BlueBox: A cable-free Digital Jukebox for compressed-quality Audio Delivery

Andreas Floros, Nicolas – Alexander Tatlas and John Mourjopoulos

Abstract — *The recent growth of home networking applications has raised the requirement for cable-free distribution of an increasing amount of media to portable playback devices, which lack the resources for local storage. In this work, the "BlueBox" wireless digital jukebox application is presented, which employs the Bluetooth wireless technology for distributing on demand, high-quality compressed audio, originally stored in a Bluetooth-enabled digital jukebox to remote portable client devices. It is shown that despite the strict throughput limitations imposed by the early-defined Bluetooth protocols, the BlueBox application setup can efficiently accommodate cable-free, high-quality audio distribution over an adequate number of receivers for home environments[1].*

Index Terms — Wireless audio, real-time playback, Bluetooth player.

I. INTRODUCTION

Compressed-quality audio such as MPEG1-Layer 3 (mp3) is an extremely popular digital audio format today [1], with a vast number of audio tracks available for download over the internet, through subscriber services or even peer-to-peer networks. In spite of the availability of solid-state memory and hard drive-equipped compressed-audio portable players, storage of mp3 files is still intrinsically tied to the Personal Computer (PC). Most portable audio players come equipped with a wired interface, such as the Universal Serial Bus (USB) or IEEE1394, allowing the direct transfer of files from a PC. While home networking applications are constantly evolving over the last years, the possibility of streaming directly the audio content to a portable player for immediate playback over a network is generally under research. A number of system prototypes have been designed lately offering streaming capabilities over typical networking protocols, however wireless audio streaming to portable devices is not yet exploited.

On the other hand, wireless networks are becoming a necessity these days, meeting the need for a cable-free but networked environment. Several Wireless Local Area Networks (WLANs) protocols have been standardized and are broadly employed, such as the IEEE802.11b/g [2] operating in the 2.4GHz band, providing hi-speed communication with data rates up to 54Mbps. WLANs are able to offer wideband wireless connectivity between Desktop PCs, Laptops, Personal Digital Assistants (PDAs), and other portable electronic devices for voice, video and data applications.

In the area of Personal Area Networks (PANs) where the low power consumption and the reduced complexity and cost are prime objectives for the design and development of portable, self-powered devices, Bluetooth [3] represents a widely-adopted wireless technology operating mainly in home, office and automotive environments. Although this technology was initially developed for replacing cables, it evolved into a way to create small radio LANs.

The Bluetooth specification defines synchronous connection- oriented mechanisms for servicing time-critical applications (for example voice and audio) at a reserved constant rate equal to 64kbps. However, this limited bandwidth performance represents a major drawback even for compressed-quality digital audio delivery. Hence, packet-oriented (or asynchronous) connections can alternatively be employed, provided that the physical layer offers sufficient bandwidth (at least equal to the maximum encoding bit rate) [4].

This work presents the development and employment of a set of client-server applications that facilitate wireless digital audio streaming to portable devices from a server PC using the Bluetooth v1.2 protocol specification [5]. Section 1 outlines the Bluetooth protocol basic operations and features. In Section 2 the BlueBox jukebox server and client applications are presented in detail. In Section 3 the employment of these applications for a typical audio delivery setup implementation is described, leading to the measured results Section 4. Finally, the conclusions are summarized in Section 5.

II. BLUETOOTH OVERVIEW

Bluetooth-enabled devices employ an embedded radio module operating in the ISM (Industrial Scientific Medicine) frequency band in the range of 2.4GHz. In order to operate in the presence of electromagnetic interference and fading, a frequency-hopping algorithm is employed which theoretically achieves a total bandwidth equal to 1Mbps within a typical distance of 10 meters. This range can be extended up to 100 meters for Bluetooth class 1 devices, which incorporate RF modules that provide larger transmission power, equal to 20dBm.

