Performance Optimization of MTP Stack

Greeshma Prakash[1], Dr.B.K.Sujatha[2]

[1]*MSRIT*, [2]*MSRIT, Bangalore, India*
greeshmaprakash@gmail.com, bksujatha08@gmail.com

Abstract

The paper deals with the steps proposed for performance optimization of PC to device file transfer. From the Key Performance Index (KPI) testing its seen that PC to device file transfer takes more time than Device to PC whose performance is good at the MTP (Media Transfer Protocol) level. Hence necessary steps are taken to increase the performance of PC to Device transfer, for ex: for a file transfer sized 626 MB used to take 143.319 seconds and after necessary optimization the time taken for the same 626MB file transfer is 113 seconds i.e. the throughput of nearly 1.163 MB increase is achieved. Secondly Enhanced enumeration is supported where the parent and child handles are called only when required.

Keywords: *Media transfer protocol (MTP), Key performance Index*

1. Introduction

Media Transfer Protocol, or MTP, is a protocol that is designed for content exchange with and command and control of transient storage devices. The primary purpose of this protocol is to facilitate communication between media devices that have transient connectivity and significant storage capacity. This includes the exchange of binary objects and the enumeration of the contents of that connected device.

The secondary purpose of this protocol is to enable command and control of the connected device. This includes the remote invocation of device functionality, monitoring of device- initiated events, and the reading and setting of device properties. This paper refers to the connection established and data transfer between the initiator (PC) and the responder (MOBILE PHONE) which supports the MTP stack as shown in the Figure 1.

Figure 1: General set up

Time taken for the file/data transfer plays a vital role in the throughput determination of the device. In this aspect Key performance index (KPI) test is performed on two test devices namely Device 1 and Device 2 for the PC to device and device to PC file transfers.

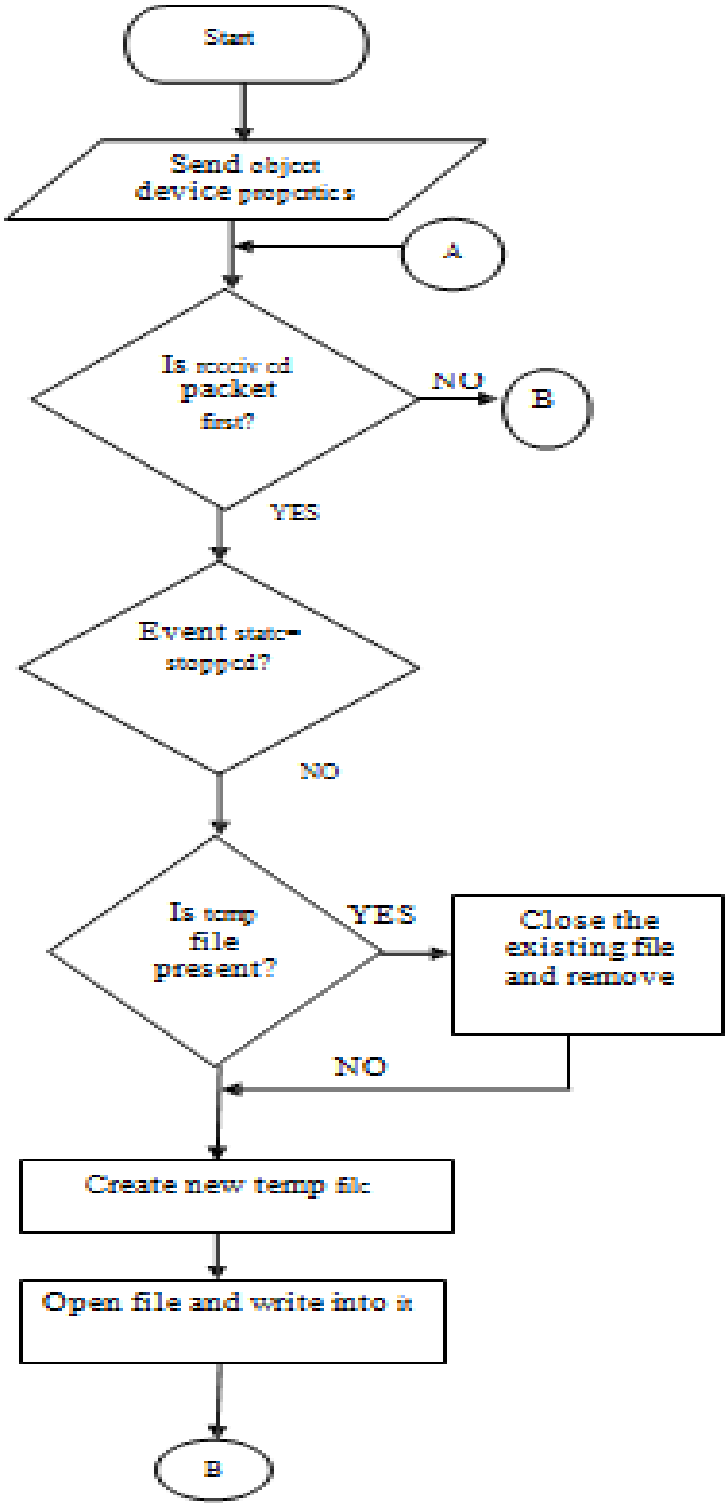
Table 1: KPI Parameters Before Optimization

