DOI: 10.21917/ijct.2020.0325

CONSTRUCTING LOW-DENSITY PARITY-CHECK CODES IN DIGITAL COMMUNICATION SYSTEM

P. Dhivya Lakshmi

Department of Electronics and Communication Engineering, Government College of Technology, Coimbatore, India

Abstract

In data communications, errors occur when the data travels through different communication channels. Their errors have to be reduced to obtain efficient Bit Error rate (BER). Over unreliable or noisy communication channels, errors in data can be controlled effectively using Error-Correcting Code (ECC). NI USRP (National Instruments Universal Software Radio Hardware) 2954R support 5G with multi-radio cooperation using Multiple- Input, Multiple Output (MIMO) techniques. The frequency of NI USRP 2954R ranges from 10 MHz to 6GHz with 160MHz bandwidth. In a software radio by mean of software upgrade a new technology can be easily implemented whereas the functionality is defined in software itself. Thus, in 4G communication, a Forward Error Correction Code (FEC) like LDPC is used. In this paper LDPC codes with Binary Phase Shift Keying (BPSK) modulation is simulated with NI lab-view and the performance of the LDPC code for the dimension (100,50,3) is analyzed. The experimental results demonstrate that as a Signal to Noise Ratio (SNR) increases, BER decreases.

Keywords:

LDPC code, BER, AWGN, Modulation

1. INTRODUCTION

Communication has got many applications in which along the communication channel the messages are encoded at the transmitter end and then decoded at the receiver end. In digital communication systems, there are two different ways commonly used to reduce errors caused by a noisy channel. One is called Automatic Repeat Request (ARQ). With ARQ, a receiver is able to detect if a message was received in error. If an error occurred, the receiver sends a request back to the transmitter to repeat the message that was in error. This method of error correction requires a two-way channel and is not a feasible method for one-way systems such as digital video broadcasting. During the message transfer, error detection and correction method is essential where the data might get corrupted in the communication channel due to lots of disturbances [1] [6].

An error can be both detected and corrected by the Forward error correction (FEC) approach. In which there are two types of Forward Error correcting codes namely Block codes and Convolution codes. Generally for error correction in block codes redundant bits are added to the fixed-sized blocks of bits which is achieved by dividing the original message. The message comprises of data streams of arbitrary length and parity symbols are generated by the sliding application of a Boolean function to the data stream in convolutional codes [3] [8].

Low Density Parity Codes are good error correcting codes which can achieve the better performance near to Shannon limit with the practical decoding complexity like turbo codes on an Additive White Gaussian Noise (AWGN) channel [4]. MIMO-LDPC-TD helps to achieve the good error rate performance with reduced decoding complexity on a flat Rayleigh fading channel. High encoding complexity is the major drawback of LDPC codes. Complexity means number of mathematical operations required per bit. Several modulation techniques are used along the communication technique [5]. In this work LDPC codes have been implemented and analyzed in Phase Shift Keying (PSK) modulation which effectively decreases the Bit Error Rate (BER) and increases to SNR value.

2. OVERVIEW OF SDR AND LABVIEW

The Software Defined Radio (SDR) is a design paradigm for wireless communications devices. Its creator, Joseph Mitola, defined the term in the early90s as an identifier of a class of radios that could be reprogrammed and reconfigured through software. SDR system is based upon programmable dedicated hardware and associated control software. It enhances the functionality through software that would automatically reconfigure the radio parameters and interacts with the network using it. The physical components of Software Defined Radio are only an antenna and an Analog Digital Converter (ADC) on the receiver side.

Likewise, the transmitter would have a Digital Analog Converter (DAC) and a transmitting antenna. The rest of the functions would be handled by reprogrammable processors. The NI USRP (Universal Software Radio Hardware) connects to a host PC to act as a software-defined radio. USRP provides a solution for integrated hardware and software by rapidly prototyping high-performance wireless communication systems.

Labview is used to create and develop programs using a graphical design platform for a visual programming language. In graphical Programming module, it has three components- block diagram, front panel, and connector pane. The graphical source code is there in the back panel or a block diagram which is used to perform operations on supply data and also to indicators and controls.

