in QoS-guaranteed bi-directional data transfer like voice transmission. ACL links can support point-to-multipoint communication with up to 723Kbps bandwidth one direction, which is suitable for streaming media using MP3 and other codec formats.

Bluetooth specification defines a protocol stack to transfer data and implement advanced features required by applications. The protocol stack consists of several layers, starting from the physical layer, comprising the Radio, BaseBand, Link Manager, Logical Link Control and Adaptation Protocol, and Host Controller Interface as shown below, in Figure1. The functions of the various layers mentioned are:

- Baseband is the physical layer where Bluetooth performs all low-level data processing including basic Forward Error Correction (FEC) and Automatic Request for Transmission (ARQ), packets handing, data whitening, hop selection and security.
- Link Manager Protocol (LMP) [9] handles link control, power-sensitive states changing, and data encryption.
- Logical Link Control and Adaptation Protocol (L2CAP) [9] provides both connection-oriented and connectionless data services to upper layer protocols with segmentation and reassembly operation. L2CAP only supports ACL links with packet size up to 64Kbytes. L2CAP and LMP serve as a Media Access Control (MAC) layer.
- Host Controller Interface (HCI) [9] provides a uniform interface method to access hardware capabilities. It is responsible for transmitting data between L2CAP and baseband through a physical bus (e.g., USB, RS232 and PCI), using LMP.

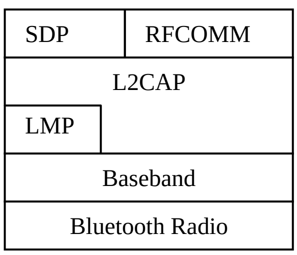

Figure 1: Protocol stack of Bluetooth

The service discovery protocol enables applications to discover the services and service characteristics that are available on other Bluetooth devices without any prior configuration. The functionality of the different layers of the stack is built in a profile.

A Bluetooth profile is a specification that defines the minimum requirements that the Bluetooth device must support in a specific usage scenario. These requirements define the end-user services, features and the procedures that the Bluetooth device must support, to enable interoperability between peer devices.

Audio streaming over Bluetooth is also developed as a profile based on the Audio Video Transport Protocol Specification. Audio streaming is defined as playing the audio data incrementally, as the user receives the data fragments from the audio file present in the Media server and plays it without waiting for the entire file to be downloaded.

2. ARCHITECTURE OF GENERIC AUDIO DISTRIBUTION PROFILE OVER BLUETOOTH

MP3, an audio compression technique [4, 5, 6] is based on a perceptual coding scheme to remove redundant information from the digitized audio sequence. This technique is based on a psycho-acoustic model, using the masking effect, trying to mimic the human ear's behaviour. As the bandwidth of a Bluetooth link is limited to 723 Kbps, the raw audio data must be compressed before transmission to achieve efficiency. Figure2 below represents the streaming architecture over Bluetooth with their functional components.

