

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/283113151>

L2CAP-based prototype media streaming via Bluetooth technology

Article in *International Journal of Networking and Virtual Organisations* · January 2014

DOI: 10.1504/IJNVO.2014.065785

CITATIONS

3

READS

742

5 authors, including:



Ebaa Fayyumi

Hashemite University

34 PUBLICATIONS 284 CITATIONS

[SEE PROFILE](#)



Sahar Idwan

Hashemite University

51 PUBLICATIONS 342 CITATIONS

[SEE PROFILE](#)



Hiba Muhareb

Princess Sumaya University for Technology

6 PUBLICATIONS 24 CITATIONS

[SEE PROFILE](#)



Izzeddin Matar

Petra University

12 PUBLICATIONS 26 CITATIONS

[SEE PROFILE](#)

L2CAP-based prototype media streaming via Bluetooth technology

Ebaa Fayyoubi* and Sahar Idwan

Computer Science Department,
Hashemite University,
P.O. Box 150459, 13115 Zarqa, Jordan
Email: enfayyoubi@hu.edu.jo
Email: sahar@hu.edu.jo
*Corresponding author

Hiba A. Muhared

Faculty of Information Technology,
Princess Sumaya University of Technology,
P.O. Box 1438, 11941 Amman, Jordan
Email: hiba.m99@gmail.com

Izzeddin Matar

Software Engineering Department,
University of Petra,
P.O. Box 961343, 11196 Zarqa, Jordan
Email: imatar@uop.edu.jo

Obaidah A. Rawashdeh

Faculty of Information Technology,
Applied Science University,
P.O. Box 166, 11931 Amman, Jordan
Email: obirawa@gmail.com

Abstract: A new model is required for wireless Bluetooth technology that enhances media streaming between different devices. This paper presents a novel system that utilises the intermediate protocol L2CAP and a newly proposed integrated buffering architecture to stream media file *via* Bluetooth technology that is called 'mBlueThunder'. The new system overcomes the jittering playback problem, time delay and compatibility issues. The experimental results show a remarkable improvement in streaming file reaching up to 35% comparable to the state-of-the-art technique. The overall results highlight great impact, significance and applicability of the proposed system.

Keywords: mBlueThunder; Bluetooth technology; buffering system; logical link control applications protocol; L2CAP.

Reference to this paper should be made as follows: Fayyoubi, E., Idwan, S. Muhared, H.A., Matar, I. and Rawashdeh, O.A. (2014) 'L2CAP-based prototype media streaming via Bluetooth technology', *Int. J. Networking and Virtual Organisations*, Vol. 14, No. 3, pp.221–238.

Biographical notes: Ebba Fayyoubi received his BSc degree from The Hashemite University, Jordan in 2000, MSc degree from the University of Jordan, Jordan in 2002, and PhD degree from Carleton University, Canada, in 2008. She has been with the Faculty of Prince Hussein Bin Abdalla II for Information Technology, The Hashemite University, since November 2008, as an Assistant Professor. Her current research interests include statistical syntactical pattern recognition, micro-aggregation techniques, secure statistical databases, automata learning, mobile applications, e-learning and applied algorithms. She received many awards during her academic life; one of them is Carleton University Medal on outstanding graduate work in 2008.

Sahar Idwan is an Associate Professor in the Computer Science Department at Faculty of Prince Al-Hussein Bin Abdullah II for Information Technology at Hashemite University. She received her PhD in Computer Science from Colorado School of Mines, USA in 2005 and joined Computer Science Faculty at Hashemite University immediately. She has 12 publications resulting from her research in computer science. In particular, her research interest includes applied algorithms, moving objects, mobile computing, graph theory, mobile applications, and application of computing as a scientific paradigms; her work appeared in different publications.

Hiba A. Muhared is an MSc student in the Computer Science in Princess Sumaya University for Technology. She received her Bachelor degree from Hashemite University, Zarqa in 2012. Her research interest include Bluetooth technology and concurrency control in database.

Izzeddin Matar is a Lecturer in the Software Engineering Department at Faculty of Information Technology at University of Petra. He received his MSc in Software Engineering from Colorado Technical University, USA in 2004. In particular, his research interest includes software engineering, Bluetooth technology, and application of computing as a scientific paradigm.

Obaidah A. Rawashdeh is an MSc student in the Computer Science in Applied Science University. He received his Bachelor degree from Princess Sumaya University for Technology, Amman in 2012. His research interest include Bluetooth technology and concurrency control in database.

1 Introduction

Bluetooth is considered as a wireless protocol that enables devices to communicate without manual composition or wired framework and support low power consumption and short range communication (Butt and Abbas, 2008; Heidari, 2008; Shepherd, 2001). Data transfer through the Bluetooth is based on master-slave mechanism. This data can be documents, images and sounds. The master Bluetooth device communicates with a maximum of seven slave devices. This group is called a piconet where at any specific time the master communicates directly with one slave (Morrow, 2002). The round-robin approach is used to switch quickly from one slave to another (Holtmann, 2008). Recently,

linking several piconets together has produced a global wireless ad hoc network to enhance the throughput and reduce packet delays (Reddy et al., 2010).

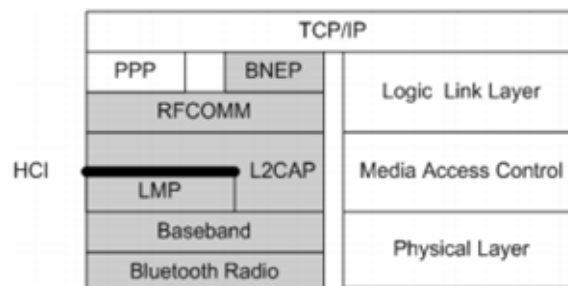
The Bluetooth system consists of many existing protocols that are directly used for different levels of communication (Harte, 2008):

- low-level protocols are only used to create, manage and disconnect transmission between specific points such as protocols that are used to manage a radio link between specific points
- mid-level protocols are used to create, manage and disconnect a logical connection between endpoints that may have multiple link connections such as transmission control protocols
- high level protocols are used to launch, control and close end-user applications such as application layer protocols.

Bluetooth uses 2.4 GHz ISM band that is divided into 79 channels (1 MHz each). The used frequency hopping spread spectrum (FHSS) jumps from one channel to another with a pseudo-random sequence. Devices that communicate with each other share this channel and jump together with a hop rate equals 1,600 hops per second, called a slot. Many piconets form a scatternet which can mainly support three kinds of connections in an ad-hoc network: peer to peer, master-slave and many to many (Bluetooth SIG, 2002). The peer to peer connection allows two devices to connect with each other directly, while master-slave connection allows one device (master) to connect to another device (slave). The most complicated connection is many to many where slaves and masters multiplex through time division and participate with each other.