Controls and indicators as input and output is used to build the front panel. The connector pane defines the inputs and outputs wired to the VI so that it can be used as a sub VI. Applications in Labview are usually designed using well-known architectures, known as design patterns. Labview programming is undertaken on the block diagram of the Virtual Instrument. G programming is a technical name for the Labview programming language but nowadays the term is largely unused and the name Labview has become to mean the language as well as the software itself.

3. SYSTEM MODEL

The basic elements of a digital communication system can be visualized as follows in which Input Information which is to be transmitted is obtained from the source. Messages which are to be transmitted to the receiving destination are produced by the

source. All these messages in digital source are considered as the sequences of symbols which is taken from a finite alphabet [2].

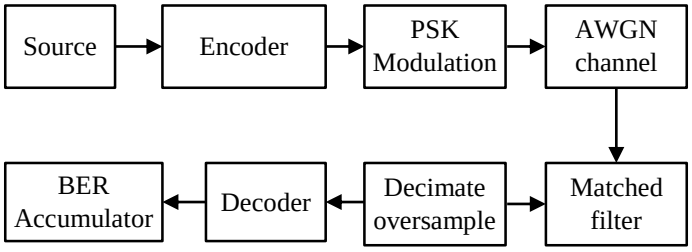

Fig.1. System model

The source data as input which is chosen by the transmitter produces an associated signal. It ensures a reliable transmission across the noisy channel. By the addition of structured redundancy in the message and the encryption is done which encodes the message so that unauthorized parties cannot discern the real information content from the message. Without changing the parameters of the original signal, the strength of a signal is increased by the modulation [9]. In order to transmit the signal from the source to the destination, Channel acts as the medium. The inverse operation which is done by the transmitter is ordinarily performed by the receiver. From the received signal, the original message is reconstructed. The message is intended to the system or person which is the destination.

3.1 LDPC CODE

LDPC codes were developed in 1960 by Robert G. Gallager at MIT. With recent advances in parallel computing power, LDPC codes have been re-discovered and studied. They are used in many high speed communication standards such as digital video broadcasting, WiMAX, 4G wireless systems, to name a few. Depending on the low-density parity check matrices *H*, LDPC codes are considered as the linear block codes, i.e. in the matrix *H*, the number of zeros is greater than the number of non-zero elements [7].

In LDPC code the code rate *R* can be calculated by *N*-*M*/*N* where *N* is the number of columns and *M* is the number of rows in matrix *H*. The number of columns *N* is denoted as code length. The number of rows *M* is denoted as the number of parity check equations. LDPC codes can be classified into two type's namely regular or irregular code based on the non-zero elements of the matrix [10]. The non-uniform distribution of non-zero elements on columns and/or rows is mainly due to the irregularity in LDPC code. A parity check matrix *H* of size 4×6 of a regular LDPC code is given by an example as follows:

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (1)$$

Tanner graph also known as the bipartite graph composed of two types of nodes. One is the variable nodes which represents the symbols of the codeword and second is the parity nodes which represents the parity control equations. In the parity check matrix two types of nodes are connected by branches based on the non-zero elements. The Fig.2 represents the tanner graph corresponds to matrix *H*.

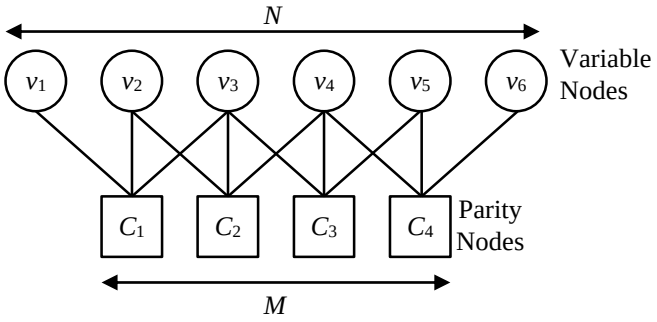

Fig.2. Tanner Graph 4×6 of regular LDPC code

3.2 ENCODING

There are several algorithms available for the encoding of *x*. With sparse LU decomposition method, this demonstration creates parity check vector bases. The algorithm strives to generates an upper triangular and lower triangular matrix which is as sparse matrix. The algorithm is a sparse matrix method and probabilistic inference algorithm and consists of the following basic steps:

Step 1: Partition *H* into an invertible *M*×*M* left part *A*, and an *M*×*N* right part *B*.