A Bluetooth piconet typically consists of a master device (or node), which is connected with a number (maximum seven) of slave devices (Fig. 1). As Bluetooth was initially designed to service both data and real-time traffic transmissions, it supports both asynchronous and isochronous services through two different types of links: (a) point-to-point or point-to-multipoint Asynchronous Connection-Less (ACL) links and (b) Synchronous Connection Oriented (SCO) point-to-point links. When using an ACL link, a “slotted” channel is applied with a slot time equal to 625μseconds, while a Time-Division Duplex (TDD) scheme is employed for full duplex transmissions. A packet transmission can cover up to 5 channel slots. The packet-oriented transmissions through the ACL links can achieve a maximum of 721kbps effective bitrate and can be acknowledged through a retransmission mechanism. However, in the presence of excessive channel interference, packet retransmissions significantly degrade the above throughput performance.

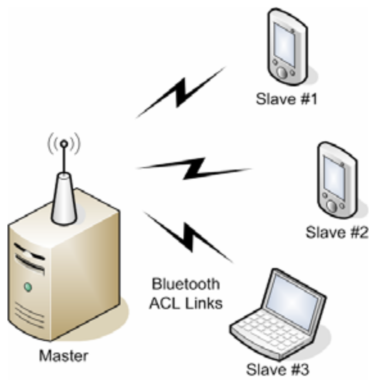

Fig. 1: Typical Bluetooth piconet setup

ACL connections support two packet types: Data-Medium rate (DM) and Data-High rate (DH) packets, with the former additionally containing Forward Error Correction (FEC) information. The Bluetooth specification defines three types of DM packets (named as DM1, DM3 and DM5). A DM1 packet length is limited to a single channel slot, while DM3 and DM5 packets are extended to 3 and 5 channel slots respectively. DH1, DH3 and DH5 packet lengths are also defined accordingly. ACL links can accommodate both point-to-point and point-to-multipoint (or broadcast) packet transmissions.

As it is shown in Fig. 2, the Bluetooth specification defines the Bluetooth protocol stack in layers, in common with all the well established wired and wireless networking protocols. More specifically, the Bluetooth lower layer stack consists of (a) the 2.4GHz radio (RF) layer that implement the physical transmission and reception interface (b) The baseband or link control layer that realizes the timing, ordering and sequencing of the transmitted bits, including channel coding and (c) The link manager layer, which manages the behavior of the wireless link and implements service discovery procedures between Bluetooth-enabled devices, that are in range. The upper part of the link manager co-operates with the Host Controller Interface (HCI) and the Logical Link Control and Adaptation Layer (L2CAP) for providing higher layer protocol services, such as data multiplexing and packet segmentation and reassembly. Both the link manager and HCI layers are usually implemented as software modules, but often can be found as embedded firmware, to assure lower power consumption and implementation complexity.

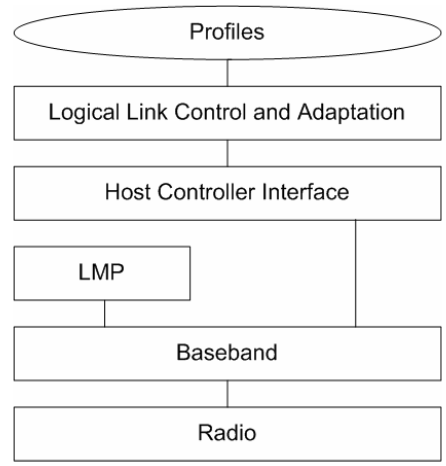

Fig. 2: Block diagram of the Bluetooth layer stack

The Bluetooth specification also introduces the concept of profiles. A profile is a selection of procedures and messages that extend the basic application level Bluetooth functionality and assure interoperability between the remote services. For audio-oriented applications, several profiles have been already defined (such as the Generic Audio/Video Distribution Profile – GAVDP [6] and the Advanced Audio Distribution Profile – A2DP [7]), which define procedures for establishing audio/video streaming applications and basic compressed quality transmission over Bluetooth.

A previous work [8] has defined the basic framework for transmitting compressed quality audio over Bluetooth using both point-to-point and point-to-multipoint transmissions through ACL links. This work has also assessed the overall performance achieved during the wireless audio transmissions as a function of the Bluetooth protocol parameters (e.g. the packet length and type, etc). The transmission optimizations derived from this work were employed here in order to efficiently develop the BlueBox wireless digital jukebox application, which is described in the next Section.

