Analysis and Optimisation of Various Transmission Issues in Video Streaming over Bluetooth

Siddharth Gupta[1], Sunil Kr Singh[2], Rachna Jain[3]
[1]UG Research Scholar, [2]Asst. Professor, [3]Lecturer
Department of Computer Science Engineering
Bharti Vidyapeeth's College of Engineering
New Delhi, India

ABSTRACT
This paper elucidates the research and implementation of various factors affecting video streaming over Bluetooth RFCOMM protocol. The advent of ubiquitous wireless technology makes it feasible to transmit real-time video in mobile and pervasive environments over Bluetooth. We have studied the performance of the Bluetooth RFCOMM protocol to carry the streaming video in real experimental test scenarios. Comparison is made on the basis of various dependencies such as distance, hardware/software, WLAN interference affecting the data rates of transmission and its impact is investigated. An optimised solution is proposed to mitigate the anomalies due to latency and the factors studied.

Keywords - J2ME, Bluetooth, MMAPI, RFCOMM, Optimisation, Video Streaming

I. INTRODUCTION
Bluetooth is increasingly being used as the "last meter" technology for indoor universal wireless coverage. In such scenario, ad-hoc Bluetooth networks provides cost effective and simple communication of various mobile devices and are now targeted to extensively support multimedia traffic [1]. Two types of links Synchronous Connection Oriented (SCO) and Asynchronous Connectionless Link (ACL) are supported by Bluetooth .ACL links provide flexibility for transmitted data over network [4][2]. The Enhanced Data Rate (EDR) of IEEE 802.15.1 version 2.0 has a peak user payload of 2.2 Mb/s, which is the same as IPTV [4][2]. Bluetooth employees RF technology in the 2.4 GHz ISM band as 802.11b (WLAN). It uses frequency hopping spread spectrum (FHSS) technology at a rate of 1600 hops/s with a slot period of 625 milliseconds and hops over 79 MHz of bandwidth [6]. Streaming live video over wireless channel degrades the video quality and affects the data rates due to outside hindrance, co-channel interference, variable distance between the source and destination. Our approach is the development of a test-bed implementing wireless connection between client (Mobile device) and server (PC) and send data in the form of stream of byte array over RFCOMM (Radio Frequency oriented emulation of the serial COM ports on a PC) channel. In summary, the contribution of this paper is a scheme for video streaming over Bluetooth in which: firstly an optimised data size is selected at which the rate is maximum as well as consistent. Then the whole data is segmented in this size to have enhanced data rates. However transmission over Bluetooth being practically a challenging problem partly due to error and interface conditions. The impact of varying distances, change in hardware/software, interference due to WLAN on data rates and time taken for the data to transmit over the channel is investigated.

II. TECHNOLOGIES USED

J2ME / J2SE
Java 2 Platform, Micro Edition is a revolution in congruous to the diversities of the market aiming at embedded devices, cell phones, palmtop as new set of clients. It is prevailing due to its transformable, low cost and secure functional domain. J2ME is basically a subset of J2SE and most of its functionalities are derived from it [7]. The architecture of J2ME is divided into Configurations and Profiles. Configuration (CLDC/CDC) combines virtual machine (KVM for cell phones) is a set of predefined libraries showing device's power consumption, network connectivity and memory [8]. On the other hand profiles adds an additional layer of API's offering set of GUI's and local data storage. Java SE is a widely used platform for developing and deploying Java applications today's demanding embedded and real-time environments. It consists of a virtual machine, responsible for platform independence [3].