Step 2: The codeword *x* is partitioned into *M* check bits *c* and *N*-*M* source bits *s*.

Step 3: To find *Uc* = *y*, set *z* = *Bs* and row reduce *Ac* = *z*, where U is the upper triangular matrix.

Step 4: Record the reduction as the solution to *Ly* = *z*. A vector [*x*|*c*] is obtained by combining the check bit vector *c* and the codeword *x* that can be transmitted to the receiver for decoding.

Labview parameters of LDPC encoder is described as follows. The incoming bit stream to be mapped to LDPC symbols are specified by Input bit stream. The internal state of the encoder is cleared by Reset (*T*). Error in (no error) is used to describe the Error conditions that occur before this node runs. This input provides standard error in functionality. There are three check nodes and 5 variable nodes. Hence, the graph can be expressed by the 3×5 *H* matrix.

$$H = \begin{matrix} \text{Variable Node} \\ \begin{bmatrix} 1 & (1) & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & (1) & 1 \end{bmatrix} \\ \text{Check Nodes} \end{matrix} \quad (2)$$

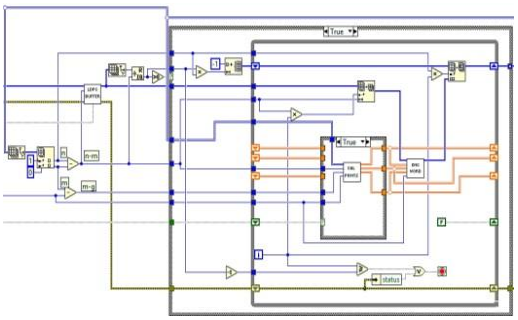

Fig.3. Labview implementation of LDPC encoder

The preprocessed parity check matrix that is used by the encoder for encoding is specified by Parity check matrix out. This parity check matrix must be provided to the decoder. Output bit stream returns the encoded data bits. Generated parity bits at the end of the output bit stream array by default are placed by the encoder. Hence, the encoded word format is [message bits | parity bits]. Error out consists of appropriate error information. Thus the output provides standard functionality of error out.

3.3 DECODING

The bit-flipping decoding algorithm is the main idea of simple LDPC decoders. Consider a received codeword is x = {1 0 1 1 1}. This means that there is an error, because the check nodes are not all zeros (i.e., H_x = 0) for x = {1 1 1 1 1}. Compare the error free codeword x = {1 0 1 1 1} with x = {1 1 1 1 1}. Thus it shows that there is an error on the second bit of the received codeword x.

The decoding algorithm for LDPC can be explained with the following steps [3].

Step 1: Compute each check node sums with the given codeword. (i.e., x = {1 1 1 1 1}).

Step 2: Count the nonzero parity checksums for each variable node.

Step 3: The variable node bit having the largest number of nonzero parity checksums is flipped.

Step 4: Repeat step 1 through step 3 until all parity checksums are zero.

In Labview, the LDPC serial decoding schedule is done based on a serial update of symbol node messages. Thus shuffling of the flooding schedule can be considered from this update. Instead of sending all messages from symbol nodes to check nodes and vice versa. We go through the check nodes in some order as done in the flooding schedule. For each node we send all messages in to the node and then all messages out from the node.

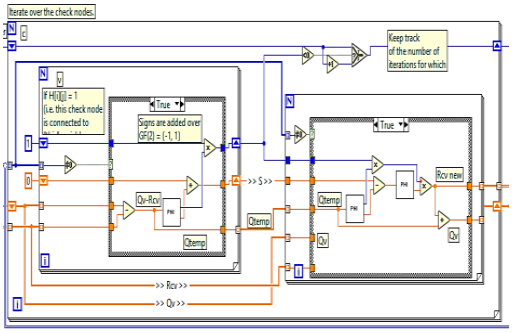

Fig.4. Labview implementation of the LDPC decoder

Parity check matrix specifies the parity check matrix used by the encoder for encoding. Likelihoods specify the likelihoods of the received symbols. In the iterative decoding process Maximum number of iterations are specified. When the number of iterations exceeds the value of the maximum number of iterations parameter or if the decoder satisfies other conditions then the decoder stops the iterating process. The internal state of the decoder is cleared using Reset. Error conditions that occur before this node runs is specified by error in (no error). This input provides standard error in functionality. Output bit stream returns the decoded data bit stream. Error out holds the error information. Thus the output provides the effective standard error out functionality. Further to increase the Tanner graph of decoding with decoded bits accuracy, the algorithms can be iterated.