Some of the layers and protocols associated with the Bluetooth technology including the baseband layer (physical layer), link management protocol, host controller interface (HCI), logical link control applications protocol (L2CAP), RF communications protocol (RFCOMM), object exchange (OBEX) and service discovery are presented in Figure 1. Further information can be found in Fayyumi et al. (2014), Gratton (2003), Huang and Rudolph (2007), Idwan et al. (2008, 2013), Idwan (2009), Kammer et al. (2002) and Kapoor et al. (2003).

Figure 1 The Bluetooth wireless protocol stack includes a physical layer, link layer and application layer



The great development in the information communication technology facilitates wireless mobile application due to its flexibility and applicability, which solves many problems to make things possible in various application-service-domains such as:

- *Healthcare*: the Bluetooth wireless technology has also been industrialised to support people in telemedicine applications such as designing portable ambulatory ECG monitoring systems (Hu et al., 2009), measuring the body temperature of kids using Fever-Watcher Kit (Fayyoubi et al., 2014) and providing people of motor disability with a handset device that has the ability to interface with large range of input devices (Kim et al., 2011). It is worth mentioning that the usage of telemedicine has recently been increased by combining the network-centric of healthcare (Lubitz et al., 2006), designing various frameworks for a wireless healthcare monitoring system with ZigBee communication link (Hamza, et al., 2011) and developing a low-cost wireless telemedicine network in rural area (Zatari, 2007).
- *Business*: recent studies (Lindner et al., 2010; Nilsson et al., 2011; Venkatraman and Yoong, 2009) have presented the role of emerging mobile technology on various application areas such as creating a dynamic business model that aimed to make money and created custom value (Lindner et al., 2010), utilising the e-business by proposing a high level web service oriented architecture, balancing the required cost of business construction and increasing the productivity that have a competitive edge in terms of quality of service and customer satisfaction (Venkatraman and Yoong, 2009). Developing a framework endorses the multi-criteria decision-making model with a mobile payment venture from business view and customer view (Nilsson et al., 2011). The latter frame also enhances the delivery mechanism for the current multimedia messaging service of the m-commerce applications *via* mobile phone.
- *E-learning*: the usage of the Bluetooth wireless technology enhances the e-learning domain in different aspects such as developing a classroom tool, which helps lecturers to automate their assignment tests and to enhance student portal in universities by connecting students' mobile phone devices to a server application (Idwan, 2009). Another aspect appears in utilising the social media environment by investigating the potential of Facebook-enabled mobiles to leverage learning in informal learning environments (Rambe, 2009).

Several studies have reported simulating media streaming over Bluetooth (Selvaradjou et al., 2010). Some of them worked on the packet size (Chong and Beg, 2003), data rate (Peterson et al., 2003) and network connection or enhancing the quality of service (Scheiter et al., 2003). Other studies investigated the feasibility of several protocols such as RFCOMM and L2CAP (Bluetooth SIG, 2002). However, the need to build an implemented infrastructure to conduct media streaming applications without jittering playback problem, time delay and compatibility issues by using mobile phones is of high concern to many service-domain applications as mentioned earlier (Razavi et al., 2013; Smith, 2012). In this paper, the proposed *mBlueThunder* is a novel system that utilises the L2CAP and a newly proposed integrated buffering architecture to stream a media file to overcome the mentioned problems.

The skeleton of this paper is organised as follows: Section 2 describes the state-of-the-art of Bluetooth technology and media streaming. Section 3 presents the novel system of media streaming called methodology of the proposed model called *mBlueThunder*. Section 4 discusses experimental results of the whole system and its implications and practices. Finally, conclusions is drawn in Section 5.

2 Literature survey

During the last years, Bluetooth community worked hard to establish a solution to replace the wired communication media with the hierarchical architecture of Bluetooth wireless communication in order to increase efficiency and speed of transformation (YuHai, 2008). Other researchers were focused on proposing different solutions to enhance either the audio streaming, compatibility issues or connectivity of delivering different kinds of files with minimum time *via* Bluetooth wireless technology. Exploring a competitive approach to enhance a quality of internet-radio improved the advertising mechanism *via* the internet (Gartley and Smith, 2006; Idwan et al., 2008). This solution was based on different standards for headfast and headphones; also supported remote control and metadata information such as song titles.

TCP has widely been used by commercial video streaming systems. The guideline of using TCP in streaming can be found in Wang et al. (2003). The problem of determining the right size of consumer buffer in memory-constrained applications over TCP has been solved in Kim and Ammar (2006). Their analysis shows that the consumer buffer requirement is determined by the network characteristics and desired buffer under-run probability. The study of Nguyen and Cheung (2008) proposed and implemented consumer-driven, TCP-based system multi-TCP for multimedia streaming over the internet using multiple TCP connections for the same applications. Their proposed system provided resilience against short-term insufficient bandwidth and enabled the application to control the sending rate in a congested scenario. The proposed system was implemented at the application layer without any kernel modifications.

Bluetooth wireless technology is a hot research topic (Amrit and Singh, 2003; Chong and Beg, 2003). Unfortunately, It is not suitable for traditional media encoding and real time transition (Kassem et al., 2009) due to limited bandwidth (Smith, 2012; Tariq et al., 2000), high degree of error rates (Peterson et al., 2003; Smith, 2012) and the time varying nature of radio link (Selvaradjou et al., 2010).

There are many important applications require media streaming over Bluetooth that have been developed and enhanced over the time. The quality of services has been enhanced by the integration of MPEG-4 codec video transmission over Bluetooth for mobile applications (Scheiter et al., 2003). MPEG-4 is an ideal compression technique that meets the requirements of the Bluetooth data bandwidth by providing a very low bit-rates and high compression (Smith, 2012). A link layer approach improves the quality of MPEG streaming over a wireless link by reducing time varying nature that causes video frames to be dropped/delayed (Kapoor et al., 2003). While others suggested a solution to solve net connection for video applications in built-in cell phone camera *via* overlaid Bluetooth piconets and its simplified version (Jung et al., 2006). For that reason, Razavi et al. (2006) proposed an automatic repeat reQuist (ARQ) that is sensitive to mobile device buffer size and combined it with an active discard policy based on picture type. It is worth mentioning that the buffer size has a huge impact on the quality of the video streaming over Bluetooth. Therefore, high quality of video streaming is achievable by dynamically packed bit stream into packets. This allow effective data rate swapping without RF noise (Razavi et al., 2013).