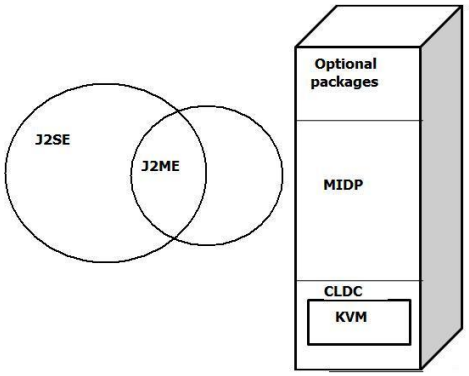

Fig 1 J2ME Architecture

Bluetooth
Bluetooth wireless technology is an open specification for low power, low cost, short range radio technology for ad-hoc wireless communication operating at 2.4 GHz and variable range of 10-100 m [1][9]. It uses radio technology called frequency-hopping spread spectrum, which divides the data being sent and transmits chunks of it up to 79 frequencies [6]. Bluetooth stack is the controlling agent for implementing the Bluetooth protocol and controlling Bluetooth device programmatically. The Bluetooth protocol is divided into Layers (Protocol Stack) and Profiles.

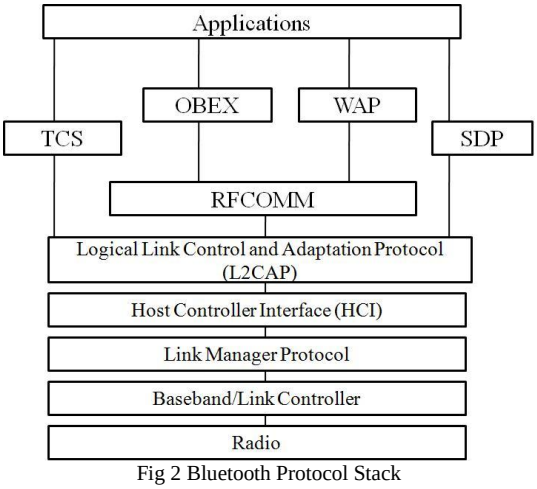

Fig 2 Bluetooth Protocol Stack

The essential steps for a Bluetooth communication includes: Stack initialisation, Device Management, Device Discovery, Service Discovery, Service registration, Communication [8][9]. The protocols used for the communication are: L2CAP (Handles data transmission in packets), RFCOMM (allows to create virtual serial port and to stream data), OBEX (allows sending and receiving objects) . RFCOMM forms a reliable peer to peer connection and is easy to implement. Similar to communicating over socket connection, data is sent via streams. L2CAP on the other hand supports multiplexing of devices and data is sent in packets. JSR 082 is used for its implementation in mobile devices using J2ME [7]

