IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 06, Issue 11 (November. 2016), ||V1|| PP 72-75

Improving the bandwidth efficiency for sending English alphabets using prefix codes

Mahmoud Z. Alja'fari,
Communication Engineering Department,Faculty Of Engineering Technology, AlBalqa Applied University
Amman-Jordan

Abstract:- This paper presents a new data structure and codes based on Huffman encoding that used to send English alphabet text, by adding the most repetitive words in the English language in order to decrease the total number of bits and then maximize the bandwidth efficiency.

Keywords: *bandwidth, data rate, Huffman, prefix.*

I. INTRODUCTION

A prefix code is a type of code system (typically a variable-length code) distinguished by its possession of the “prefix property”, which requires that there is no whole code word in the system that is a prefix (initial segment) of any other code word in the system.
Every message encoded by a prefix-free code is uniquely decodable. Since no code word is a prefix of any other we can always find the first code word in a message, peel it off, and continue decoding [1, 7].

1.1 Huffman encoding

Huffman developed a nice greedy algorithm for solving this problem and producing a minimum cost (optimum) prefix code. The code that it produces is called a Huffman code. Huffman coding is a popular method that satisfies the prefix property for compressing data with variable-length codes [1, 5]. Given a set of data symbols (an alphabet) and their frequencies of occurrence (or, equivalently, their probabilities), the method constructs a set of variable-length code words with the shortest average length and assigns them to the symbols. Huffman coding serves as the basis for several applications implemented on popular platforms. The Huffman encoding algorithm starts by constructing a list of all the alphabet symbols in descending order of their probabilities. It then constructs, from the bottom up, a binary tree with a symbol at every leaf. This is done in steps, where at each step two symbols with the smallest probabilities are selected, added to the top of the partial tree, deleted from the list, and replaced with an auxiliary symbol representing the two original symbols. When the list is reduced to just one auxiliary symbol (representing the entire alphabet), the tree is complete. The tree is then traversed to determine the code words of the symbols [2].

2. English alphabet codes:

In fig. 1, we can see the English alphabet codes [3] and we can see that the codes are prefix so we will use this property to add the most repetitive English words to this structure.

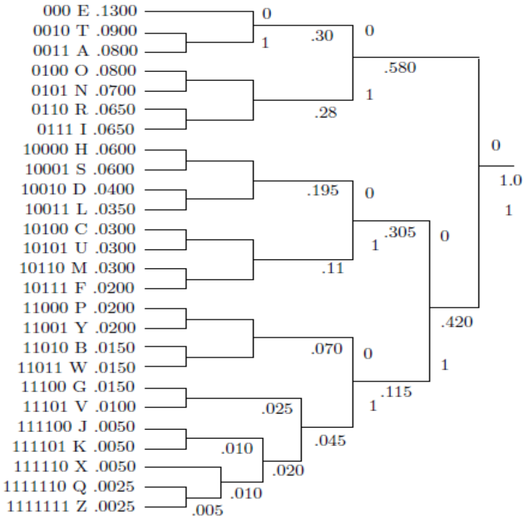

figure.1: show the English alphabet codes

These codes are made by Huffman encoder technique that will guarantee a prefix property and as small as possible code lengths [6, 8].

3. The new structure for sending English alphabets:

3.1 The words frequency

Now by adding the most repetitive words in the English language that are based on distillation of the Google books data [4] and they are:

{The, and, that, for, was, with, not, this, are, his, from, which, but, have, had, and they}

The choosing of and the sort of the words is based on the maximum gain we have in preserving the number of data bits we need to send a text of English language message.

3.2 The newly added codes

By a prefix property, we can give these words a code that will not interfere with the other codes and to do this, I change just one code in the Huffman tree for English alphabet, and I choose the alphabet (I) because of its small frequency and number of bits to change it by adding just one bit in the end so it will be like this (01111) [1].

Now the field of (01110) is open and not interferes with the other codes in the Huffman tree.

The new codes for the most repetitive words in English language and the alphabet (I) can be seen in table.1

Table.1

I	01111
The	011100000
and	011100001
That	011100010
For	011100011
Was	011100100
With	011100101
Not	011100110
This	011100111
Are	011101000
His	011101001
From	011101010
Which	011101011
But	011101100
Have	011101101
Had	011101110
they	011101111

Now we can see in table.2 the code length differences between the ordinary English language alphabets codes and the new coding scheme proposed.