On the other hand, the feasibility of Bluetooth RFCOMM and L2CAP protocols for audio and video streaming between mobile phones by using Java ME Platform has been investigated in Sahd (2010). The incompatibility problems between devices and, simultaneously, the reliability of the data streaming and quality of the transformed audio

via wireless technology have been solved in Idwan et al. (2013). The applicability of the proposed solution is tested on audio streaming between incompatible devices modelled by two parked cars separated by a distance of two to five metres. The study of Gupta et al. (2010) presented an optimised solution to eliminate anomalies that exist in video streaming *via* Bluetooth over RFCOMM protocol in real experiments. A Bluetooth enabled surveillance systems (BESS) was proposed by Banerjee et al. (2010) to help transmitting live video streaming between two mobile phones *via* built-in cell phone camera.

Many researchers (Sillen and Nordlund, 2005; Vazquez-Briseno and Vincent, 2007) have implemented frameworks for streaming media by utilising various protocols. The authors of Vazquez-Briseno and Vincent (2007) proposed adaptable system architecture by taking in consideration the limitation of the architecture in any mobile device. This system consists of a streaming server, a multi-cast proxy and a mobile client that partially handles downloadable adaptive multi-rate (AMR) files. AMR files are compressed and divided into a number of frames, where each frame contains a 1 byte header and 20 ms of audio data. Then, frames are sent in small real time transport protocol (RTP) packets. The two players alternatively play part of the audio file instead of playing the entire file.

A similar system has been enhanced by using double buffer architecture to stream audio books to mobile phones that use general packet radio service (GPRS) and a third generation mobile telecommunication technology (3G) (Sillen and Nordlund, 2005). They decided to use AMR audio format for streaming due to its good quality and its small size compared to other used audio types. Herein, there are two main disadvantages of the previous approach:

- 1 the limited number of devices that supports other audio types without affecting the quality of audio streaming
- 2 the required time to recognise the entire audio file before being able to commence with playback.

Traditional video streaming over a wired and wireless connection networks has bandwidth complications, delay and loss of quality according to its real time nature, out of range devices and interference with other external devices are real problems in media streaming. Therefore, we address the particular issues that exist in these connections and enhance the usage of media streaming under Bluetooth environment covering three major aspects including intermediate protocols by building L2CAP-based video streaming prototype, media compression of streamed data and integrated buffering architecture.

The main contributions in this paper are:

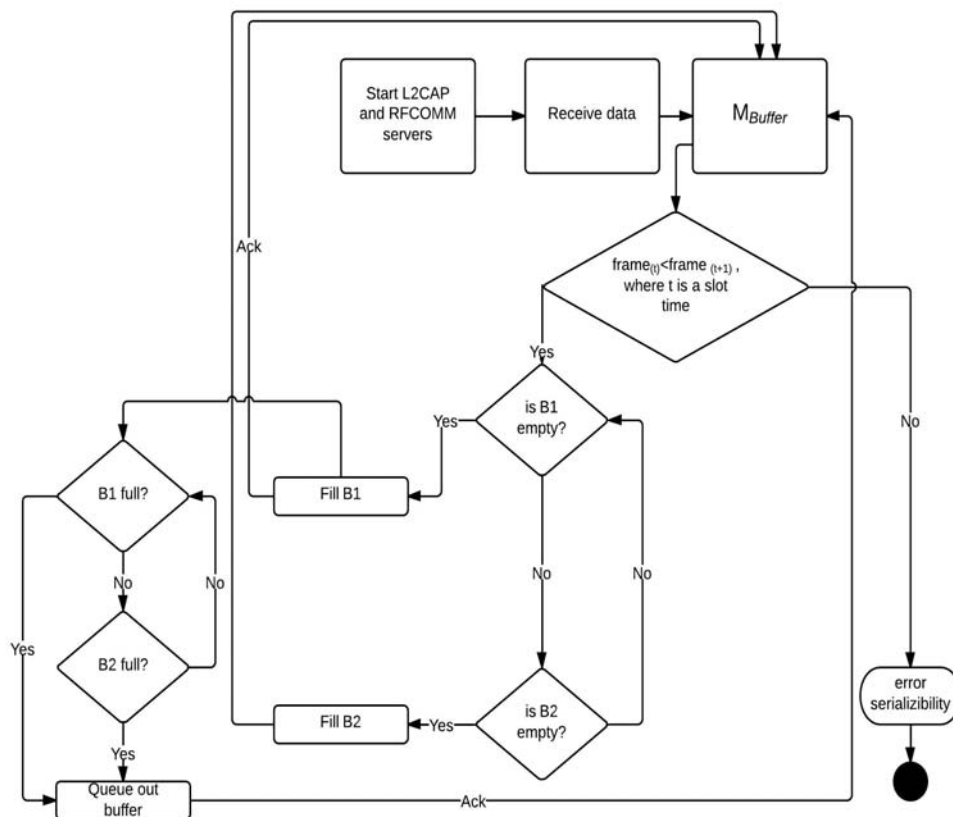
- 1 utilising L2CAP in Bluetooth media streaming that leads to a huge quality in the processed file
- 2 enhancing the double buffer method of Sillén and Nordlund that reduces the jittery playback problem in media streaming
- 3 achieving a new system by integrating the L2CAP-based prototype with the integrated buffering architecture between two different devices to stream media files *via* Bluetooth.

3 Methodology of the proposed model

Bluetooth was introduced in order to abandon the use of wired connections and to make a wireless transfer of data without involving in any synchronisation problems. Various types of data such as, documents, images, sounds and videos are transferred through the Bluetooth technology using master-slave mechanism (Holtmann, 2008). Nowadays, wireless Bluetooth connection is highly used in mobile phones, laptops, computers and many other peripheral devices (Harte, 2008). Therefore, the need for enhancing the ability of exchanging audio, pictures and videos streaming is required. Additionally, the growth of Bluetooth software has to take compatible and homogeneous architecture development that includes protocol stack and the software of the Bluetooth in the form of mobile media application program interface (MMAPI). The MMAPI allows programmers to employ the multimedia of mobile devices and to play different media that stored in the local file, database or even an application archive.

In this paper, we propose a newly implemented prototype called '*mBlueThunder*'. It is based on L2CAP that allows transferring media files through several devices connected via Bluetooth technology. The *mBlueThunder* uses master-slave architecture by embedding an integrated buffering system attached to it as shown in Figure 2.

Figure 2 The *mBlueThunder* system integrates buffering architecture model with L2CAP-based prototype



When the connection is established, the master device represents a ‘producer’ while the slave device represents a ‘consumer’. The connected devices are capable of transmitting continuous compressed media file that is partitioned into segments of chosen intermediate layer.

The Bluetooth specification contains a protocol stack that provides Bluetooth devices with the interconnectivity among other devices in the network layer. In general, the well-defined intermediate protocols that are used to access Bluetooth network are TCP/IP¹, HCI and L2CAP² protocols.