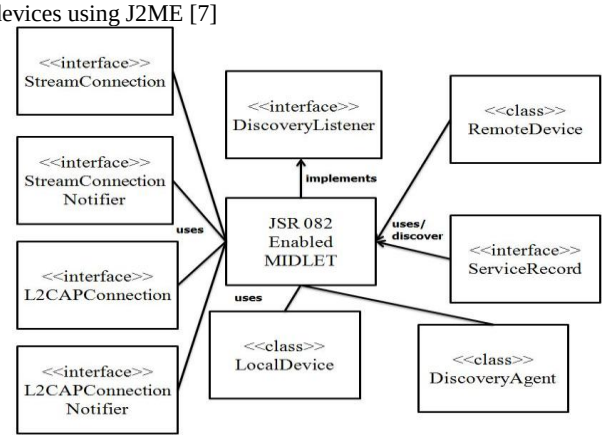

Fig 3 JSR 082 Taxonomy

III. IMPLEMENTATION FRAMEWORK

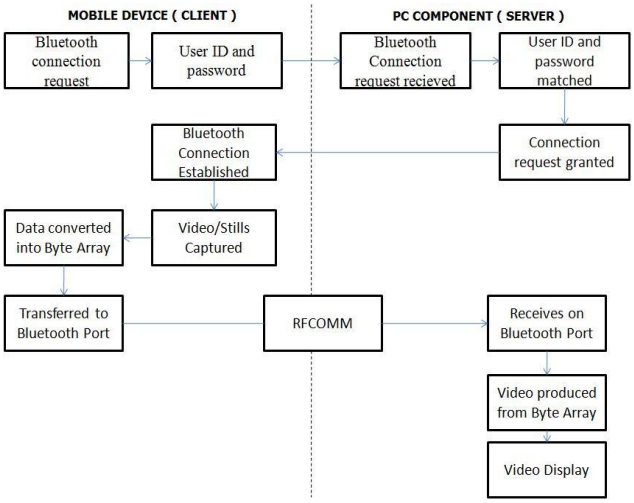

Fig 4 CLIENT SERVER ARCHITECTURE

Our test-bed mainly consisted of Nokia (Symbian OS) S60 2nd Edition Feature Pack 3 mobile phone devices, specifically the Nokia N70 and Acer Laptop with Intel® Atom™ CPU N450 @1.66 GHz, 1.33 GHz, 0.99 GB of RAM and Windows platform. The choice of using Nokia is justified by its ubiquitous use in the market and wide availability of Symbian which supports variety of languages as C++, Java ME [1]. The implementation of the server (J2SE) and client (J2ME) modules were done using Netbeans IDE v 6.8. Microsoft's Bluetooth API supports the RFCOMM protocol, making it suitable and easier to use for our study. The test-bed consisted of two parts: Mobile device as a client and PC/Laptop as server. The architecture of the test-bed application has been illustrated in the figure above. The client side is developed using J2ME which makes it platform independent and to be used in various Java enabled devices. Server side is developed in Java SE using swings API and establishes a connection with client through Bluetooth. The protocol used is btspp [8] which transmits data in streams similar to socket connections. Only one channel exists between client and server for communication. Firstly, client searches for the Bluetooth server in the vicinity and requests for a connection with already ready server. After authentication, server establishes wireless connection with the client and displays the UUID of the client device on its interface and vice-versa. Secondly, the video/stills are captured by the client using MMAPI (Multimedia API) [7] and the player thread is started with the camera of the mobile device. The data is then taken and sent via channel in the form of stream of bytes to the server over RFCOMM connection. StreamConnection and StreamConnectionNotifier classes are used in the implementation of the test-bed. The video snapshots are sent continuously to the Bluetooth port and transmitted wirelessly to the server till the thread is stopped. The data is saved in the server folder and a simultaneous analysis of the size of data sent, time taken by the data to reach the server and rate at which the data is sent is examined. Different size data is sent across the server in range of different resolution and the size with maximum and consistent data rates is chosen for transmission. Results are simulated in the next section.

IV. RESULTS AND ANALYSIS

After successful implementation of the framework with the hardware and software described, various cases were investigated for the performance analysis of the RFCOMM protocol in streaming video wirelessly. The impact of change in hardware/software specifications, variable distances and interference due to WLAN were studied and following results were obtained.

Case I: Optimisation of data size sent in order to achieve maximum and consistent data rates.

In this case, our test-bed consisted of Nokia N 70 as client and Acer Intel® Atom™ CPU N450 @1.66 GHz, 1.33 GHz, 0.99 GB RAM as server. The video streaming is done over Bluetooth and data is sent in the form of byte array over RFCOMM channel. The size of data sent was fixed and data rates and time taken for the transmission is compared with data size of 10Kb,

20Kb and 30Kb sent over the channel, with increased resolution encoding ranging from 320x240 to 500x500 pixels. At a constant distance of 0.5 m, when the data size being sent is 10Kb, the data rates and time taken for the transmission is noted for 12 consecutive transmissions. Similar tests were done with 20Kb and 30Kb of data size sent over the channel.