TEST ITEM	AVG TIME (secs)	AVG. THROUGHPUT (Mbps/sec)
1GB File Transfer (PC to Device)	166.4	50.34
1GB File Transfer (Device to PC)	102.4	80.10
100MB File transfer (PC to Device)	13.19	61.31
100MB File transfer (Device to PC)	11.116	72.18
250 Files Transfer (PC to Device)	157.8	12.74
250 Files Transfer (Device to PC)	74.88	26.71
Enumeration (default packages)	1	10.00
Enumeration (100 files/folders)	5.49	18.49
Enumeration (1000 files/folders)	69.00	14.70
Enumeration (2000 files/folders)	136.00	14.79
Enumeration (4000 Files/Folders)	319.40	12.55
Delete objects (2000 files/folders)	144.2	13.99

From the table 1 it‟s seen that PC to device file transfer takes more time as compared to device to PC file transfer. This large time affects the throughput of the device. Hence the need for optimization was found by this and necessary steps are taken to improve the performance of PC to device file transfers. As a result of this, necessary changes in the code are employed by removing four “IF” conditions and increasing the buffer size which has lead to the throughput increase of nearly 1MB. Enhanced enumeration feature is made available which again adds to the increase in performance of the device.

2. Before Optimization

The flow of the code before optimization for PC to device transfer is explained in this section by the flowchart below in figure 2.The flowchart below shows the general flow of the code before the necessary changes are done for the increase in the performance.


Start
Send object device properties
A
Is received packet first?
NO
B
YES
Event state= stopped?
NO
Is temp file present?
YES
Close the existing file and remove
NO
Create new temp file
Open file and write into it
B