The most important issue in implementing a Bluetooth protocol stack is to choose an appropriate intermediate protocol to segment media streams. Keeping in mind, the segmentation process has to be achieved in a flexible and effective way between the link layer and the physical layer. We present a qualitative comparison between well-defined protocols with respect to the speed, quality of segmentation and reassembling processes as illustrated in Table 1. It is clearly shown that L2CAP have been chosen over HCI and TCP/IP as an intermediate protocol for the media streaming *via* Bluetooth due to the effectiveness on the overall performance and hardware compatibility issues. It is worth mentioning that the proposed implementation of MPEG-4 over Bluetooth using L2CAP is called ‘MPEG4BT *via* L2CAP’.

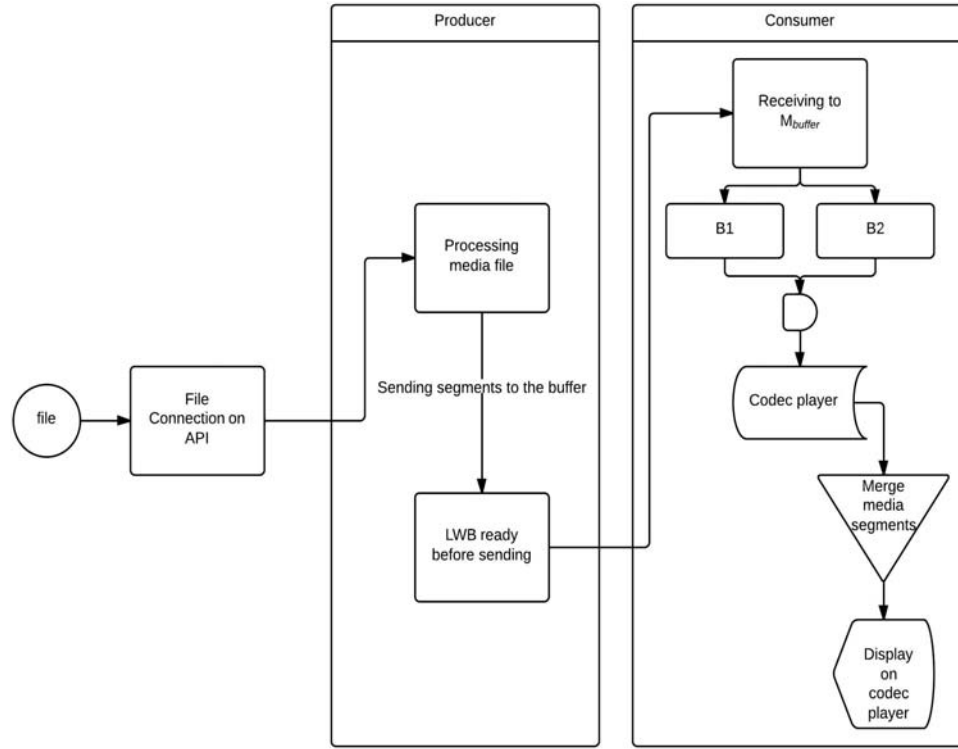
Table 1 Qualitative comparison of media streaming *via* different protocols applicable to Bluetooth technology

<i>Perspective</i>	<i>Protocols</i>		
	<i>TCP/IP</i>	<i>HCI</i>	<i>L2CAP</i>
Speed	Slow speed due to the overhead by using BNEP and LANA2P that overloads resources with additional encapsulation layer.	Slow speed due to overhead on the system recourses that deals with MPEG-4 compression.	High speed due to the compatibility with AVDTP with no overhead.
Segmentation process	SAR functionality is available but it needs additional encapsulation layers for TCP/IP solution by using BNEP and LANA2P.	Requires additional software at application layer.	SAR functionality is available with no need for additional software requirements.
Reassembly process	Easy and simple by using BNEP/LANA2P.	Requires additional software at application layer.	Easy and simple by using AVDTP.
Hardware	No modification is required.	Needs hardware modification.	No modification is required.

Our methodology depends on proposing an integrated four buffers distributed among the producer and the consumer sides as shown in Figure 3. The light weight buffer (LWB) is used at the producer side, while the master buffer (M_{buffer}) with a dynamic size and the two other attached buffers (B1, B2) of a fixed size are used at the consumer side, in addition to a ready codec player that runs files streaming. The producer side uses a file connection API that is responsible to send a serialised frame one at a time to LWB. As soon as the LWB starts sending the serialised frames to the consumer side an illusion is

created to the consumer side that the whole file is transmitted after establishing the Bluetooth connection.

Figure 3 The integrated buffering architecture



At the consumer side, M_{buffer} receives the serialised segments of a fixed size and alternatively distributes them among B1 and B2. In the default case, B1 is empty, therefore the serialised segment will be sent to it. Otherwise, B1 is busy in queuing out the received segment to the codec player while B2 is ready to receive the next segment. It is worth mentioning that, the acknowledge message (Ack) will be send to the producer informs that M_{buffer} is ready for receiving new segments. If the acknowledgment message is not received, then the connection is down. This prevents buffering overflow that results in loss of segments and guarantees that the codec player will continuously play the received segments from both buffers. The proposed methodology overcomes the jittery playback and eliminates the delay time of playing the entire file after completing the transmission process as explained in Algorithms 1 and 2.

3.1 Example

Consider file of size 20 minutes at the producer side will be sent to consumer side. The file will be partitioned into set of segments according to formula 1 $\left(file = \sum_{i=1}^n S_i \right)$. Each segment (S_i) has a fixed size equal to four minutes. The streaming schedule in the buffering architecture is explained in Table 2.

Table 2 Example of streaming schedule in the buffering architecture

<i>Time</i>	<i>LWB</i>	<i>M_{buffer}</i>	<i>Buffer 1</i>	<i>Buffer 2</i>	<i>Codec player</i>
T ₁	Queue_out (S ₁)	Receiving (S ₁)	Queue_in (S ₁)		
T ₂	Queue_out (S ₂)	Receiving (S ₂)	Queue_out (S ₁)	Queue_in (S ₂)	S ₁
T ₃	Queue_out (S ₃)	Receiving (S ₃)	Queue_in (S ₃)	Queue_out (S ₂)	S ₂
T ₄	Queue_out (S ₄)	Receiving (S ₄)	Queue_out (S ₃)	Queue_in (S ₄)	S ₃
T ₅	Queue_out (S ₅)	Receiving (S ₅)	Queue_in (S ₅)	Queue_out (S ₄)	S ₄
T ₆			Queue_out (S ₅)		S ₅