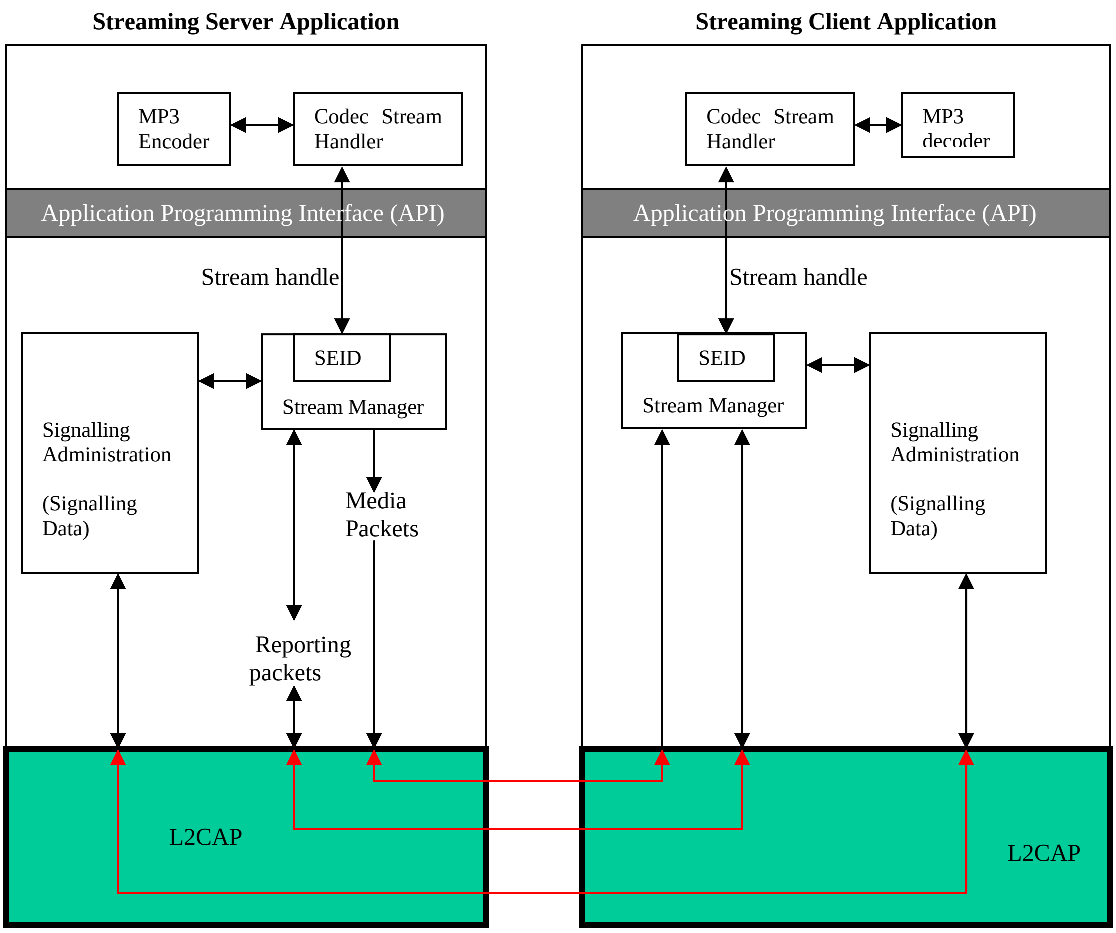

Figure 2: Streaming Profile Architecture over Bluetooth

2.1 Audio Distribution Profile Functionality and Implementation

The signalling administration block and the stream manager are the important functional components developed in an audio streaming profile, which handles streaming and related functionalities.

Applications using this profile must provide/handle already encoded media data as it is exchanged via the profile interface. The profile is built on Bluetooth specifications, covering the AVDTP [2].
AVDTP helps:

- discover the capabilities of the device
- negotiate audio/video stream set up
- establish and tear down the stream
- message for real time streams

The audio streaming profile developed using AVDTP provides an interface to the upper layer for the streamed data and creates a transport layer to provide stream connection to the application. An audio stream on a Bluetooth link is sent or received by the Bluetooth devices via an abstract Stream End Point (SEP), which is referred by a Stream End available transport service and application service capabilities in order to negotiate a stream.

The transport service capabilities correspond to more specifically transport related services inside the transport layer. These comprise of framing and segmentation, encapsulation, packet loss detection, packet recovery and some optional capabilities.

Application service capabilities correspond to the services provided to the application layer. These services include negotiation and configuration of source codecs and some optional capabilities.

The description of services is organized in a hierarchy of service categories and service parameters. The service category represents top-level structuring element. Examples of service categories are media transport, reporting and media codec.

Each service category can comprise one or multiple service parameters, which may apply to various options. For example, the service category Media Codec may imply to service parameter as "Media Type", "Media Codec Type" and "Media Codec Specific Information Elements".

The Audio streaming profile implemented uses MP3 [5], [6] for streaming. The Media Type indicates audio, and Media Codec Type indicates MP3. The Media Codec Specific Information Elements include the Layer, CRC Protection, Channel Mode, Sampling Frequency, and Bit Rate Index supported by MP3 Codec [6].

SEID represents a cross-device reference to a specific stream. This reference is used in the signalling transactions of the air interface, between peer AVDTP entities.

A stream represents a logical end-to-end connection of streaming media data between two A/V devices. Streaming data is source-encoded data, as the audio or video encoder- /decoder conceptually resides in the upper layer. The Audio Streaming Profile developed, uses MP3 audio data for streaming. It provides an interface to the application, to create a stream end point.