Table.2

letter	Old code length in bits	New code length	The difference
I	4	5	+1
The	12	9	-3
and	13	9	-4
That	17	9	-8
For	13	9	-4
Was	14	9	-5
With	18	9	-9
Not	12	9	-3
This	18	9	-9
Are	11	9	-2
His	14	9	-5
From	18	9	-9
Which	24	9	-15
But	14	9	-5

Have	17	9	-8
Had	14	9	-5
they	17	9	-8

We can observe the following notes from table.2:
A- The positive sign means that the number of bits is increased and the negative sign means that the number of bits is decreased.
B- The only increase in the number of bits per code word is in the letter (I) by just one bit to make the field 01110 available.
C- We decrease the number of bits per code word in all words between (2-to-15) bits.
3.3 Measuring tool:
Now we can measure the differences between the two methods by counting only the letter (I) on the English text (except in the words: with, this, his, and which) by replacing it with +1 bit and the words in table .2 by the difference number in negative sign.
Example: the difference in the number of bits in the text *"The frequency of letters in a text has been studied for use in Huffman's encoder"* between the two methods in the next sentence is:
Each I in words (in, studied, and in) means +1.The values for each special word are as follow: The = - 3 bits, for = -4 bits.
Then the final difference will be:
(+1+1+1-3-4) = -4 bits.
This means we decrease the number of bits in sending the last sentence by four bits.
Now let use this method on a larger text as:
"Water is a transparent and nearly colorless chemical substance that is the main constituent of Earth's streams, lakes, and oceans, and the fluids of most living organisms. Its chemical formula is H_2O, meaning that its molecule contains one oxygen and two hydrogen atoms, that are connected by covalent bonds. Water strictly refers to the liquid state of that substance, that prevails at standard ambient temperature and pressure, but it often refers also to its solid state (ice) or its gaseous state (steam or water vapor)."
The number of the letter I in the text is: 24 which mean an increase of 24 bits, now the decrease in the number of bits because of the words in the table is -76 bits. Then the final result means we have a decrease in the number of sending bits by 54 bits.
Fig.2 gives a comparison in the number of bits decreased for 10 text samples contains just 10 lines of words:

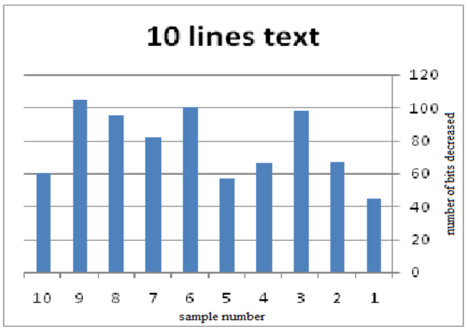

figure.2 : shows the performance for 10 lines text.

Fig.3 gives a comparison in the number of bits decreased for 10 text samples contains just one page:

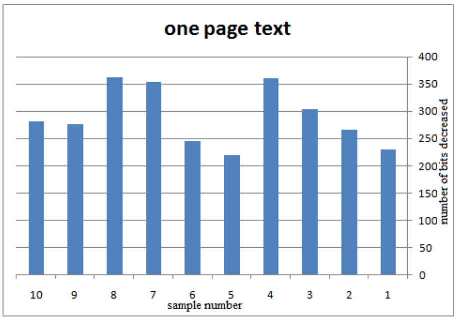

figure.3: shows the performance for one page text.

II. CONCLUSION

The new structure gives us an enhancement on the bandwidth efficiency on every text size so we can use it to maximize the number of data that can be sent at the same time.

REFERENCES:

[1] S.Hykin, *Communication systems* 4th edition John Wiley & sons 2001.

[2] A.Shahbahrami, R.Bahrampour, *Evaluation of Huffman and Arithmetic Algorithms for Multimedia Compression Standards.* (International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.1, No.4, August 2011.)

[3] R. Togneri. deSilva, *Fundamentals of information theory and coding design* (Taylor & Francis e-Library, 2006).

[4] *http://norvig.com/mayzner.html*

[5] S. ling C. xing , *Coding Theory* (Cambridge University Press 2004)

[6] Mordecai. j. Golin, Claire Mathieu, *Huffman coding with letter costs: a linear-time approximation scheme*. (2012 Society for Industrial and Applied Mathematics Vol. 41, No. 3, pp. 684–713)

[7] Luciano G. Buratto , *A Comparison of Huffman codes across languages* (April 19, 2002).

[8] S.Porwal, Y.Chaudhary, *Data Compression Methodologies for Lossless Data and Comparison between Algorithms.* (International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 2, Issue 2, March 2013).

[9] M.Sharma, *Compression Using Huffman Coding*. (IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010).