III. BlueBox Application overview

The BlueBox jukebox architecture (see Fig. 3) is based on the well-established server-client model. An application running on the server side (i.e. the BlueBox server) is responsible for handling the stored media lists and for establishing wireless Bluetooth connections with the remote client devices. These devices are controlled by the BlueBox client application, which realizes the interface for all the basic user operations, such as track selection and playback control, connection and disconnection from the server etc. A detailed description of both applications is provided in the following paragraphs.

A. The BlueBox Server application architecture

The BlueBox server application is typically hosted on a Bluetooth-enabled PC in master mode, with sufficient disc space for storing audio media. A special, user friendly procedure was designed and developed for easily organizing the audio content locally stored on the jukebox server side in play-lists. Provided that a Bluetooth connection is successfully established between the server and a remote client device and upon a user request, a specific, user-selected track is transmitted to the BlueBox client application and is reproduced in real-time.

Due to the bandwidth limitations of the Bluetooth protocol, only compressed quality audio can be serviced. Currently, the MPEG1 Layer III (mp3) and the Windows Media Audio (WMA) formats are supported with data rates up to 320kbps in mono, stereo and joint stereo modes. Compact Disc quality media can be also employed (2 audio channels with bit resolution equal to 16bit and sampling frequency 44.1kHz), but in this case, the digital audio data are compressed in real-time using the MPEG1 Layer III standard at 128kbps, prior to the wireless transmission.

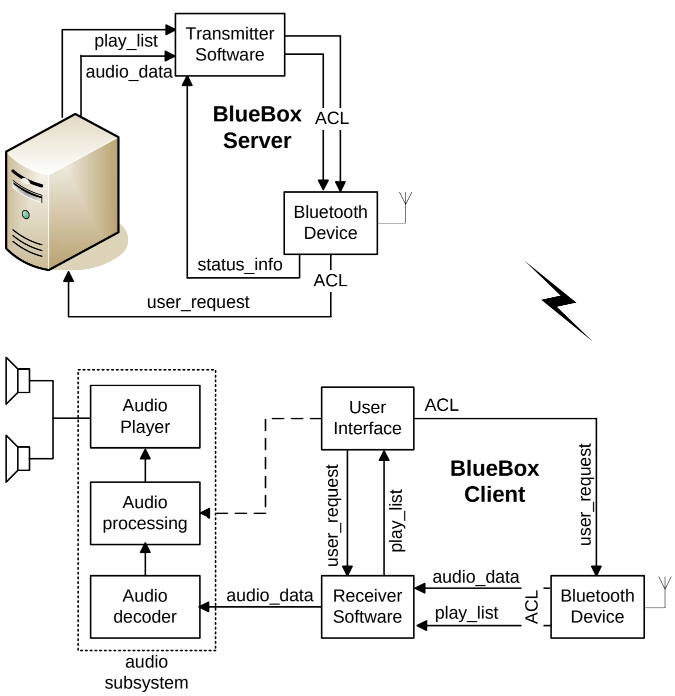

Fig. 3: Architecture of the BlueBox application

As it is shown in Fig. 3, after the successful ACL connection establishment with a BlueBox client device, the BlueBox server automatically transmits the available music track titles (play_list data) contained in the user defined play list through the wireless link. All Bluetooth transmissions are performed using DH5 packets, unless otherwise stated. The audio data (audio_data) that correspond to a specific audio track contained in the play-list are also transmitted through the same link upon a remote user request. As it will be explained in the next Section, these requests are initiated on the BlueBox client side and transmitted through the bi-directional ACL link (user_request traffic flow) using a custom-defined protocol developed in this work. This protocol also defines the necessary signaling for supporting a complete set of playback control functions, such as stop, pause etc, as well as the procedures for distributing the available play-list and it's updates to all the connected devices.