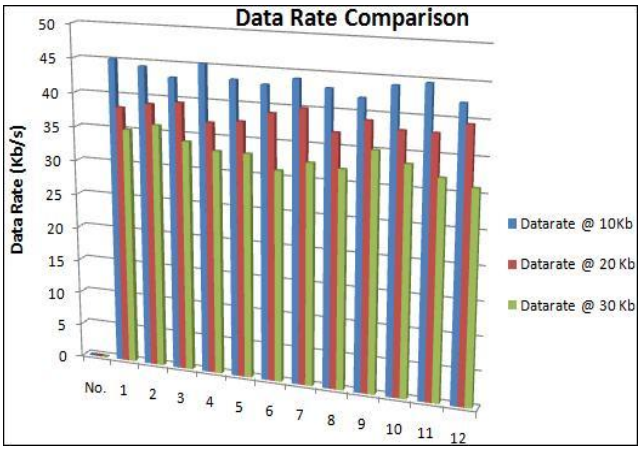

Fig 5 Data Rates at different sized data

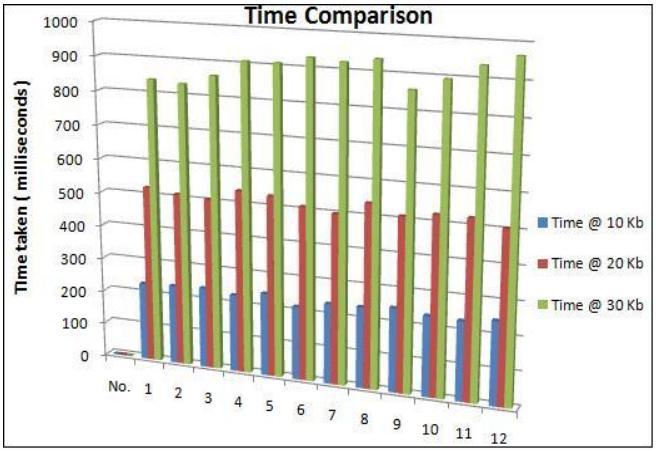

Fig 6 Time taken by different sized data

The reading were plotted on a graph, which shows that with increase in resolution and size of data transmitted, the time taken increases significantly from 229 milliseconds to 900 milliseconds. The data rate is maximum and consistent with 10 Kb of data with average data rate of 43.6 Kb/s, whereas the data rates at 20Kb and 30Kb data size were found to be comparatively less. It can be inferred that if the whole data is segmented in the size of 10Kb and sent over RFCOMM channel then data rates should be high and consistent and time taken is less, hence let the optimum data size be 10Kb.

Case II: Varying distance

In this case, the test-bed is same as previous case and the optimum data size as 10Kb for transmission at a distance of 0.5m. We checked if the data rates vary due to change in distance by sending 10Kb of data at varying distances in a closed environment so as to mitigate the external disturbances and get accurate results.

Fig 7 Data Rate Vs Distance

We notice, that on increasing the 2m distance for each set of readings, the data rates gradually decreased with increase in distance, at 0.5m the average data rate was 43.6 Kb/s for 12 consecutive transmissions. At a distance of 2m the rate was almost same with minimal drop of 1.9 Kb/s. As the distance is further increased to 4m , 6m , 8m and 10m the data rates decreased gradually and the time taken for the transmissions increased considerably. Significant decrement in data rate and increment in time taken is observed at a distance of 8m. So let the optimum distance be within 2m for maximum data rate in 10 Kb segmented streaming over Blueooth.

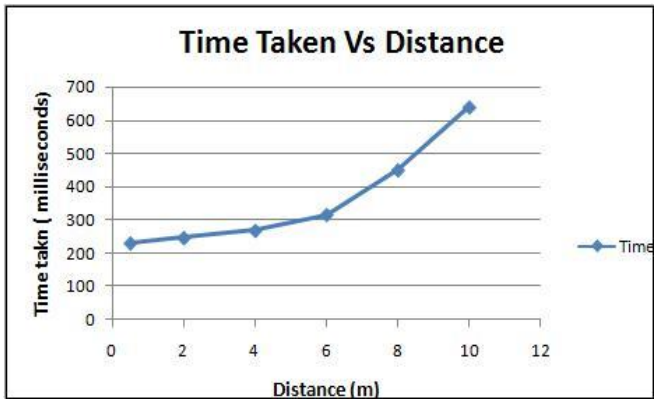

Fig 8 Time Vs Distance

Case III : Hardware and Software dependencies

Hardware dependency:

