

## Message Communication A New Approach

Poonam P. Bhamare<sup>1</sup>, Prof. Prakash P. Rokade<sup>2</sup>, Prof. Devyani P. Bhamare<sup>3</sup>

<sup>1</sup>Department of Computer Engineering, SND COE & RC, Yeola  
Savitribai Phule Pune University,  
Maharashtra, India

<sup>1</sup>[bpoonam1212@gmail.com](mailto:bpoonam1212@gmail.com)

<sup>2</sup>Department of Computer Engineering, SND COE & RC, Yeola  
Savitribai Phule Pune University,  
Maharashtra, India

<sup>2</sup>[prakashrokade2005@gmail.com](mailto:prakashrokade2005@gmail.com)

<sup>3</sup>Department of Information Technology, Sanjivani COE, Kopargon  
Savitribai Phule Pune University,  
Maharashtra, India

[bhamare.devyani29@gmail.com](mailto:bhamare.devyani29@gmail.com)

**Abstract:** *With the increase in smart phone users and communication through messaging. In all smart phones there are various applications (apps) which can be used for communication with each other. But many of these apps are sending text data in plain text format through network. When using such apps in a public Wi-Fi network, anybody can able to sniff incoming and outgoing messages. Data compression is a common requirement for most of the computerized applications. There are number of data compression algorithms, which are dedicated to compress different data formats. Even for a single data type there are number of different compression algorithms, which use different approaches. Mobile communication devices have become popular tools for gathering and disseminating information and data. This paper proposes an efficient data compression technique by doing some modification to Huffman Coding. The aim of this paper is to compress data to send through Android based mobiles to optimize bandwidth.*

**Keywords:** Huffman Coding, Mobile application, Data Compression.

### 1. Introduction

Last decade has witnessed exponential rise in use of smart phone and communication applications of it. Actual reason behind it is the need of accessing online services and information and for staying connected to the world around them. As Carrying smart phone is easy than to carry the laptops and modem together. The users those are communicating online are also increased as the cost of data planes are in budget than the normal recharge options. The sudden rise in such users have created great opportunity in designing such applications which assures proper use of bandwidth and security issue.

Compression implies reducing the quantity of data used to represent a file, image or video content without excessively reducing the quality of the original data. It also reduces the number of bits required to store and/or transmit digital media. To compress something means that you have a piece of data and you decrease its size.

Huffman coding is a type of lossless data compression technique. Huffman coding is based on the frequency of occurrence of a data item i.e. frequency of occurrence of characters in text message. The technique is to use a lower number of bits to encode the data in to binary codes that occurs more frequently.

Now a day's mobile communication is becoming very popular for getting and giving information and data. In normal Huffman coding the tree generation and coding or decoding both the processes are done at the same side. In this paper we are going to do some modification in the normal Huffman code such that the tree will get generated at the server side and compression and decompression will be done at the client side with the help of tree generated by server.

## 2. Problem Definition

In May 2011, a security problem was reported which left WhatsApp user accounts open for session hijacking and packet analysis. WhatsApp messages were neither encrypted nor compressed and data was sent and received in plaintext, which means messages could easily be read if packet traces were available. When using WhatsApp in a public Wi-Fi network, anybody was able to sniff incoming and outgoing messages (including file transfers). The company claims that the latest version of the software will encrypt messages without giving any details about cryptographic methods they are using. So we can say the encryption and security problem has been handled by the company, but still they did not handle compression issue.

The very idea behind this paper is to provide a message communication system that compresses the text messages.

## 3. Compression

**Compression** refers to reducing the quantity of data used to represent a file, image or video content without excessively reducing the quality of the original data. It also reduces the number of bits required to store and/or transmit digital media. To compress something means that you have a piece of data and you decrease its size. There are different techniques who to do that and they all have their own advantages and disadvantages.