The application obtains a unique stream handle for the stream end point and associates a stream handler to it. A callback function is handling the stream related messages sent by the audio profile. The stream handle is a local identifier and represents a top-level reference to the stream.

The peer devices explicitly assume the role of source and sink. In the implementation, the server assumes the role of source and since MP3 data is present in the server, the client assumes the role of sink.

The application derived from the Generic Audio Video Distribution Profile can register itself to publish its services in the SDP ([9]) database. The client device performs a remote service search to retrieve the service characteristics of the audio service, registered in the SDP database of the server device using the Service Discovery Application Profile.

The client application then establishes a signalling channel using the signalling administration block of the profile, to perform stream management signalling.

2.1.1 Signalling Procedure

Stream management signalling requires a L2CAP channel. The L2CAP channel is established over the ACL link with a Protocol Service Multiplexer Value (PSM) allocated for AVDTP transactions. The stream management signalling consists of discover stream end point capability, stream configuration, stream establishment and stream start procedure.

2.1.1.1 Discover Stream End Point Capability

A Client device discovers the SEPs of the remote server device by the Stream End Point Discovery procedure. The result of this procedure provides a list of SEPs along with the media types (audio/video) that the remote device supports. Using the SEID as a reference, the client device can query the media codec capabilities of the remote SEP using the Get Capabilities Procedure [2].

2.1.1.2 Stream Configuration

The client device selects the required remote stream end point indicated with remote SEID and its service capabilities for receiving an audio stream from the remote device, after receiving the remote capabilities using Get Capabilities procedure [2].

The client device then configures the SEP of the remote server device using Set Configuration procedure [2]. The Set Configuration procedure is used to notify the local stream end point and its service capabilities to the server. The service capabilities comprise of the Application and the Transport Service capabilities.

2.1.1.3 Stream Establishment and Start

The client device opens a streaming connection to the server for a configured SEP and the associated transport session for media packets and reporting packets. A new L2CAP channel may be established and uniquely mapped to that transport session.

The L2CAP channel acts as a transport channel for the transport session. The client device then sets the streaming connection in a streaming state, ready to accept streaming media through the stream start procedure. The server device also moves into the streaming state, ready to stream the MP3 Audio. The stream manager component is used for streaming.

2.1.2 Audio Streaming Procedure

When a stream is established, the application layer in the server can send streamed media through the media transport channel. The reporting packets are sent both by the server and the client giving details on the quality of data distribution.

2.1.2.1 Media Transport (MP3 Audio)

The stream manager provides the functionality of streaming MP3 Audio using media framing, time stamp management, media packet sequence numbering, reporting of packet loss to peer and higher layer and jitter calculation. It provides the services of Media Transport and Media Reporting to the application layer.

The MP3 audio is streamed to the client encapsulated in Real Time Transport Protocol (RTP) [7] packets. The Reporting packets are transmitted as Real Time Control protocol (RTCP) [7]. The RTP/RTCP packet format used by the profile is based on RFC 1889. RTP uses a 12 byte Header and then the MP3 data is encapsulated in the RTP payload.

An important consideration here is that the server should send data at a constant rate over the Bluetooth link to avoid the audio decoder from running out of stream data. Lost data packets may also cause the decoder buffer underrun problem.

On the other hand, if data is sent very fast, then that data will be buffered up at the audio decoder, eventually causing congestion or data loss when the device runs out of buffer space. Since there is no flow control mechanism built into AVDTP or L2CAP, other mechanisms must be used to prevent data loss.

The mechanism used by the audio source, or the device sending the stream, depends on the type of source. If the source is "live" and an audio encoder provides audio stream data, then the encoder itself will provide the constant bit rate. If the source is from a file, then a timer must be used to maintain a constant bit rate.

In the Audio Streaming Profile the server application sends a MP3 stream from a file encoded at 128 kbit/sec and 48 kHz sample frequencies. The MP3 audio frame size is 384 bytes. The frame length of the audio frame is calculated by parsing the MP3 frame header [10]. Each RTP packet contains three MP3 packets plus the header information (12 bytes). Thus, the RTP packet size is 384*3 + 12 = 1164 bytes and the time interval between two packets sent on the server side is (384*3*8) / 128000 which is equal to 72 ms.