4. PHASE SHIFT KEYING MODULATION

Phase-Shift Keying (PSK) is the constant-amplitude digital modulation and it is an angle modulated type. The simplest form of PSK is the Binary Phase-Shift Keying (BPSK), where N = 1 and M=2. Generally two phases for the carrier with BPSK are possible. Logic 1 and logic 0 is used to represent the phases. The state is changed from a 1 to a 0 or from a 0 to a 1 by the input digital signal, the output carrier phase shifts between two angles that are separated by 180°. Phase Reversal Keying (PRK) and Biphase modulation are the other names for BPSK. They are in the form of square-wave modulation of a Continuous Wave (CW) signal.

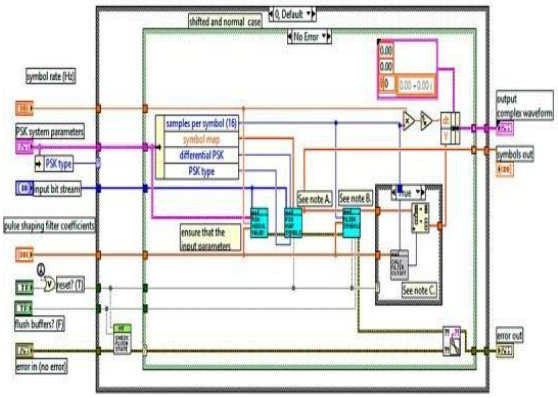

Fig.5. Labview implementation of PSK modulation

In labview MT modulate PSK VI receives a sequence of data bits which performs PSK modulation, and returns the modulated complex baseband waveform in the output complex waveform parameter.

Binary Input	Output phase
Logic 0	180˚
Logic 1	0˚

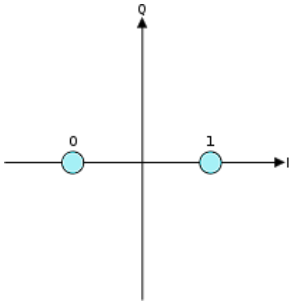

Fig.6. Truth table and constellation graph

The sequence of information bits to be modulated is specified by Input bit stream. PSK system parameter specifies parameter values defining the PSK system such as samples per symbol, symbol map, PSK type. Symbol rate specifies the desired symbol rate, in Hertz (Hz) where its default value is 1. An ordered array containing the desired pulse-shaping coefficients is specified by Pulse shaping filter coefficients. The VI continues modulating using the previous iteration states is specified by Reset. Flush

buffers specify whether samples are forced out from the modulated waveform that is affected by the FIR pulse-shaping filter delay. Output complex waveform returns the PSK-modulated complex baseband waveform data. The array of mapped symbols before pulse shaping is applied is returned by the Symbols out. The array represents the complex value of each mapped symbol. Error out contains error information. This output provides standard error out functionality.

5. RESULTS AND DISCUSSIONS

There are several parameters involved in estimating the Bit Error Rate (BER) and Signal to Noise Ratio (SNR). In which seed in value is estimated to be not more than 4. Thus, the above given table provides Performance analysis of output with and without LDPC code. Parameters involved: seed in, pseudo noise sequence order, maximum iteration, *n*, *m*, *j*, E_b/N_o samples and maximum iteration for decoder.

Table.1. Parametric Analysis with and without LDPC

Parameters	No LDPC	With LDPC
n	100	200
m	50	100
Seedin	-1	-2
E_b/N_o	5.2	6.18
PN sequence order	9	19
Max iterations	30	30
BER	5.1077m	0.04450m

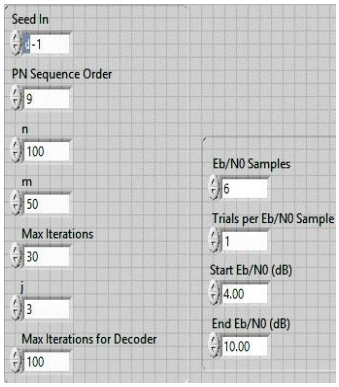

Fig.6. Output of the without LDPC in PSK modulation system

From the Fig.6, it is clear that various parameters of the LDPC VI in Labview are varied and analyzed. The overall block also includes the theoretical LDPC sub VI which results with the Shannon's limit theorem. Thus, BER are achieved both in milli and micro units. From the Fig.7 BER graph is analyzed accurately which is similar to algorithm specified. AWGN usually generates zero-mean complex Additive White Gaussian Noise (AWGN) with uniform power spectral density and adds it to the complex baseband modulated waveform.