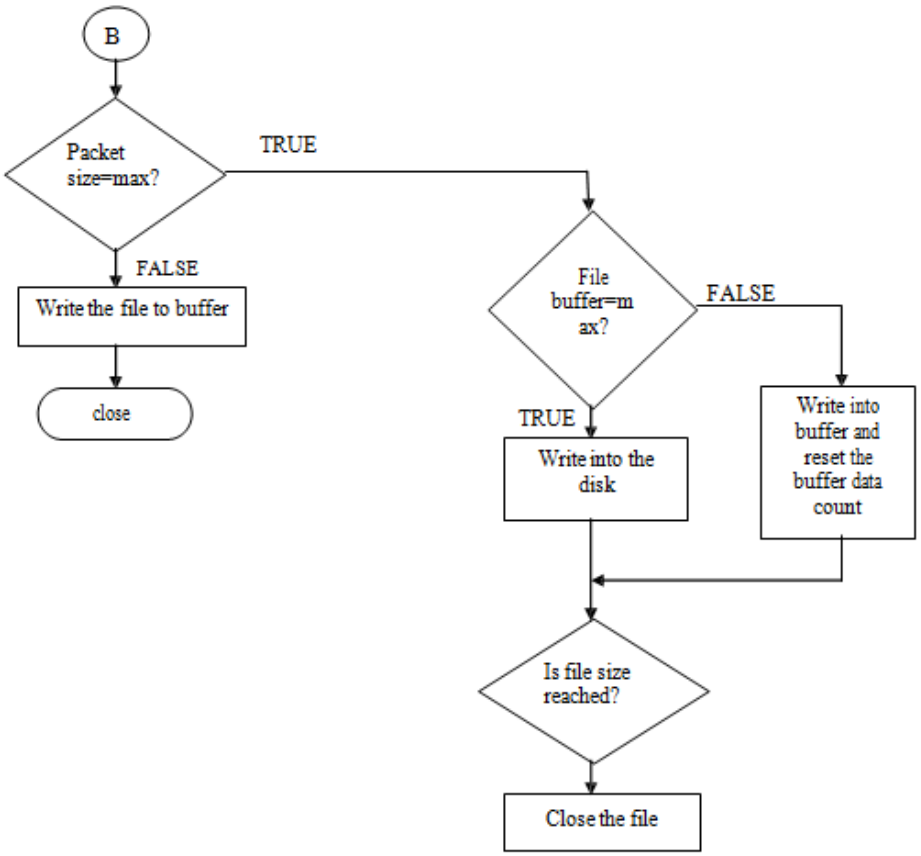


Figure 2: Flowchart of PC to device transfer code

Figure 2 above shows the code flow of PC to Device file transfer. Initially when the device is connected to the PC, a process called enumeration takes place where in which the device properties like vendor ID, manufacturer details, the file size and folder details present in the device, file creation time, modified time, attribute modes are passed into the PC .i.e. device object handles are passed into the host (PC). After getting the device property the packet is checked to see if it"s the first packet or not. If the packet is first packet in the session then the event status is checked to see if the device is in the stopped state. If the device is in other state then the presence of the „temp file" is seen. If a temp file already exists, then the existing old temp file is closed and removed, else a new temp file is created and writes into the new temp file. The above process happens only once in a session and for the first packet.

After the first packet the count is incremented and the function for the other packets is called. Here the data length is compared with the maximum packet size by ex-oring and the result obtained is performed NOT operation. If the result of the NOT operation is 0, it signifies false and 1 signifies true condition. In the process it"s checked if the maximum packet size has reached or not. If it has not reached max size then the file/data is written into the buffer and the process is closed. If the maximum size is reached then the file buffer size is checked if it has reached maximum or not. Here the maximum buffer size is added 1 and subtracted with the buffer data size and the data length.

If the result of this is a positive value it signifies that the max buffer size has not reached and if the result is a negative value it means that that the maximum buffer size has reached

and hence the data should be written to the disk space. And after writing the file it is checked if the file size has reached. If yes then the file is closed. In case of the positive value obtained the data is written into the buffer and in this particular operation the 32 bit data is shifted to the right 31 times to check for the sign bit. The result of the shift operation is ex-ored with 1 to check if the result is 0 or 1. If the result is 0 it signifies that maximum buffer size has reached and 1 signifies that the buffer size has not yet reached maximum. The above PC to device code involves the usage of four IF loops which consumed more time and hence time optimization was required in order to increase the throughput of the file transfer.

3. Optimization Process

The result of KPI testing leads to the need for the optimization of PC to device process of file transfer. Basically two types of optimization are possible namely: time optimization and memory optimization. There is no need for the memory optimization due to the availability of enough memory and due to the "not up to the mark" performance of PC to Device file transfer necessary changes has been done in the code to improve the performance. In this process of optimization four "IF" loops have been replaced with function pointers and small functions are used to perform the same operations. The function pointers used here check if the incoming packet is first or other packet, reached maximum packet size or not and also whether it reached the maximum buffer size. If the above conditions are met then the concerned operation is carried out. The figure 2 is the flowchart showing the operations carried out in the function pointers. For ex: Here since four "IF" loops are removed, for a file transfer of 1 GB, the reduction in the number of cycles with the optimized code can be explained as:

Number of Packets (N) = (file size + header size) / packet size (1)
File size=1GB= 1073741824 bytes (2)
Header size = 12 bytes (3)
Packet size = 1024 bytes (4)
Substituting Equations (2), (3) and (4) in Equation (1) we get;
N= (1073741824 + 12) / 1024
N= 1073741836 / 1024
N=1048576
Other packets = (N-1) = 1048576-1 = 1048575

Consider IF statement consists of three cycles. Since four IF loops are removed, hence a total of
4*3 i.e. 12 cycles is eliminated. In file transfer of size 1GB, a total of (N-1)*12 cycles are reduced i.e. 12582900 cycles have been reduced which in turn increases the performance by reducing the time taken for the file transfer.The second work the paper deals is the enhanced enumeration feature which is added as another performance improvement factor in the optimization process. Before enhanced enumeration was used the enumeration consisted of following two parts:

1. Listing of files in the folder and
2. All the properties received to every particular file

Initially at the time of initialization of MTP, we were doing first step as mentioned above

i.e. listing of files in folder recursively. After open session is established we will send all files and folders list to the PC. Later PC will ask for the properties of the file in particular as said in the 2nd step above. In enhanced enumeration, change is made in the first step where list of files in the folder is not sent before initialization. Now after initialization process the properties of the file of selected folder only are sent to the PC. By this, the time taken to list and pass the properties of the files is saved.