Since it is stored as an audio file the initial timestamp in the RTP packet header is chosen randomly and is incremented by the time interval calculated for each packet sent. To maintain the constant bit rate, the server simply sets a periodic timer for 72 msec and sends a packet at the timer expiry. If it is a live audio source, the timestamp corresponds to the sampling instant of the first octet in the RTP packet.

Some devices have problems using timers or processing data at very short intervals so the profile sends three MP3 frames in one RTP packet. This suggests that it is better to send a larger packet containing several frames at longer intervals rather than sending a small packet containing a single frame at very short interval. The maximum size of MP3 frames gives us an idea of how large to set the L2CAP MTU of the AVDTP transport channel such that audio frames do not need to be fragmented across AVDTP packets.

If the timer is not so accurate or if a packet arrives late, then audio decoder runs out of data resulting in a glitch on the receiver side. So, even a little inaccuracy can cause problems if the profile relies on every packet to be sent right on time.

The solution to this problem would be to give some slack in the data flow. Assuming the device receiving the stream can buffer up at least a few packets, the server can send a number of packets as fast as possible when streaming starts. This helps overcoming the timer inaccuracy and data delayed by lost packets as well. However, the number of packets that can be buffered depends on the implementation of the client device receiving the stream. A more elaborate approach is to send feedback to the streamer regarding the buffer status. While approaching the buffer full condition can slow down, whereas getting closer to an empty buffer can speed up the streaming server.

Following this approach, the client receives the RTP packet over the air, extracts the header information using the sequence number in the header, restores the packet sequence and buffers the MP3 data in a buffer before sending the data to the decoder. The MP3 frames are buffered for about 4.4 seconds (60 RTP packets nearly 180 MP3 frames) and part of the buffered data is sent to the decoder. By the time decoder plays the buffered data, the buffer would be occupied by the next set of frames.

2.1.2.2 Media Reporting

The RTP streamer and receivers both send the media reporting packets. RTP streamer and receivers provide reception quality feedback using report packets like RTCP Sender Report packets (SR) [7] or RTCP Receiver Report packets (RR) [7]. These packets should be sent periodically to minimize bandwidth constraints.

The server sends an RTCP sender report packet based on the RTCP transmission interval set by the sender. The RTCP transmission interval is calculated based on the number of participants present in the session the L2CAP MTU [9] of the AVDTP transport channel and the fraction of Bluetooth bandwidth allocated for the RTCP sender report. In the Generic Audio Video Distribution Profile there are two participants; client and server.

The RTCP sender report is sent by the streaming device. It contains information regarding the cumulative number of packets and the number of bytes sent. It also provides information about the inter-media synchronization (Audio/Video), which is not required in the profile developed.

The receiver can calculate the cumulative number of RTP packets lost from the server and the inter-arrival jitter, after receiving the sender report packet and prepare reception report packet. The inter-arrival jitter is a measure of the statistical variance of RTP data packets' inter-arrival time, measured in timestamp units and expressed as an unsigned integer.

For example, if Si is the RTP timestamp from packet i, and Ri is the time of arrival in RTP timestamp units for packet i, then for two packets i and j, D (delay or inter arrival jitter) may be expressed as

$$D(i,j) = (Rj-Ri) - (Sj-Si) = (Rj-Sj) - (Ri-Si)$$

Theoretically it would be equal to the RTP timestamp of the sender plus the time taken by the packet to travel in the Bluetooth environment. The RTP packet with 1164 bytes is sent as DH5 [1] and DH3 ([1] packets in the baseband layer. Three DH5 ([1] packets and one DH3 packet would be required to transmit the data.

Therefore, the minimum transmission time for each RTP packet is 0.625*(3+1) +0.625(5+1)*3 which equals 13.75 ms, where 0.625ms is the Bluetooth slot time, (each DH5 packet occupies 5 time slot in one direction and one in the opposite direction and DH3 packet occupies 3 time slot in once direction and one in opposite direction).

The inter arrival jitter value may be different from the value calculated, since the interference from other wireless devices operating in the same ISM band should be taken into consideration. The inter arrival jitter and other parameters in the RTCP Reception Report packet gives information to the sender about the reception quality feedback. The sender may modify its transmission based on the feedback. The RTCP report packet is used to maintain streaming statistics at the sender's end, comprising the jitter and packet loss rate.