In this case, our client was fixed i.e. Nokia N 70 and server was tested with range of processors. Results with Acer Intel® Atom™ CPU N450 @1.66 GHz, 1.33 GHz, 0.99 GB RAM as server have already been simulated with data rate 43.6 Kb/s. Now processor with lower configuration and higher configurations are tested against it and the data rates are noted with simultaneous transmission of data size of 10Kb across RFCOMM Channel. When Intel® Pentium IV processor with 512 MB Ram was taken as server then the data rates dropped significantly to 23.5Kb/s and with Inspiron 1525 Intel® Core™ 2 Duo CPU T5450 @ 1.66 GHz, 1.67 GHz 2 GB RAM the data rate increased almost twice to be 96.6 Kb/s, which shows the hardware dependency of the transmission.

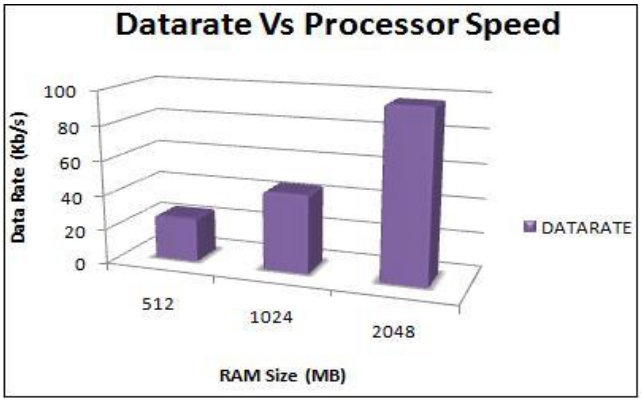

Fig 9 Data Rate Vs Processor Speed

Software Dependency:

In this case our client was varied from Nokia N 70 (Symbian OS) and Sony Erricson K750i (Sony Erricson OS) with almost same set of specifications. The transmission was done and dependency was checked due to change in client OS. The data rates and time taken for the transmission changed negligibly. It shows the test-bed being platform independent and data rates doesn't depend on change in client OS with same set of specifications.

Case IV: WLAN interference

As we know both Bluetooth and WLAN exist in the same frequency band of 2.4 GHz that is 83 MHz wide[10]. WLAN uses Direct Sequence Spread Spectrum instead of FHSS. When a Bluetooth radio and WLAN are operating in the same vicinity the single 22 MHz wide WLAN channel occupies the same frequency space as 22 of the 79 Bluetooth channels which are 1 MHz wide [5]. This is evident from the graph obtained by taken readings at 0.5 m distance when the WLAN is on and off . The data rates were affected in the presence of WLAN and further reduced with decrease in distance between the system and WLAN. According to wave theory when two waves overlap, they superimpose constructively or destructively [6]. The second graph shows the time taken for transmission which is increased in the presence of WLAN. Because of the interference there is increase in network congestion and data is continuously till it is reached, ARQ is used for effective transmission, but as the collisions are increased, hence data rate decreases and time increases for transmission in the vicinity of WLAN.