In the MTP initiation process when the device is inserted thread called event handler and enumerate structure are created where enumeration process takes place with the enumeration finish event at the end. As shown in figure 3 in enhanced enumeration process these threads are not created whereas once the device is connected, the initialization takes place and after that, the property of the folder which is needed/clicked is only sent to the PC. The enhanced enumeration feature results in improving the time taken as only object handles of the folder required are sent to the initiator thus saving the time and increasing the throughput. Next section deals with the simulation results from the KPI testing before and after optimization for the two test devices.

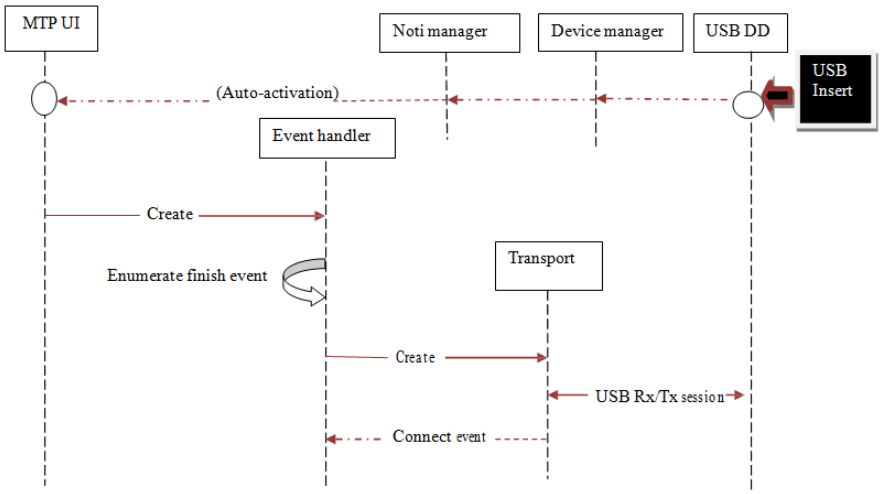


Figure 3: Enumeration process set up

4. Results

The result of the above changes done in the code leads to the following values taken for the Device 1 and Device 2 for a file sized 626 MB. The time taken for the transfer of the file and the throughput can be seen as tabulated below:

Before optimization (Device 1)--626MB
==========

1. 143.319	=>	4.367 MB/s
2. 143.843	=>	4.351 MB/s
3. 177.086	=>	3.535 MB/s

After optimization (Device 1) --626MB
==========

1. 113	==>	5.53 MB/s
2. 118	==>	5.30 MB/s
3. 128	==>	4.89 MB/s

Figure 4 below shows pictorially the comparison of KPI Para meters of Device 1 before and after optimization has been done.

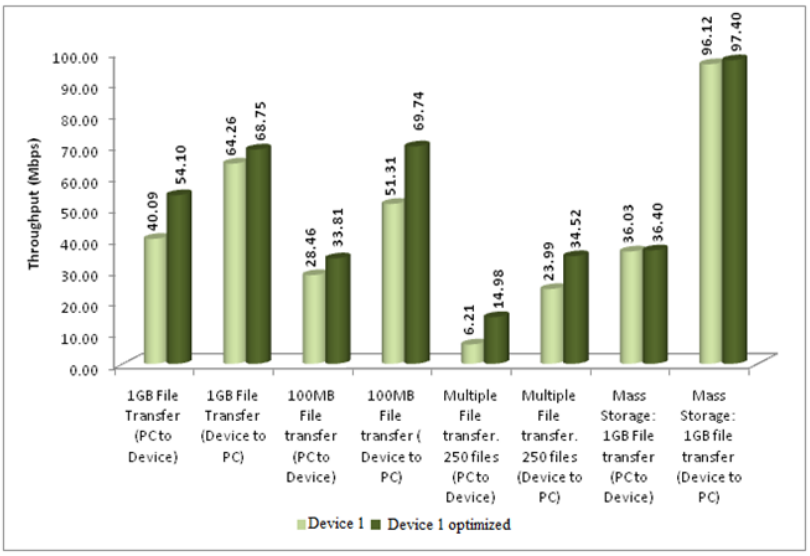


Figure 4: Throughput comparison before and after optimization for device 1

Figure 5 below shows the KPI parameters for the device 2 before and after the optimization process has been done.