### 3.1 Types of Compression

- **Lossy compression** means that some data is lost when it is decompressed. Lossy compression bases on the assumption that the current data files save more information than human beings can "perceive". Thus the irrelevant data can be removed.

- **Lossless compression** means that when the data is decompressed, the result is a bit-for-bit perfect match with the original one. The name lossless means "no data is lost", the data is only saved more efficiently in its compressed state, but nothing of it is removed.

### 3.1 Huffman Encoding

Huffman coding is an entropy encoding algorithm used for lossless data compression in information theory. The term refers to the use of a variable-length code table for encoding a source symbol, where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence or frequency for each possible value of the source symbol [4]. In Huffman coding the characters in a data file are converted to a binary code, where the most common characters in the file have the shortest binary codes, and the least common characters have the longest binary codes. Huffman encoding uses a strictly binary tree where each non leaf node has two children. The Huffman algorithm works as follows [4]

### 3.2 Creating Huffman Tree

1. Start with as many leaves as there are symbols.
2. Enqueue all leaf nodes into the first queue (by probability in increasing order so that the least likely item is in the head of the queue).
3. While there is more than one node in the queues:
  - 3.1. Dequeue the two nodes with the lowest weight.
  - 3.2. Create a new internal node, with the two just removed nodes as children (either node can be either child) and the sum of their weights as the new weight.
  - 3.3. Enqueue the new node into the rear of the second queue.
4. The remaining node is the root node; the tree has now been generated.

### 3.3 Code Generation of each symbols

1. Start from the root node. For each down left traversal, add a '0' to the code and a '1' for each down right, add a '1'.

- When you reach a leaf node, the current code is the code for that character.
- When travelling to the parent of a node, delete the last added bit from the code.[1]

Let's consider an example

Message to be encoded is

“dad ade fade bead ace dead cab bad fad cafe face”

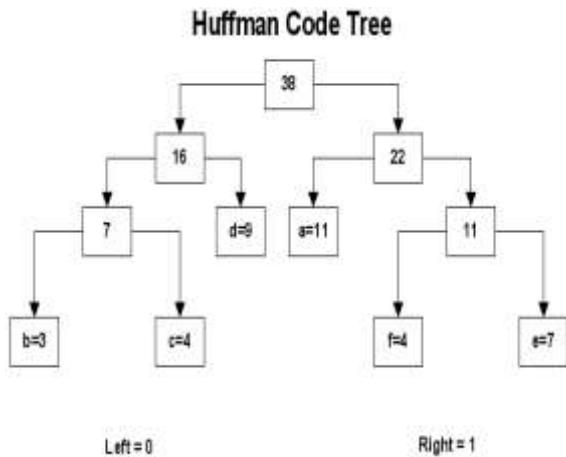


Figure 1: Huffman Code Tree

From the above figure 1. we can generate the following Huffman codes.

Table 1 Huffman Code Generation

Element	Frequency	Huffman Code
a	11	10
b	03	000
c	04	001
d	09	01
e	07	111
f	04	110

Encoded Message: 01 10 01 10 01 111 110 10 01 111 000 111 10 01 10 001 111 01 111 10 01 001 10 000 000 10 01 110 10 01 001 10 110 111 110 10 001 111

The Huffman encoding of the message is 94 bits long. The Huffman encoding saves 20 bits. The compression ratio is 1.21 to 1. The compression rate is 17.5%.

#### 4. Proposed System

In this proposed system we have used client server architecture to design the system. In which we have used divide and conquer method to divide the Huffman coding functionality of generating tree and compression at server and client respectively. The admin will provide the text file of messages to the server to generate Huffman tree. When the user registers to the application the server will send the generated Huffman tree on to the user's mobile device. When user sends the message, before it passing to the network the application itself compresses the message by using the Huffman tree in local data base. Such way the compressed message will be sent to the network. At the server, it will store all the client details. It will also store the message details. At receiver side when receiver touches the particular message to read form list the message will get decompressed and will be open to read in plain text format.