2.1.2.3 Controlling the Streaming

The Audio profile sends the stream related control data over the signalling channel. When the client device invokes pause functionality, the profile sends a stream suspend command to the server and sets the server stream end point in a suspended state. The server pauses streaming. On invoking the resume functionality, the server device resumes streaming by performing the stream start procedure.

2.1.2.4 Stream Release

After the total file has been streamed to the client, the server application through the profile interface closes the stream.

On invoking stream close, the profile initiates stream release which closes the stream end point, releases the stream handle associated with the stream end point and releases the L2CAP channels used for AVDTP Transport and Reporting.

3. APPLICATION and SUPPORTING INFRASTRUCTURE

A project was established requiring the application of streaming MP3 to multiple users for an audio tour application. The project contained several elements, each running on different target hardware and operating system targets. A development environment as well as a Bluetooth protocol stack was required to facilitate the application development. For these elements Object Oriented SoftFrame and ClarinoxBlue both from Clarinox Technologies were used. This provided advantages of single application development on one platform and the ability to recompile and run with little change on the other required platforms plus inherent handling of adhoc networking and multiple simultaneous profiles.

Together SoftFrame and ClarinoxBlue provide an Application Interface (API) that distances the application developer from the intrinsic complexity of Bluetooth protocol specification. ClarinoxBlue encapsulates the vast number of operations, protocol layers and profiles within the Bluetooth architecture.

The application developer needs a thorough understanding of both the architecture of the Bluetooth specification and the ClarinoxBlue interface, to be able to articulately implement a solution with ClarinoxBlue, but does not require specific knowledge of the intricacies of the protocol, such as detailed knowledge of numerous PDUs and primitives. This approach minimizes the potential for error due to the reduction in the complexity of the task.

Figure 3 shows the overall view of the ClarinoxBlue API layer implementation. The infrastructure allows for the functionality of components GAP and SDAP to be inherited with each additional profile implementation. In the terminology used, the merging of an additional profile, with the infrastructure is termed a "Service". A Service is a control unit incorporating the functionality of GAP and SDAP into a miniature-contained application. Such a contained application may reflect proprietary requirements for a user or may be as defined by a Bluetooth profile.