Algorithm 1 ProducerPart ()

```

1 EstablishConnection()
2 FileConnectionAPI(media file)
3 do{
4   3.1 segment = Segmentation(media file)
5   3.2 Serialisation(segment)
6   3.3 Allocating(segment, LWB)
7   3.4 Streaming (LWB, Mbuffer)
8   3.5 Acknowledgment (Mbuffer)
9 } While (segment != 0)
5 TerminateConnection()

```

Algorithm 1 starts media streaming between producer and consumer *via* Bluetooth technology by establishing a connection (line 1). The producer part uses a FileConnectionAPI that defines the process of accessing onboard and removable file systems on the device (line 2). Then the segmentation process is started by fragmenting the media file into a set of segments (line 3.1). Each segment is serialised by assigning a priority number to avoid congestion at the codec player in the consumer part (line 3.2). The serialised segment will consequently be allocated at the LWB and streamed to the M_{buffer} at the consumer part (line 3.3 and line 3.4, respectively). The next serialised segment will not be streamed unless the acknowledge message is received from the consumer part (line 3.5); otherwise, the producer part should wait until consumer part finished handling the previous streamed segment to prevent overflow and the loss of frames. The producing process will be terminated when there is no more serialised segment to be streamed or connection is down (line 5).

While Bluetooth connection is established between producer and consumer (line 1). M_{buffer} at the consumer should acknowledge the LWB at the producer in order to receive the streamed segment and to check its serialisability (line 1.1 and line 1.2, respectively). In the non-serialisable case an error will be reported (line 1.4.1); otherwise the status of B1 will be checked to receive the streamed segment (line 1.4.2.1). If B1 is available, the segment will be allocated, consequently queued out to the codec player and an acknowledge message will be sent to the M_{buffer} (line 1.4.2.2.1, line 1.4.2.2.2 and line 1.4.2.2.3, respectively). If B1 is busy in handling the previously streamed segment, B2 will absolutely be available based on Lemma 1. In this case, the segment will be sent to B2, queued out to the codec player and an acknowledge message will be sent to the M_{buffer} (line 1.4.2.3.1, line 1.4.2.3.2 and line 1.4.2.3.3, respectively).

Algorithm 2 ConsumerPart()

```

1  While (connection is available)
    1.1  Acknowledgment ( $M_{buffer}$ , LWB);
    1.2   $M_{buffer}$  = Receive (Segment)
    1.3  Result = CheckSerialisability ()
    1.4  If (Result == false)
        1.4.1  Return error serialisability
        1.4.2  else
            1.4.2.1  Status = GetStatus (B1)
                    1.4.2.1.1  If (status == Ready)
                        1.4.2.1.2  Allocate (segment, B1)
                        1.4.2.1.3  QueueOut (B1, codec_player)
            1.4.2.2  Acknowledgment (B1,  $M_{buffer}$ )
            1.4.2.3  else
                    1.4.2.3.1  Allocate (segment, B2)
                    1.4.2.3.2  QueueOut (B2, codec_player)
                    1.4.2.3.3  Acknowledgment (B2,  $M_{buffer}$ )
    1.5  TerminateConnection()

```

Lemma 1

$$Data\ rate_{Queue-in}(B1) = Data\ rate_{Queue-out}(B2),\text{ vice versa.}$$

Proof: Bandwidth detection of the API codec player module can be set to either initial bandwidth detection (dynamic) or set default buffer time (fixed). This controls the player's bandwidth detection and buffering behaviour. The M_{buffer} has a dynamic size, while the size of B1 and B2 is set to be fixed and equal. The buffer size refers to the duration of the media in the player's buffer, not to the required time needed to be loaded to the buffer.

Once playback has started, the adaptive streaming Algorithms 1 and 2 respectively, take place and calculate the best settings to use the buffer effectively to assure the continuity of file playback. Keep in mind the buffering resources might change from one device to another while the Bluetooth connection is available.

The bandwidth has been converted to a slot of time that is compatible with the segment size in order to assure that B1 and B2 has the same data rate. Thus, the synchronised time between Queue_in in B1 is equal to Queue_out in B2. Applying the same rules on B2, vice versa situation is defiantly clear. $\square$

Using our proposed L2CAP prototype in the explained architecture will reduce jittering by increasing the quality of streaming from the producer to the codec player at the consumer side. The best quality will be delivered by choosing the appropriate compression method for video streaming over Bluetooth. MPGE-4 video compression is a very good candidate due to its encoding techniques that provides more flexibility in meeting different demands of streaming along with the integrated buffering architecture (Chong and Beg, 2003). However, the codec player will be busy receiving and playing segments which are alternatively directed from B1 and B2 as shown in Lemma 1.

The best video streaming quality is served by the L2CAP prototype integrated with this buffering architecture that guarantees an equal data rate for both buffers (B1, B2). Hence, the time will be constant for queuing in and queuing out in both buffers which will allow B1 to receive the transmitted segments and queue it in at the same time that B2 is queuing out. This will keep the codec player continuously busy and prevent it from silent moments and jittery noise caused by buffering issues.

4 Discussion

4.1 Experimental results

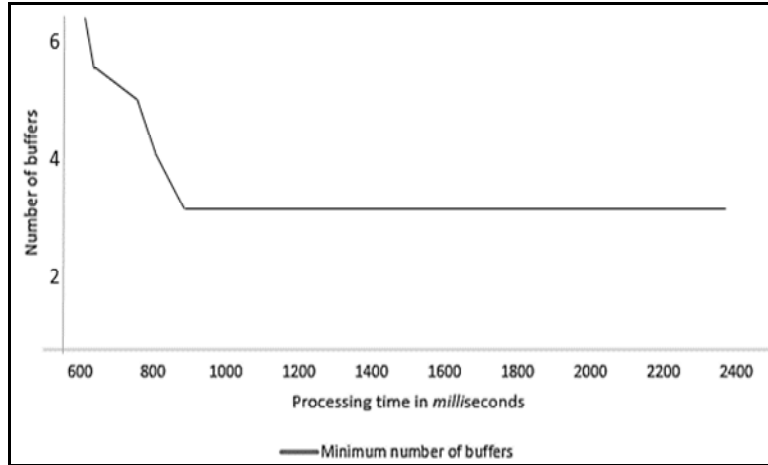
We have implemented a video streaming framework with the integrated buffering architecture as it has been shown in Figure 2. All buffers were tested virtually on PC with the following specifications: processor Intel-Inside Core i5 ($\times 64$) 2.4 GHz, RAM memory 6 gigabytes, Windows 8. The implementation of the integrated buffering architecture has been achieved by using C# V.5.0 (Griffiths, 2012). Where it has been instigated with the following specifications of the attached buffers; LWB array size of ten minutes, M_{buffer} is a dynamic buffer, B1 and B2 are two arrays of size equal or less than LWB^3 in the simulation case, each array size of ten minutes.