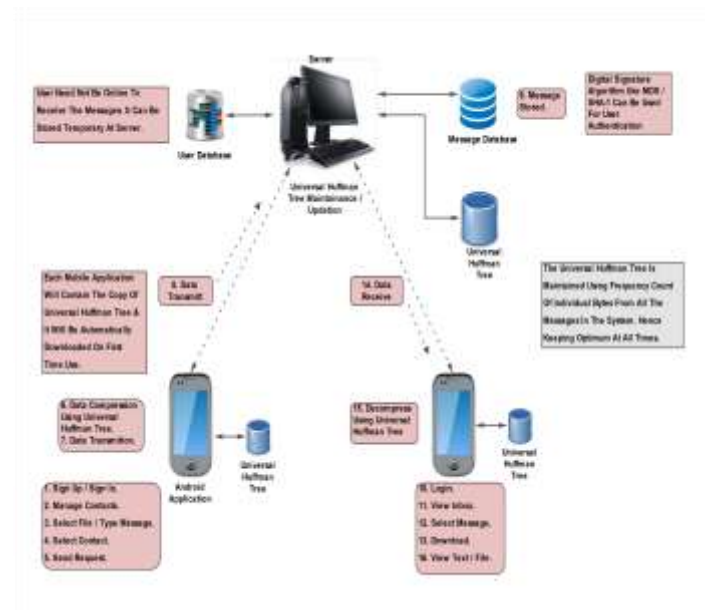


Figure 2: Proposed System Architecture.

In proposed system we have created a chatting application with data compression over network. In this we have created the forms as follow:

##### 1) Admin Module:

- Tree Generation: In this admin will upload the text file of messages to generate tree.
- Compress: In this we could compress any text file .

- Decompress: In this by uploading compressed file we could generate decompressed version of it.

## 2) Client Application:

- Server Connection: In this form we will connect to the server by entering IP Address of it.
- Main Menu: In this form there are menus like read message, send message, help, about. By selecting particular we will get that functionality.
- Message Send Form: In this form we could select the contact number of the person to whom we want to send message. This form also shows the main results of compression.
- Message List: In this form there is list which shows the messages received by selecting particular we could read it.
- Message Read: In this form we get the decompressed message.

3) Servlet Modules: This module is for the server to handle the client requests.

## 4.1 Flow Of System

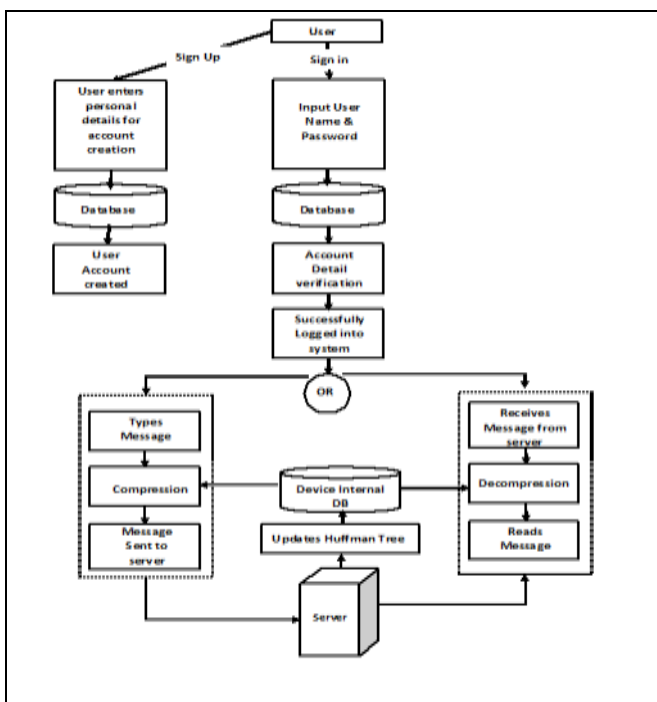


Figure 3 System Flow Diagram

## 5. RESULTS ACHIEVED

We studied this application for different text messages, Using these statistics, we observed that the new approach of Huffman coding is giving good results for sending text messages over internet to save bandwidth of a network. It is observed that using the new approach of Huffman coding we will get around 50-80 percent compression ratio on text messages.