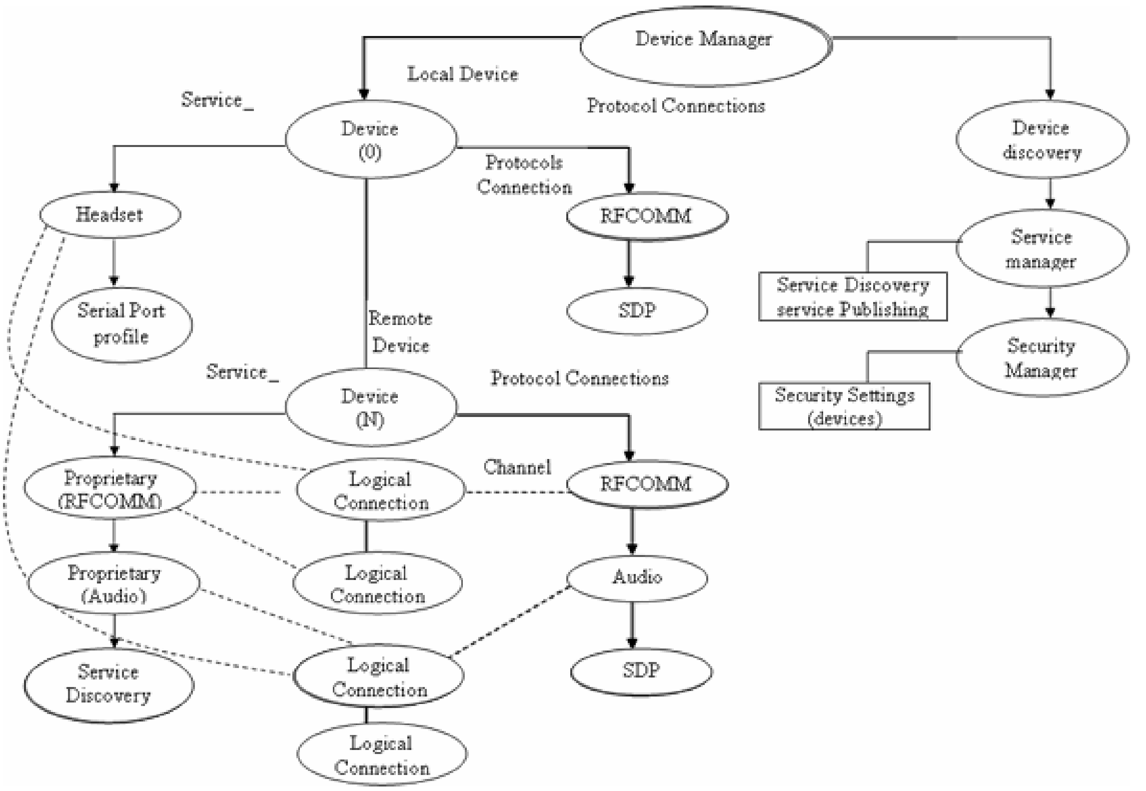

Figure 3: API Overview

The Device Manager handles the Bluetooth hardware module specific issues related to the local and remote Bluetooth devices. The application must pass parameters detailing its requirements to the Control Center on behalf of the Device Manager.

The local and the remote device abstractions are handled in the Device Manager so that the user does not need to know the physical details of devices. Complexity is lessened by the provision of a similar interface for both local and remote devices. The Local Device Manager contributes to the configuration of the transport layer to communicate with the Bluetooth Hardware module as well as the initialisation of the Bluetooth hardware module.

Device manager file contains the DeviceManager, DeviceGroup and the SdpTransaction classes. The device manager is responsible for the management of inquiry and service discovery procedures. The DeviceGroup is used to manage the grouping of the devices. The SdpTransaction class is used to maintain the SDP transaction handling.

The device manager controls or contains the information of the local device and the remote device. The device manager performs the device discovery. The LocalService class notifies the devices discovered to the device manager. The device class creates the proxy service in the device manager using the information retrieved during the device inquiry or discovery. Generic Audio Video Distribution Profile is implemented by inheriting it from the LocalService class.

The Service Discovery Protocol (SDP) transaction functions in the device manager class are used to retrieve the values of certain attributes of the specified service in the remote Bluetooth device. This information is used by the device class to create a copy of the specified service of the remote Bluetooth device in the local device manager. A logical connection is created between the local device and the created proxy service of the remote device in the local device manager.

4. CONCLUSION

This paper proposed a solution architecture for streaming audio (MP3 data) over Bluetooth. The proposed implementation only discusses point-to-point streaming. This was extended to support point-to multipoint streaming without any major changes in the architecture in the audio tour example application. ClarinoxBlue and SoftFrame from Clarinox Technologies provided suitable infrastructure and development environment to support the application development.

In this paper, use of RTCP packets in reception quality feedback has been discussed extensively. The RTCP data can be used for providing QoS control over Bluetooth wireless links. An optimal QoS strategy, namely Error Control, Congestion Control, adaptive packetization, enhances the robustness and efficiency of Audio transmission over Bluetooth links. Providing optimal QoS strategy for streaming Media over Bluetooth provides a vital area of research.

The approach discussed is not specific only to Bluetooth, but also can be used for streaming audio over other wireless technologies.

5. REFERENCES

[1] *Advanced Audio Distribution Profile*
[2] *Audio Video Distribution Transport protocol specification*
[3] Ling-Jyh Chen, Rohit Kapoor, Kevin Lee, M. Y. Sanadidi, Mario Gerla, *Audio Streaming Over Bluetooth, An Adaptive ARQ time out approach*
[4] MPEG Layer3 *Bit stream Syntax and decoding*, www.mp3tech.org
[5] MPEG Layer 3 *Frame Header Documentation,* www.mp3tech.org
[6] MP3 Play Lib, *mpg123 player* www.mp3tech.org
[7] RTP/RTCP RFC 1889 *www.ietf.org/rfc/rfc1889.txt*
[8] Staffan Gadd & Thomas Lenart, *A Hardware Accelerated MP3 Decoder with Bluetooth Streaming Capabilities*
[9] *Specification of Bluetooth System Core* Vol 1.2 www.bluetooth.org
[10] Wang Xiaohang, *Video Streaming over Bluetooth a survey*
[11] Singh, Mritunjay, *MP3 Audio streaming Over Bluetooth*
[12] Messiter, Trish, *The place of Wireless, Radio Communications, 2004*