The experimental results consist of two parts: The first part relates to the estimation of the minimum number of buffers needed in the integrated buffering architecture, while the second part verifies the overall performance of the L2CAP-based prototype with the buffering architecture attached to it.

4.1.1 Part I

The main objective of this experiment is to determine the minimum number of buffers used in our integrated buffering architecture. The simulation has been performed by using the network simulator (ns-2, V2.28) (Issariyakul and Hossain, 2009) with the support of integrated buffering architecture implemented by C# (Deitel and Deitel, 2012). The MPGE-4 encoded file has been employed for transmission simulation test of size 7.3 MB.

The proposed integrated architecture consists of a producer and a consumer parts. Each part is equipped with a specific number of buffers. It is worth to mention that one buffer at the producer part is enough to handle transmission process and to keep it going. Therefore, the case of having one buffer in the whole system is not applicable. While having two buffers in the system, one at the producer part and another one at the consumer part, is the base step in our experiment. During the simulation experiment, we have determined the use of minimal number of buffers at the consumer part starting with one up to six buffers. Using one or two buffers at the consumer part is not viable, because it suffers a jittery playback problem (having two buffers at the consumer part is exactly the same model developed by Sillén and Nordlund model without involving L2CAP in media streaming). By increasing the number of buffers to be equal to three buffers, experiments show a consistency in the buffering architecture performance. Figure 4 shows a graph that compares different number of buffers used with respect to the processing time at consumer part. The overall performance is consistent by using three buffers to achieve convenience and modularity of the integrated buffering architecture.

Figure 4 Minimum number of required buffers in the integrated buffering architecture

Enhancing the throughput of the proposed architecture is concerned with processing time that shows to be linear after using three buffers at most at the consumer part. Therefore; using more than three buffers at the consumer side will definitely decrease the required processing time of media streaming, but this decrement will be lost with the required communication time between the buffers and the codec-player (wasting resources). We conclude that four buffers in the whole system is the optimal number. Taking in our consideration that the switching between the two buffers (B1 and B2) at the consumer part is not perceptible as it is internally processed to be transferred to the codec player. The only predictable delay occurs when M_{buffer} receives the first segment from the producer part in order to start the streaming process as it has explained previously.

4.1.2 Part 2

We focused on estimating the improvement of the processing time obtained by the proposed integrated buffering architecture compared with the Sillén and Nordlund model. The set of experiments is used to compare the average processing time needed for a file streaming between the producer and the consumer in the proposed integrated buffering architecture by using the network traces that have been recorded at both sides. The calculated processing time includes segment time interval between the submission of the segments and its completion at the consumer part, including the time spent waiting for resources. Our experiments consist of two parts: Part 1, investigates the behaviour of the proposed architecture with respect to different length of the file. Part 2, investigates the behaviour of the proposed model with respect to different buffer sizes (B1 and B2).

The behaviour of our system with different length of video file related to the processing time is presented in Table 3. It shows a comparison between our proposed system and the Sillén and Nordlund model. There are two sets of experiment where a frame size varies from one up to ten million seconds and the fixed buffer size is equal to 1,000 and 2,000 seconds, respectively. Each value presents the average of ten successive trials under the same constraints. It is clearly shown that the *mBlueThunder* has a great improvement in the total processing time required to stream the file reached up to 31%. This proves the strength of utilising the integration between L2CAP and the proposed

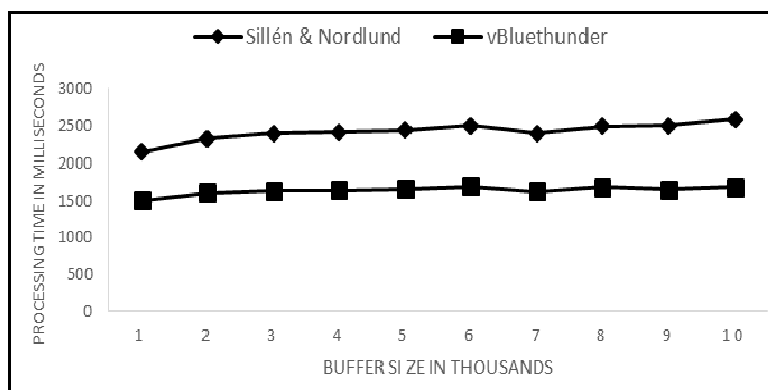
buffering architecture by using the MPEG-4 video compression⁴ in the streaming process. The *mBlueThunder* reduces the consumption energy as well as the required processing time with a high quality of streaming. This makes our system preferable compared to the state-of-the-art.

Table 3 The set of average processing time (millisecond) required for file streaming in Sillén and Nordlund system and *mbluethunder* system, respectively

<i>Dataset 1: the frame sizes of file streaming vary from one up to ten million seconds and the size of the B1 and B2 buffers are equal to 1,000 seconds.</i>			
<i>Frame size</i>	<i>Sillén and Nordlund</i>	<i>mBlueThunder</i>	<i>Percentage %</i>
1,000,000	106	89	16.04
2,000,000	374	277	25.94
3,000,000	568	405	28.70
4,000,000	763	544	28.70
5,000,000	1,021	723	29.19
6,000,000	1,102	778	29.40
7,000,000	1,218	852	30.05
8,000,000	1,820	1,268	30.33
9,000,000	2,199	1,528	30.51
10,000,000	2,322	1,602	31.01
<i>Dataset 2: the frame sizes of file streaming vary from one up to ten million seconds and the size of the B1 and B2 buffers are equal to 2000 seconds.</i>			
<i>Frame size</i>	<i>Sillén and Nordlund</i>	<i>mBlueThunder</i>	<i>Percentage %</i>
1,000,000	93	75	19.35
2,000,000	338	258	23.67
3,000,000	558	423	24.19
4,000,000	709	534	24.68
5,000,000	977	725	25.79
6,000,000	1,188	871	26.68
7,000,000	1,382	1,005	27.28
8,000,000	1,499	1,060	29.29
9,000,000	2,080	1,429	31.30
10,000,000	2,388	1,630	31.74

A remarkable behaviour of our system with different buffer sizes related to the processing time is illustrated in Figure 5. A clear difference in the turnaround time varies from 15%–40% positively toward our architecture that depends on the size of the used buffers and the size of transmitted frames comparable with different architectures. It is worth to mention that, managing the system's resources is important to complete the streaming process with best quality, to save consumption of energy and to guarantee the playback quality with a minimum overhead. This absolutely increases the system throughput and maximises the number of segments transmitted and received per unit of time.