The BlueBox server must be able to service more than one remote clients simultaneously. For this reason, data transmissions are controlled by a transmitter software module, which is responsible for coordinating the transmissions above the Bluetooth stack layer. For example, each time a playback initiation request is received from a client for a specific track, this module initiates a thread that performs the audio data delivery by sending the data packets to the Bluetooth device.

The transmitter software module controls the execution of all the active transmission threads and synchronizes them in order to meet the necessary bandwidth requirements for the audio data transmissions, taking into account packets transmitted periodically by the clients (status_info traffic flow) containing information about the playback status. More specifically, in order to avoid any data overflows on the remote client reception buffers, if the bandwidth allocated for a specific audio stream exceeds the value required for error-free, uncorrupted playback, the thread execution is blocked. On the other hand, if many remote clients request service and/or under heavy channel interference, the data delivery to one or more clients will be delayed and the playback (reception) buffer will get empty, causing audible playback interrupts.

As concurrent transmissions and/or reception may occur for the above transmitted data types (audio_data, status_info and user_request), the Bluetooth L2CAP layer is employed for creating logical data channels and efficiently multiplexing the transmitted data flows through the same bi-directional ACL link. It should be noted that, although the choice of the L2CAP layer reduces the overall BlueBox application complexity (as all the required multiplexing functionality is realized in the Bluetooth lower stack), as it was concluded in [8], it's presence in the transmission chain may significantly affect the effective bitrate measured on the application level. Hence the maximum number of the remote clients that can be concurrently serviced is somehow decreased.

B. The BlueBox client application architecture

As it is illustrated in Fig. 3, the BlueBox client application consists of three basic modules: (a) The interface with the wireless sub-system (Receiver Software) that is responsible for receiving data packets from the Bluetooth transceiver and for sending playback control messages (status_info) back to the server (b) the User Interface that locally stores the received play-list and initiates any user requests following the user actions and options and (c) the audio sub-system which decodes the received compressed audio packets, processes the decompressed audio data and drives them to the audio playback sub-system (e.g. the PC soundcard).

More specifically, upon the successful establishment of a Bluetooth connection with the BlueBox server, the client application receives the play-list data. The information contained in this play-list is used for mapping the track selected by the user to an original track identifier (TrackID) stored on the BlueBox server side. This mapping is performed by the custom protocol running on the application layer, which is also responsible for all the signaling and control information exchange between the server and the remote clients. Upon a track selection, an appropriate message is transmitted to the server, which contains the unique TrackID that corresponds to the specific track. After the successful reception of this message, the BlueBox server starts transmitting the compressed audio data to the requesting client.

In order to minimize the effects of the variable channel conditioning (especially in the presence of strong electromagnetic interference which would increase the number of retransmissions required for safe wireless delivery), audio data pre-buffering is performed. The pre-buffering length id equal to 2 seconds, meaning that the playback will start 2 seconds after the first audio data packet reception by the client. Provided that the highest audio data rate equals to 320kbps, the minimum length of the memory required for pre-buffering purposes equals to 80kBytes. In this work, and for safety reasons, the length of the receiver buffer was equal to 100kBytes.

IV. BLUEBOX APPLICATION IMPLEMENTATION

The Jukebox client-server applications were developed using the C++ programming language. The BlueBox server can run on any Bluetooth-enabled PC having sufficient hard disk space for audio storage, while two versions of the BlueBox Player were implemented: one for desktop PCs and one for PDAs and Pocket PC devices. Both the PC and Pocket PC versions were designed to provide exactly the same functionality described here.

In order to setup a test topology, a typical office PC was employed as the Jukebox server, while a typical notebook and a PDA were used as BlueBox clients. The Bluetooth adapters used were based on a proprietary single chip solution baseband controller [9] with PCMCIA physical interface, connected with Class 3/0 dBm RF modules.

Fig. 4 shows the main window of the BlueBox server running on the Master PC host. Using this dialog, the user is able to conveniently select folders were media files reside through the “Configure” submenu; upon a folder selection, all compatible audio files contained into the folder are added to the server play-list, while a unique TrackID is assigned to each one. As mentioned previously, the supported audio formats are mp3, WMA and CD-audio (through real-time encoding to 128kbps mp3 prior to transmission).