From the Fig.8 and Fig.11 the constellation graph for BPSK modulation can be achieved along in phase and quadrature phase. *N* and *M* values are varied and various BER values are achieved for both with and without LDPC codes.

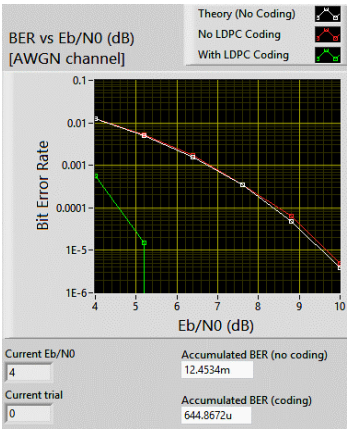

Fig.7. BER vs, E_b/N_o without LDPC in PSK modulation

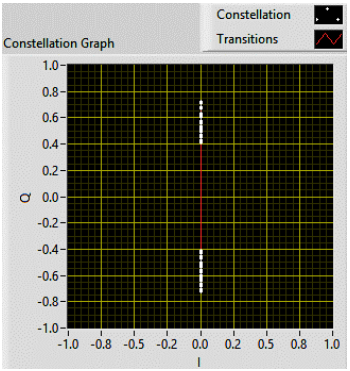

Fig.8. PSK Modulation Constellation Graph

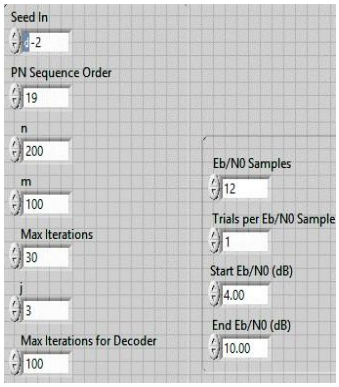

Fig.9. Output of the with LDPC in PSK modulation system

Iteration values may vary depending upon the other parameters. In which *n*, *m*, *j* values determine the code size of LDPC. Bit error Rate, sometimes BER is the most fundamental measure of system performance. That is, it is a measure of how well bits are transferred end-to-end. While this performance is affected by factors such as signal-to-noise and distortion, ultimately it is the ability to receive information error-free that defines the quality of the link BER is the number of bits received in error, divided by the total number of bits received. It is the percentage of bits that have errors relative to the total number of bits received in a transmission, usually expressed as ten to a negative power. For example, a transmission might have a BER of 10^{-5}, meaning that on average, and out of 100,000 bits transmitted only 1 bit exhibit an error. The BER is an indication

of how often a packet or other data unit has to be retransmitted because of an error. If the BER is higher than typically expected for the system, it may indicate that a slower data rate would actually improve overall transmission time for a given amount of transmitted data since the BER might be reduced, lowering the number of packets that had to be present.

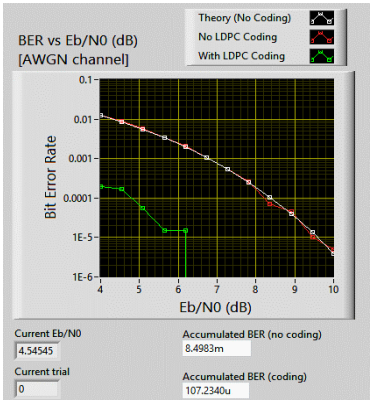

Fig.10. BER vs. E_b/N_o with LDPC in PSK modulation

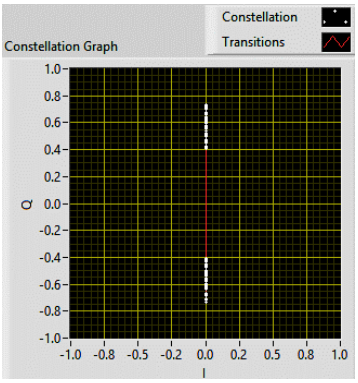

Fig.11. PSK modulation constellation graph (with LDPC)