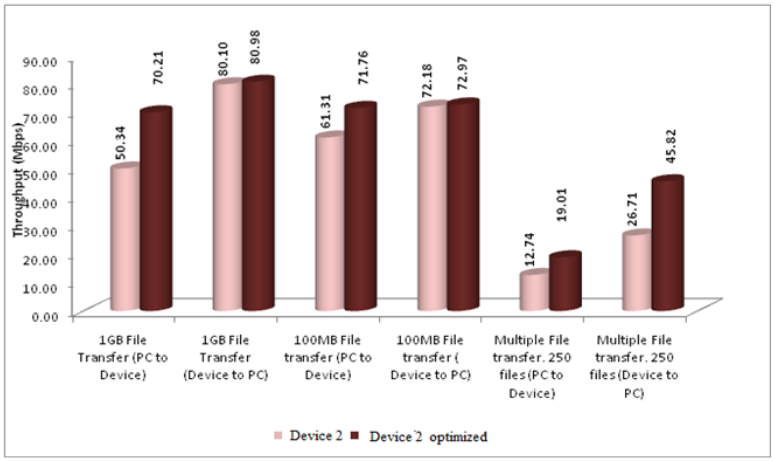


Figure 5: Throughput comparison before and after optimization for device 2

Figure 4 and Figure 5 shows the throughput comparison for the device 1 and 2 before and after the optimization process. It can be seen that the throughput has increased with the optimization process involved. After the necessary changes has been employed by avoiding recursive passing of the folder details i.e. parent and child handles the enumeration time increased to a greater extent thus reducing the time taken for the PC to device transfer and thus increasing the performance.