Provided that the play-list contains at least one supported audio file, the BlueBox server becomes ready to accept connections from BlueBox players. The addresses of the connected players, the number of active transmissions, the number of available audio tracks as well as a connection monitor is included in the BlueBox server main dialog. Currently, the server allows up to three concurrent active transmissions. However it is obvious that if the total requested bandwidth from the remote players exceeds the total available, buffer under-run errors will occur to at least one player, causing audible distortion in the reproduction.

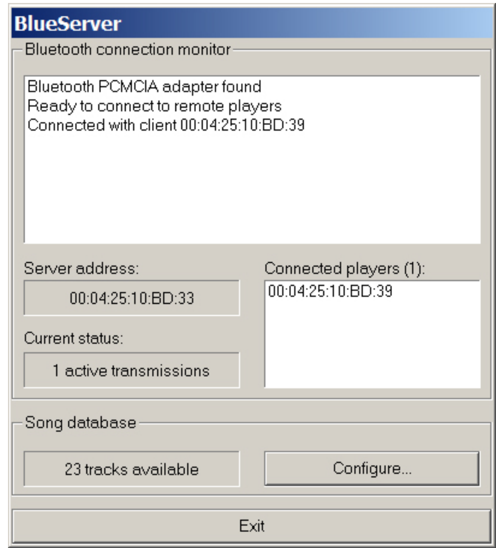

Fig 4.: BlueBox server main window

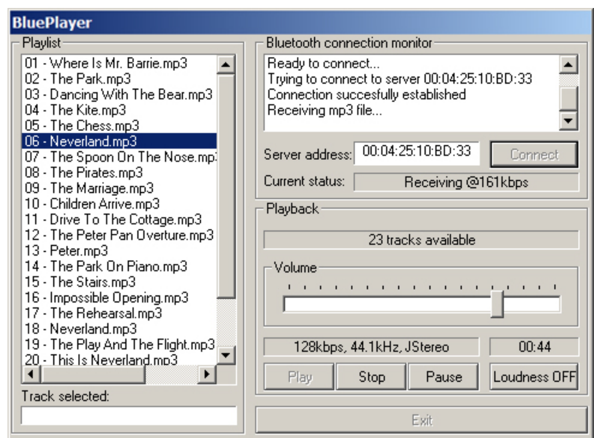

Fig 5.: BlueBox Player main window running on a desktop PC

Fig. 5 shows the main window of the BlueBox player running on a desktop PC Slave host. The application facilitates the connection to the BlueBox server by simply entering its address. Once connected, the play-list is downloaded from the server and shown in the UI. The user can select the desirable audio track for reproduction between the available sound files. While in reproduction mode, the connection bitrate is shown in the status field, as well as the audio bitrate, sampling frequency and Mono/Stereo or Joint Stereo encoding and the playback position in minute:second format. The user can pause or stop the reproduction at any time using the appropriate controls. Finally, basic digital signal processing features are available, such as volume control and filtering with the loudness curve.

V. BLUEBOX APPLICATION RESULTS

During the BlueBox testing, the effective (application level) bitrate was measured, since it implicitly characterizes the quality of real-time audio playback. The measurement was performed for one active client, in a closed room with no additional EMI existing in the range of 2.4GHz. The transmissions were performed using mp3 audio material, with the size of each track obviously depending on the selected compression rate. All ACL Bluetooth packet types were considered, by modifying the application settings. In order to measure the maximum capacity of the Bluetooth link for one client, the flow control mechanism employed in the server side was disabled.