Figure 5 The average processing time (millisecond) required for file streaming in Sillén and Nordlund model and *mBlueThunder* system, where frame sizes is equal to ten million seconds and the size of the B1 and B2 buffers are varying from 500 up to 9,000 seconds



4.2 Implications for theory and practice

mBlueThunder offers flexible enhanced multimedia streaming and live sharing over Bluetooth connections among current multimedia buffering models due to the adaptability and applicability in satisfying the required specifications in e-learning, medical and business domain.

The limitations of our system are constrained by the general limitations of Bluetooth wireless technology presented by the short range communication and low-power consumption. It is worth mentioning that rechargeable batteries are effective solution with the right battery technology that is provided by different manufactures in mobile phones.

5 Conclusions

Media streaming over Bluetooth connection is a huge challenge. Although all devices support Bluetooth technology, it requires an enhancement in the buffering scheme. Therefore, there is a room to present a new prototype of media streaming over Bluetooth by integrating L2CAP with a new buffering architecture that implement MPEG-4 for compression techniques.

A novel system utilising the integration between L2CAP and a buffering system is presented in this paper called '*mBlueThunder*'. The proposed system shows a significant result with respect to the processing time reached up to 35% comparable to the state-of-the-art. It is worthy to highlight that the integrated buffering architecture guarantees maximum capacity of continuous playback display during transmission process and the quality of media streaming. Our proposed model is adaptable and applicable to satisfy the specification required by media streaming application over Bluetooth connection, for example; learning, sharing videos without the need of Wi-Fi connection or downloading the media, among other needs.

For future work, *mBlueThunder* can be utilised to handle parallel streaming by enhancing the device with a parallel processor in order to process more than one

transmitted segments simultaneously at a time. Another avenue to be investigated is to adapt two codec players run alternatively as in Sillen's model.

References

- Amrit, P.P. and Singh, B. (2003) 'Streaming audio over Bluetooth ACL links', *International Conference on Information Technology: Computers and Communications*, Washington DC, USA.
- Banerjee, S., Mondal, D., Das, S. and Guin, R.B. (2010) 'Real-time video streaming over Bluetooth network between two mobile nodes', *International Journal of Computer Science (IJCSI)*, Vol. 7, No. 3, pp.37–39.
- Bluetooth SIG (2002) *Audio/Video Distribution Transport Protocol Specification*, Bluetooth Special Interest Group, New York, USA.
- Bluetooth Special Interest Group (2001) *Bluetooth Network Encapsulation Protocol (BNEP) Specification*, Bluetooth Special Interest Group, New York, USA.
- Butt, W.U.R. and Abbas, S. (2008) 'Study of Bluetooth technology and its impacts', *Proceedings of the 8th Conference on Applied Informatics and Communications*, Greece, pp.124–133.
- Chong, C.H. and Beg, M.S. (2003) *MPEG-4 Video Transmission Over Bluetooth Links*, IEEE International Personal Wireless Communications, New York, USA.
- Deitel, P. and Deitel, M.H. (2012) *C# for Programmers*, Prentice Hall, Michigan.
- Fayyoubi, E., Idwan, S. and Karss, A.A. (2014) 'Developing fever watcher-kit via Bluetooth wireless technology', *International Journal of Business Information Systems*, in press.
- Gartley, W. and Smith, A.D. (2006) 'Exploring the competitive and quality potentials of internet radio', *International Journal of Business Information Systems*, Vol. 1, No. 3, pp.276–291.
- Gratton, D. (2003) *Bluetooth Profiles : The Definitive Guide*, 2nd ed., Prentice Hall Professional, New Jersey, USA.
- Griffiths, I. (2012) *Programming C# 5.0: Building Windows 8, Web and Desktop Applications for the .NET 4.5 Framework*, 1st ed., O'Reilly Media, CA, USA.
- Gupta, S., Sunil, K.S. and Rachna, J. (2010) 'Analysis and optimisation of various transmission issues in video streaming over Bluetooth', *International Journal of Computer Applications*, Vol. 11, No. 7, pp.44–48.
- Hamza, N., Touati, F. and Khriji, L. (2011) 'Wireless healthcare monitoring system with ZigBee communication link support', *International Journal of Networking and Virtual Organisations*, Vol. 9, No. 2, pp.169–183.
- Harte, L. (2008) *Introduction to Bluetooth Technology, Market, Operation, Profiles and Services*, 2nd ed., Athos.
- Heidari, G. (2008) *WiMedia UWB: Technology of Choice for Wireless USB and Bluetooth*, Wiley Publishing, UK.
- Holtmann, M. (2008) 'Audio streaming over bluetooth', *The Linux Symposium*, BlueZ Project, Ottawa, Ontario, Canada, s.n.
- Hu, J-D., Wei, Y. and Wen bin, Z. (2009) 'Portable ambulatory electrocardiogram monitoring system based on Bluetooth', *International Conference on Networking and Digital Society, ICNDS '09*, Guiyang, Guizhou, Vol. 2, pp.72–74.
- Huang, A. and Rudolph, L. (2007) *Bluetooth Essential For Programmers*, 2nd ed., Cambridge University Press, New York, USA.
- Idwan, S. (2009) 'Developing a mobile application via Bluetooth wireless technology for enhancing student-instructor communication', *International Journal of Interactive Mobile Technologies*, July, Vol. 3, No. 3, pp.34–37.
- Idwan, S. et al. (2013) 'BlueThunder overcomes the current drawbacks of the Bluetooth wireless technology', *IJBIS*, Vol. 12, No. 2, pp.194–209.