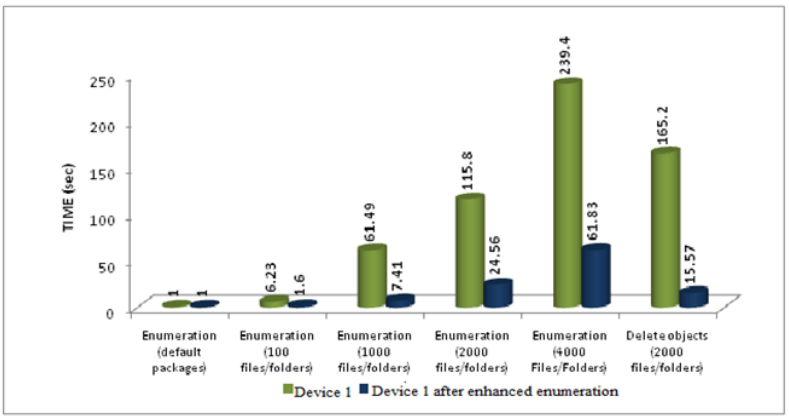


Figure 6: Enumeration time before and after optimization for device 1

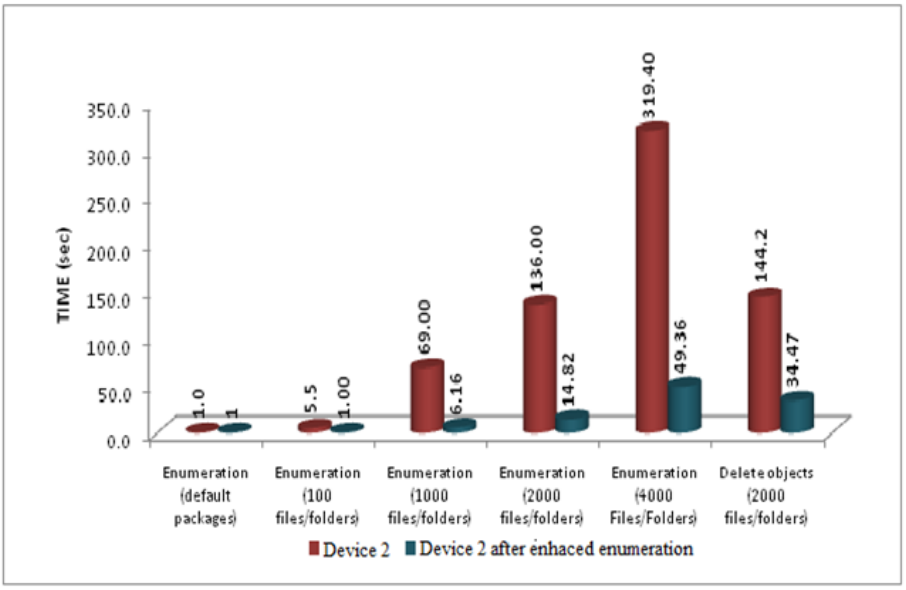


Figure 7: Enumeration time before and after optimization for device 2

Figure 6 and figure 7 shows the comparison of time taken for enumeration before and after enhanced enumeration feature has been employed in the code as explained in the above sections and tested for Device 1 and Device 2.

5. Conclusion

The necessary coding scheme employed for the optimization, taking into consideration the performance improvement of the PC to device data transfer promises to reduce the number of cycles required for optimization and hence contributes in reducing the time required for data transfer, therefore improving the throughput by a considerable amount. The new code has been used in devices which support the MTP stack and key performance index testing is carried out to check for the improvement in performance.

The future work that can be done in this paper is to support the MTP v2.0 specifications as against the MTP v1.0 which is currently being used. Secondly, initiator to be used as host.

In this case, external devices like MP3 player, digital camera etc can be plugged into the phone and data exchange can take place directly between the phone and the device.

The results of the paper can be summarized as shown in the table 2 below in terms of average time and average throughput for the old and the new MTP code after the above mentioned changes are done.

Table 2: Comparison between old and new MTP code

MTP KPI Parameter		Old MTP		New MTP	
Sl. No	Test Item	Avg Time (Secs)	Avg Throughput (Mbps or Files/sec)	Avg Time (Secs)	Avg Throughput (Mbps or Files/sec)
1	1GB File Transfer (PC to Device)	166.4	50.34	116.86	70.21
2	1GB File Transfer (Device to PC)	102.4	80.10	101.20	80.98
3	100MB File transfer (PC to Device)	13.19	61.31	11.20	71.76
4	100MB File transfer (Device to PC)	11.116	72.18	11.00	72.97
5	250 Files Transfer (PC to Device)	157.8	12.74	105.38	19.01
6	250 Files Transfer (Device to PC)	74.88	26.71	43.80	45.82
7	Enumeration (default packages)	1	10.00	1.00	10.00
8	Enumeration (100 files/folders)	5.49	18.49	1.00	100.00
9	Enumeration (1000 files/folders)	69.00	14.70	6.16	162.50
10	Enumeration (2000 files/folders)	136.00	14.79	14.82	134.95
11	Enumeration (4000 Files/Folders)	319.40	12.55	49.36	87.84
12	Delete objects (2000 files/folders)	144.2	13.99	34.47	58.66

References

[1] Jan Axelson, "*USB Complete: Everything You Need to Develop USB Peripherals"*, Lakeview Research LLC, Third edition, 2005.

[2] Steve Kolokowsky, "*Introduction to MTP: Media Transfer Protocol*", Cypress semiconductor

[3] Yashavant Kanethkar, "*Let Us C*", BPB Publications, 10th edition, 2010

[4] Media Transfer Protocol Enhanced Specification, http://www.microsoft.com/downloads/en/details. aspx?FamilyID=fed98ca6-ca7f-4e60-b88c- c5fce88f3eea&DisplayLang=en

Authors

GREESHMA PRAKASH received the B.E degree in Electronics and Communicationengineering from Visvesvaraya Technological University, Karnataka, India, in 2009 and is currently pursuing M. Tech degree in Digital communication engineering from M.S. Ramaiah Institute of Technology, Bangalore.

B.K.SUJATHA received the Bachelors degree in electronics and communication engineering from Mysore University, Karnataka, India, in 1987, M.E degree in Electronics Engineering from Bangalore University, Bangalore, in 1993, and the Ph. D. degree at Visvesvaraya Technological University, Karnataka, India in 2010. She is currently working as Professor with the department of Telecommunication Engineering, M.S. Ramaiah Institute of Technology, Bangalore. She had served as teaching faculty at SJCIT, Bangalore for 16 years up to 2004. Her research areas include speech & audio signal processing, statistical signal processing and modeling, multi-rate systems.