The test parameters values used were defined as follows:

(a) The distance between the server and the active client, from 1m to 12m

(b) The ACL packet type employed for each connection, namely DM1, DM3, DM5 and DH1, DH3, DH5 and finally,

(c) The audio packet length forwarded to the Bluetooth layer stack, ranging from 1kByte to 55kBytes

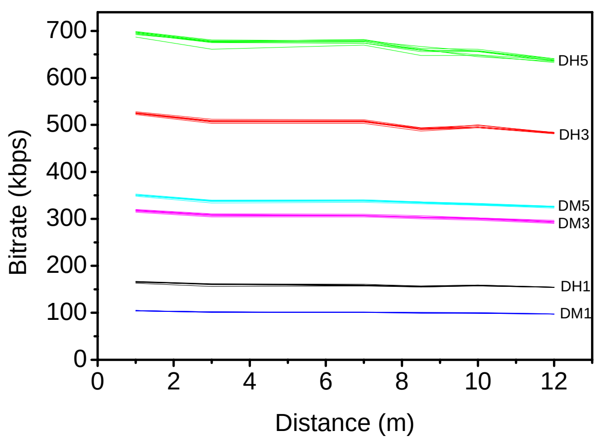

Fig 6.: Variation of the measured bitrate with ACL and application packet length, and distance

Fig. 6 shows the measured effective bitrate as a function of distance for one active connected client for all possible ACL and application packets. From this figure, the following conclusions can be drawn :

(a) A strong dependence of the measured application-level bitrate on the selected ACL packet is observed. For all cases of application packet length L_p (appearing in Fig.6 as a group of curves under the ACL packet name) the choice of DH5 ACL packets achieves the highest bitrate values, close to the theoretical maximum defined by the Bluetooth specification. On the contrary, the employment of other packet types significantly decreases the measured bitrate. This is in good agreement with results obtained in past studies that assess the effects of the ACL packet types on the measured channel bitrate [10].

(b) The effective bitrate does not depend gravely on the distance. High-bitrate coded audio transmission is possible even for distances of nearly 12m (in line-of-sight).

(c) The measured bitrate is not affected by the variation of the application packet length in a systematic way.

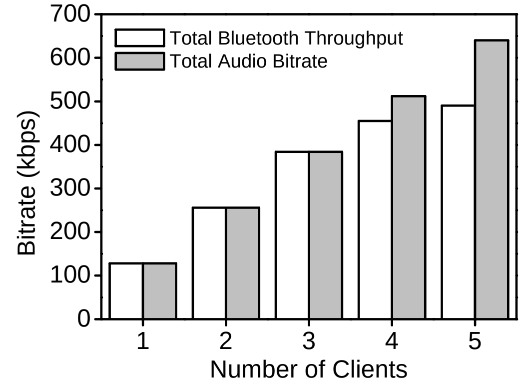

Fig 7.: Total Audio and Bluetooth bitrate for increasing number of clients

Fig. 7 shows typical (necessary) audio and the corresponding measured bitrates for different number of clients. For both cases, the calculation is performed using the summation of the required and measured bitrates for each client. The transmissions were performed in all test cases using DH5 application packets as they were found to provide the maximum bandwidth performance, in distances up to 10m using mp3 audio material encoded at 128kbps. The clients employed for the tests were both notebook PCs and PDAs. Ideally, uncorrupted real-time streaming requires a minimum offered bandwidth value at least equal to the audio data bitrate. In this sense, it is clear that the BlueBox application is able to concurrently serve up to three active clients, for real-time streaming of 128 Kbit coded audio material.

However, for the case of more than three clients, the Bluetooth throughput is less than required, leading to audible gaps in playback for at least one client device. This does not agree with the previous measurements, since approx. 700kbps maximum throughput is expected, while less than 500kbps is achieved for the case of four active clients. This throughput degradation may be either due to overheads induced by the custom upper layer transmission protocol and the multiple transmissions in the server-side, or caused by the multiple Bluetooth-enabled devices transmitting within each other range, increasing the probability for excessive number of retransmissions.

Fig. 8 shows the measured effective bit rate compared with the total audio bitrate, indicating the maximum allowed compressed audio bitrate in the case of *two* remote clients. The real-time threshold line graphically represents the cases where the effective bitrate is sufficient to service both clients in real-time. For example, for concurrent real-time playback of two 128Kbps-encoded mp3 files, employment of at least DM3 packets is necessary.