- Idwan, S., Alramouni, S., Al-Khasawneh, M. and Al-Adhaileh, S. (2008) 'Enhancing mobile advertising via Bluetooth technology', *IJMC*, Vol. 6, No. 5, pp.587–597.
- Issariyakul, T. and Hossain, E. (2009) *Introduction to Network Simulator NS2*, Springer, USA.
- Jung, S., Chang, A. and Gerla, M. (2006) *Video Streaming Over Overlaid Bluetooth Piconets (OBP)*, ACM, NY, USA.
- Kammer, D., McNutt, G., Sense, B. and Bray, J. (2002) *Bluetooth Application Developers Guide*, Syngress Publishing Inc., MA, USA.
- Kapoor, R., Cesana, M. and Gerla, M. (2003) *Link Layer Support for Streaming MPEG-4 Video over Wireless Links*, IEEE, Dallas, TX, USA.
- Kassem, A., Hamad, M. and Haddad, J. (2009) 'Real time video streaming over bluetooth using software technique', *International Conference on Advances in Computational Tools for Engineering Applications*, Egypt.
- Kim, T. and Ammar, M.H. (2006) 'Receiver buffer requirement for video streaming over TCP', *Visual Communications and Image Processing Conference*, San Jose, CA.
- Kim, Y., Jooyeon, L., Youngshin, K. and Chang, N. (2011) 'Bluetooth wireless handset for people with severe motor disabilities: capstone design project for rehabilitation technology', *Microelectronic Systems Education (MSE)*, Vol. 5, No. 6, pp.5,8.
- Lindner, A.M., Vaquero, M.L. and Rodero-Mer, L. (2010) 'Cloud economics: dynamic business models for business on demand', *International Journal of Business Information Systems*, May, Vol. 5, No. 4, pp.373–392.
- Lubitz, V.D., Wickramasinghe, N. and Yanovsky, G. (2006) 'Networkcentric healthcare operations: the telecommunications structure', *International Journal of Networking and Virtual Organisations*, Vol. 3, No. 1, pp.60–85.
- Maiou, S-G. and Huang, C-Y. (2002) *Quality Degradation and Improvment of H.263 Video Transmitted in Bluetooth Packets Under the Interference of Wireless LAN*, IEEE, Taipei, China, s.n.
- Morrow, R. (2002) *Bluetooth: Operation and Use*, 1st ed., McGraw-hill, New York.
- Nguyen, T. and Cheung, S-C.S. (2008) *Multimedia Streaming Using Multiple TCP Connections* [online] <http://www.vis.uky.edu/~cheung/doc/ipccc05.pdf> (accessed April 2014).
- Nilsson, M. et al. (2011) *Modeling Mobile Business* [online] http://www.ltu.se/cms_fs/1.47081!/file/MeMo%20slutrapport%20110531.pdf (accessed April 2014).
- Okura, H., Kato, M. and Tasaka, S. (2001) *A Media Synchronization Experiment on Continuous Media Tansmission in Bluetooth LAN Access*, IEEE, San Diego, CA.
- Peterson, B.S., Baldwin, R. and Kharoufeh, J.P. (2003) 'Refinements to the packet error rate upper bound for Bluetooth networks', *IEEE, Communication Letters*, Vol. 7, No. 8, pp.382–384.
- Rambe, P. (2009) 'Social media-enhanced phones for productive learning of South African postgraduate students', *International Journal of Networking and Virtual Organisations*, Vol. 6, No. 1, pp.92–104.
- Razavi, R., Fleury, M. and Ghanbari, M. (2013) *ResearchGate* [online] http://www.researchgate.net/publication/242172934_VIDEO-STREAMING_APPLICATIONS_ENABLED_ACROSS_BLUETOOTH_V_2.0_INTERCONNECTS (accessed 2 April 2014).
- Razavi, R., Fleury, M., Jammeh, E.A. and Ghanbar, M. (2006) 'Efficient packetization for bluetooth video transmission', *Electronic Letters*, Vol. 42, No. 20, pp.1143–1145.
- Reddy, G.R., Bhatnagar, S. and Rakesh, V.V. (2010) 'An efficient algorithm for scheduling in bluetooth piconets and scatternets', *Wireless Networks*, Vol. 16, No. 7, pp.1799–1816.
- Sahd, C.L. (2010) *Bluetooth Audio and Video Streaming on the J2ME Platform* [online] <http://eprints.ru.ac.za/2150/1/SAHD-MSc-TR11-173.pdf> (accessed 1 November 2011).
- Scheiter, C. et al. (2003) 'A system for QOS-enabled MPEG-4 video transmission over Bluetooth for mobile applications', *Multimedia and Expo, 2003. ICME '03, Proceedings*, Washington DC, pp.789–792.

- Selvaradjou, K., Sharma Shankar, A., Anandakumar, U. and Sivasundar, N. (2010) 'Optimization of Bluetooth audio stream based on the estimation of proximity', *International Journal of Computer and Electrical Engineering*, Vol. 2, No. 3, pp.93–106.
- Shepherd, R. (2001) 'Bluetooth wireless technology in the home', *Electronics and Communication Engineering Journal*, Vol. 13, No. 5, pp.195–203.
- Sillen, M. and Nordlund, J. (2005) *Real-Time Audio Streaming in a Mobile Environment Using J2ME*, Umea University, Sweden.
- Smith, B. (2012) *Scholar* [online] <http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.12.377> (accessed 2 April 2014).
- Tariq, M.F., Czerepinski, P., Nix, A. and Bull, D. (2000) 'Robust and scalable matching pursuits video transmission using the Bluetooth air interface standard', *International Conference on Consumer Electronics*, NJ, USA,.
- Vazquez-Briseno, M. and Vincent, P. (2007) 'An adaptable architecture for mobile streaming applications', *IJCSNS*, Vol. 7, No. 9, pp.79–84.
- Venkatraman, S. and Yoong, P. (2009) 'Role of mobile technology in the construction industry – a case study', *International Journal of Business Information Systems*, Vol. 4, No. 2, pp.195–209.
- Wang, B., Kurose, J., Shenoy, P. and Twosely, D. (2003) *A Model for TCP-based Video Streaming*. [online] <http://lass.cs.umass.edu/papers/pdf/TR03-TCP.pdf> (accessed November 2013).
- YuHai, P. (2008) 'The research of wireless manufacturing network based on Bluetooth technology', *Proceedings of the 2008 International Symposium on Information Science and Engineering*, pp.457–460.
- Zatari, D.I. (2007) 'Development of a wireless rural telemedicine network and management system', *International Journal of Networking and Virtual Organisations*, Vol. 4, No. 2, pp.109–117.

Notes

- Streaming via TCP/IP can be achieved by using the following specifications;
 - local area network access point profile (LANA2P)
 - Bluetooth network encapsulation specification (BNEP),
more information can be found in Bluetooth Special Interest Group (2001).
- Streaming via L2CAP is achieved by using the audio/video distribution transport protocol (AVDTP) specification. Further details can be found in Bluetooth SIG (2002).
- In order to avoid overflow of segments LWB size should always remain greater than or equal to the size of B1 and B2, respectively.
- MPEG-4 usage as video codec or streaming over Bluetooth. MPEG-4 is one of the newest video compression techniques that allow much lower compression ratios than the previous MPEG-2 (Chia and Beg, 2002). MPEG-4 is ideally suited to low bandwidth applications, exactly matching the requirements for video over a wireless Bluetooth network. MPEG-4 uses motion vectors between frames to encode temporal redundancy and the discrete cosine transform (DCT) to encode spatial redundancy. MPEG-4 provides three modes for encoding an input, these are namely:
 - intra-frame (I-frame)
 - predicted-frame (P-frame)
 - bidirectional interpolated-frame (B-frame).

For further details refer to Maiou and Huang (2002) and Okura et al. (2001).