Input Message	Input Characters	Output Characters
M1	15	10
M2	35	28
M3	10	6
M4	12	9
M5	20	15

## 6. Proof Of Concept

Our fully functional demo application has been developed in android under the development tool Eclipse ADT. We tested the application for different text files and we get the desired compression result.

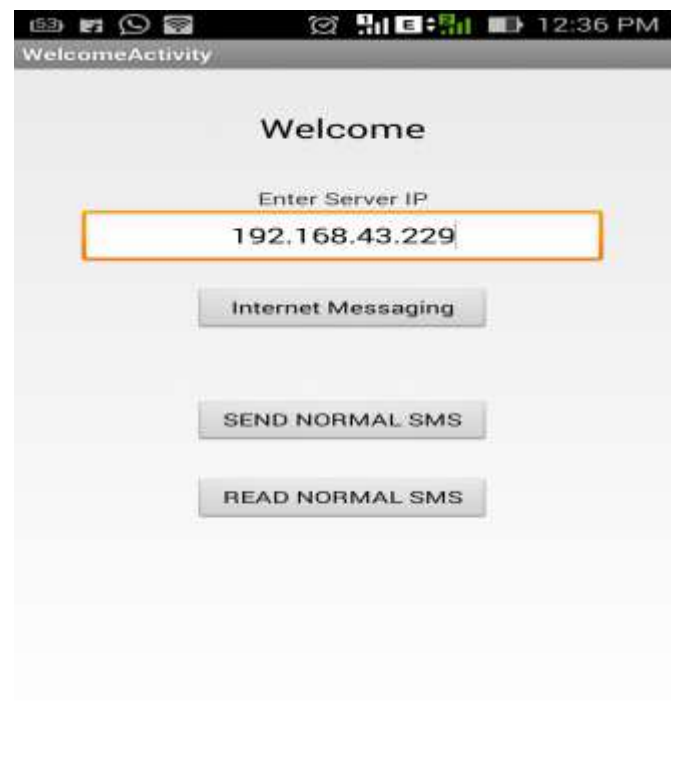


Figure 4 Server Connection Window



Figure 5 User Registration Window



Figure 7 Main Menu Window

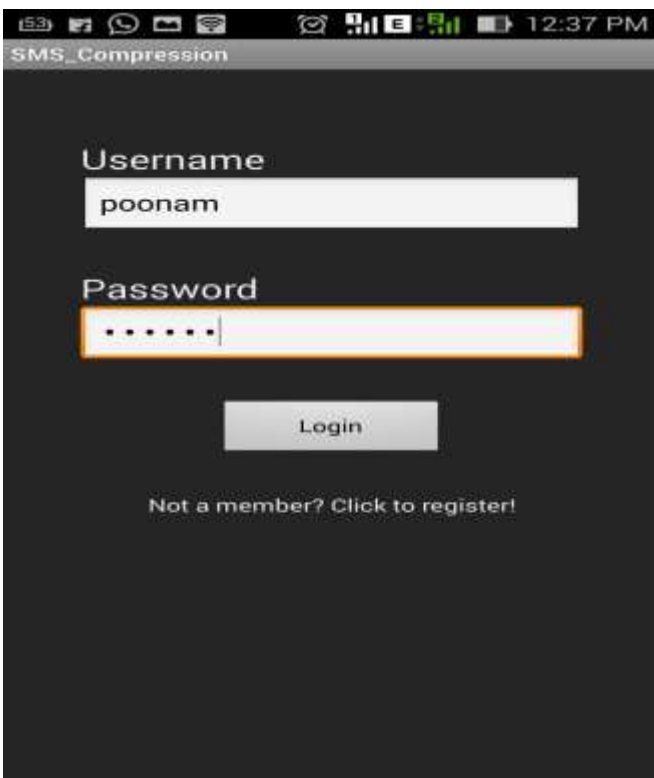


Figure 6 Login Window

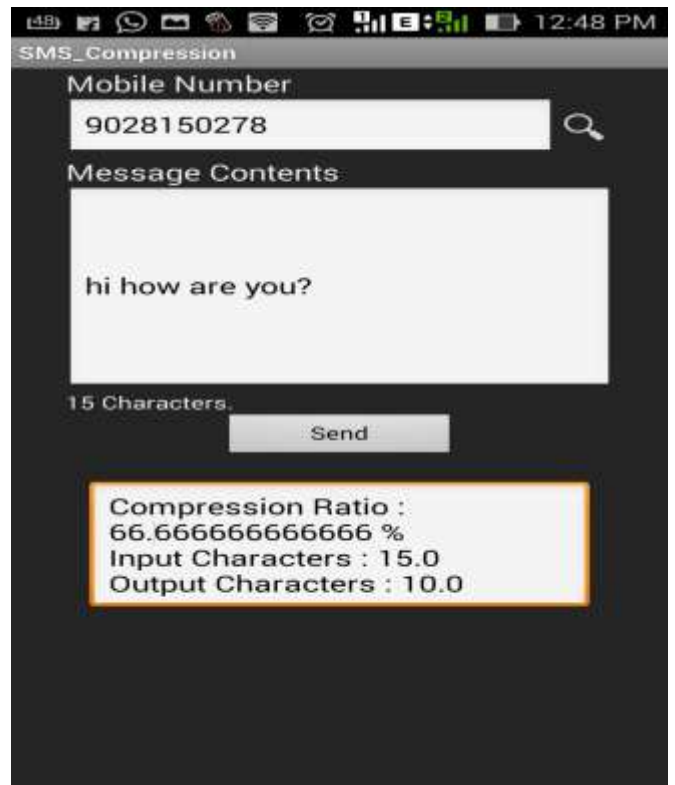


Figure 8 Send Message Window



Figure 9 Inbox Window



Figure 10 Read Message Window.

## 7. Conclusion

In today's world, many projects and applications are being developed to overcome boundaries in text messages and instant messaging. While incredible progress has been made in the field of mobile communication, support for increasing various end user demands remains a concern. Various advanced techniques are necessary in order to meet the challenges of business. "Message Communication A New Approach" comes up with a new model which can revolutionize the way messaging has been done between individuals as well as in a group. It can help messaging

become faster and economical and thus can made a good boost in mobile industry.

## 8. Acknowledgment

I express heartiest gratitude to Prof. P. P. Rokade, Prof. S. R. Durugkar, Prof. I. R. Shaikh of S.N.D COE & RC, Yeola For guidance and encouragement to write this technical paper.

## References

- [1] D.R. Kalbande, Dr.G.T. Thampi, Manish P. Mathai and Sumiran Shah "Zip it up SMS- A path breaking model in the field of mobile messaging" in IEEE 978-1-4244-5540-9/10 in 2010
- [2] Mohammed Al-laham & Ibrahiem M. M. El Emary "Comparative study between various algorithms of data compression techniques" in Proceedings of the World Congress on Engineering and Computer Science 2007, WCECS 2007, October 24-26, 2007, San Francisco, USA
- [3] S.R. Kodituwakku and U. S. Amarasinghe "Comparison of lossless data compression algorithms for text data" in Indian Journal of Computer Science and Engineering, Vol 1 No 4 416-425 ISSN : 0976-5166
- [4] Mamta Sharma "Compression Using Huffman Coding" in International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010
- [5] <http://en.wikipedia.org/wiki/WhatsApp>