5.1 INFERENCE

All these parameters are varied and effective BER is obtained with an increase in its SNR. Constellation graph is the representation of the signal modulation by the Phase Shift Keying modulation technique. Thus, it is clear from both the output that on increasing the pseudo noise sequence order and E_b/N_o samples the BER is decreased. When SNR increases, the noise power level decrease. LDPC codes play a vital role in substituting parity bits to decrease the error rate. So, the BER must also decrease in BPSK modulation.

$$BER = 0.5 \times erfc\sqrt{\frac{E_b}{N_o}} \tag{3}$$

6. CONCLUSION

This work validates the scalability of LDPC algorithm by deploying encoding and decoding cores in PSK modulation. The encoder and decoders are discussed with LPDC coding. The LDPC decoding may be best understood by the sum product algorithm and bit flipping algorithm showing the LDPC code can be implemented easily. Thus, LDPC codes give the better result in decreasing BER and increasing in signal to noise ratio. To determine the BER against different values of SNR, computer simulation is done. For various SNR, bit error rate is calculated. Codes constructed by various methods were simulated. The modulation scheme used is BPSK and computer simulation was done for various rates. The code length was fixed to 1024 bits. Performance and resource utilization comparison with varying number of parameters of the LDPC encoder and decoder is analyzed. The simulation done for different rates are showing that the short length LDPC codes are performing excellent, and the code perform better for all rates either low or high. The standard LDPC achieves an overall SNR in the NI Labview.

REFERENCES

[1] R.G. Gallager, "Low-Density Parity-Check Codes", *IRE Transactions on Information Theory*, Vol. 8, No. 1, pp. 21-28, 1962.

[2] R. Tanner, "A Recursive Approach to Low Complexity Codes", *IEEE Transactions on Information Theory*, Vol. 27, No. 5, pp. 533-547, 1981.

[3] C.E. Shannon, "A Mathematical Theory of Communication", *Bell System Technology Journal*, Vol. 27, pp. 379-385, 1948.

[4] T.J. Richardson and R.L Urbanke, "Efficient Encoding of Low-Density Parity-Check Codes", *IEEE Transaction of Information Theory*, Vol.47, pp. 638-656, 2001.

[5] M. Kakooti and J.R. Cavallaro, "Semi- Parallel Reconfigure Architecture for Real-Time LDPC Coding", *Proceedings of International Conference on Information Technology: Coding and Computing*, pp. 579-585, 2004.

[6] B. M. J. Leiner, "LDPC Codes-A Brief Tutorial", Available at: http://users.cecs.anu.edu.au/~Gerard.Borg/anu/tutorials/simple_intro_ldpc_decoding_Leiner.pdf, Accessed on 2005.

[7] H. Kee, S. Mhaske, D. Uliana, A. Arnesen, N. Petersen, T. L. Riche, D. Blasig and T. Ly, "Rapid and High-Level Constraint-Driven Prototyping using LabVIEW FPGA", *Proceedings of IEEE Global Conference on Signal and Information* Processing, pp. 1-6, 2014.

[8] Y. Fang, P. Chen, G. Cai, F.C. Lau, S.C. Liew and H. Han, "Outage-Limit-Approaching Channel Coding for Future Wireless Communications: Root-Photograph Low-Density Parity-Check Codes", *IEEE Vehicular Technology Magazine*, Vol. 14, No. 2, pp. 85-93, 2019.

[9] K.S. Andrews, D. Divsalar, D. Dolinar, J. Hamkins, C.R. Jones and F. Pollar, "The Development of Turbo and LDPC Codes for Deep-Space Applications", *Proceedings of the IEEE*, Vol. 95, pp. 2142-2156, 2007.

[10] Z. Wu, M. Zhao and S. Wang, "An Efficient Encoding Method for Spatially Coupled Low-Density Parity-Check Codes", *Proceedings of IEEE 5th International Conference on Computer and Communications*, pp. 1505-1509, 2019.