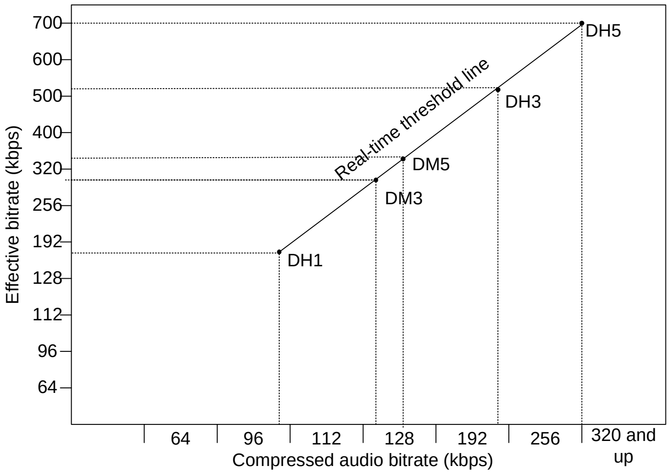

Fig 8.: Diagram of the audio data bitrate and the maximum bitrate allowed for 2 concurrent active players, for different ACL packet types

VI. Conclusions

In this work, the architecture and implementation of the BlueBox cable-free digital jukebox application is presented. Based on the Bluetooth protocol, this application is targeted mainly for home environments and medium-sized enclosures. Following the client-server architecture, the BlueBox server and client applications were developed, and a custom application layer protocol was realized which implements the signaling required for efficient audio data and control information exchange between the media server and all the connected remote clients. This protocol is running on the top of the Bluetooth layer stack, taking advantage of the L2CAP layer functionality for realizing low-complexity traffic multiplexing and packet transmissions.

Using the BlueBox server and client applications it was found that, despite the limited bandwidth offered by the Bluetooth protocol and using mp3 and WMA audio streams coded at 128kbps, a maximum of three remote audio clients can be concurrently serviced in a typical home environment. CD-quality audio is also serviced, but it is compressed to 128kbps mp3 streams prior the wireless transmission. Basic DSP procedures (such as audio data filtering using typical treble-bass controls, volume control etc) are also implemented on the client side. However, it is in the authors' future plans to add similar signal processing functionality on the server side. This would significantly decrease the BlueBox client implementation complexity and allow its development on non-PC based, hardware platforms.

Taking into account the enhancements included in the recently published Bluetooth protocol specification v.2.0 [11], which significantly increase the theoretical Bluetooth throughput performance up to 3Mbps, the authors intend to employ this standard for the BlueBox application. It is expected that this would enhance the overall application performance (mainly in terms of the total number of clients that can be serviced), as well as it would enable the development of advanced interactive features and improved functionality.

References

[1] K. Brandenburg, "MP3 and AAC explained", presented at the Audio Engineering Society 17th International Conference, August 1999

[2] IEEE802.11 WG, IEEE802.11g, "IEEE Standard for Information technology-Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band", June 2003.

[3] Bluetooth SIG, Specification of the Bluetooth System, Core Package version 1.2

[4] Staffan Gadd and Thomas Lenart, "A Hardware Accelerated MP3 Decoder with Bluetooth Streaming Capabilities", MSc Thesis, November 2001

[5] "Bluetooth and Wireless Networking", Journal of the Audio Engineering Society, Vol. 50, No. 11, November 2002.

[6] Bluetooth SIG, Generic Audio/Video Distribution Profile, version 1.0, adopted

[7] Bluetooth SIG, Advanced Audio Distribution Profile, version 1.0, adopted

[8] A. Floros, M. Koutroubas, N. A. Tatlas and J. Mourjopoulos,"A Study of Wireless Compressed Digital-Audio Transmission", presented at the Audio Engineering Society 112th Convention, Munich, May 2002

[9] ATMEL AT76C551 Single Chip Bluetooth Controller Datasheet, Aug. 2001.

[10] Mark Rison and Like D'Arcy, "…OK?", Incisor Bluetooth Newsletter, Issue 34, August 2001

[11] Bluetooth SIG, Specification of the Bluetooth System, Bluetooth core specification version 2.0 + EDR [vol 0], November 2004.