Fig. 10 Data rates in presence of Wi–Fi Interference

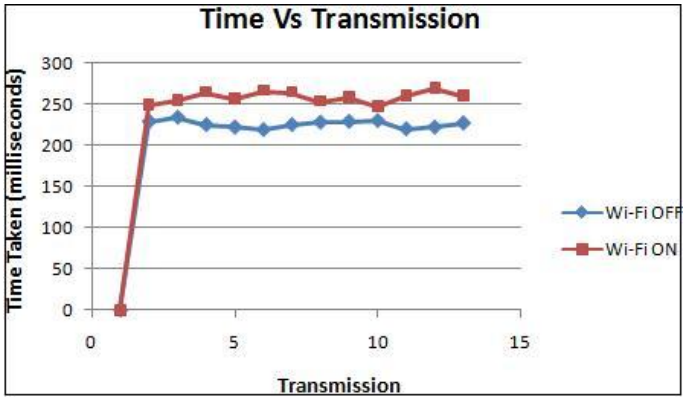

Fig 11 Time taken in presence of Wi-Fi Interference

V. LIMITATION

1. As the protocol used is RFCOMM, hence only peer to peer connection is possible over one channel. For multiple slaves to connect each other via server forming a piconet L2CAP is used .

2. In real life pragmatic situations RF noise, external and internal environment surroundings affects the rate of transmission in wireless communication. Hence latency due to unavoidable reasons.

3. The test bed can be implemented only on the mobile devices supporting Java platform.

VI. FUTURE SCOPE

1. The more advanced user interface LWUIT and MIDP 3.0 can be used and implemented in the application for concurrent Midlet execution and high graphic options.

2. L2CAP protocol can be used in place of RFCOMM where multiplexing can be done easily and data packets are sent in an efficient manner.

3. Security to J2ME mobile device can be provided using Microlog [7], which is our previous study of work.

4. 4G and IMS devices can be used for an effective video transmission wirelessly.

VII. SUMMARY AND CONCLUSION

The potential of using Bluetooth as a broadcasting medium for wireless transmission of data has been described in this paper. A test-bed was implemented successfully using Java and video streaming was done in real time. The impacts of data size, variable distances and interference due to WLAN was investigated and studies with respect to change in data rates and time taken for transmission . The data was sent using RFCOMM protocol in the form of array of bytes from client to server. It was observed that between 10Kb, 20Kb and 30Kb data size sent across 10Kb gave higher data rates as compared to others with less transmission time. With this optimum data size, the distance is varied from 0.5m – 10m and data rates were measured. Significant change in rates were observed at 8m with increase in time of transmission . The effect of WLAN interference was also

studied, which showed decrement in data rates and increase in transmission time because of the interference. Hence, if we stream video at 10Kb segmentation of transmissions in 0.5-2 m distance, with Nokia N 70 as client and Inspiron 1525 Intel® Core™ 2 Duo CPU T5450 @ 1.66 GHz, 1.67 GHz 2 GB RAM as server with WLAN at marginally greater distance, it will result in optimised video streaming over Blueooth.

VIII. REFERENCES

[1] J2ME bluetooth Programming, Master Theises by Andre N Klingsheim, University of Bergen, 2004

[2] Karma Kelzang Eudon et al, Video Streaming over 802.11b in the Presence of Fading due to Human Traffic and Bluetooth Interference, University of New Brunkswick, IEEE 2009

[3] Martin De Joed, Programming Java 2 Mcroi Edition on Symbian OS, A developers guide to MIDP 2.0 , 2004

[4] M. Siddique Khan et al, Real Time Steaming Video over Bluetooth Network, Aligarh Muslim University, IEEE 2009

[5] M. Fleury et al, Video Streaming Applications Enabled Across Bluetooth V. 2.0 Interconnects, University of Essex, IEEE 2008

[6] R. Razvi et al, Fuzzy Control of Adaptive Timeout for Video Streaming over a Bluetooth Interconnect, University of Essex , IEEE 2007

[7] Siddharth Gupta ,Sunil Kr. Singh, Microlog – Powerful Logging in J2ME, International Conference on Computer Science and Information Technology, Banglore India [in press]

[8] Siddharth Gupta, J2ME based Mobile-POS Application Intergrated with Web Services and Bluetooth Technology, Internation Conference on Computer Applications, Pondicherry India [in press]

[9] Timothy J Thompson. et al, Bluetooth Application Programming with the Java Api's , 2008

[10] Wang Xiaohang, Video Streaming over Bluetooth : A Survey , School of Comuting, NUS, IEEE 2008