Data Field Transformation from Ethernet Frame

Vaclav Oujezsky, Vladislav Skorpil
Department of Telecommunications,
Brno University of Technology, Czech Republic

Abstract—

This article deals with the possibility and the intention of transformation and data filed re-reconstruction from Ethernet frame. The proposed system is going to use implemented transformation method based on hardware solution.

Keywords— Ethernet, transformation, compression, Karhunen-Loeve, fpga, vhdl.

I. INTRODUCTION

IN some cases different core equipment with different line speed, bandwidth and data processing for interconnection is used. The subject of this article introduction are current commonly used methods of data compression in Wide Area Network (WAN) networks and also possibilities of research and implementation of the transformation methods of data transmitted over the network so that are created new protocols for transfer data with use of knowing mathematical or logical operations. For such development hardware implementation can be utilized and developed on Field Programmable Gate Array (FPGA). This article is going to describe steps to attain realization.

II. ANALYSIS OF THE CURRENT STATE

A part of the research is to analyse the current state of the use of compression e.q. substitution in data traffic processing regarding network element. Provisional results are indicating ensuing status:

- A proprietary: Although several compression algorithms are supported by many networks equipment, they are proprietary and not necessarily interoperable.
- Compression ratios: Can realize up to a 6:1 or is conservatively set up.
- Software compression solutions.

The most used types of compression methods in core networks equipment (router, switch and computer):

- HC – Header compression (ROHC, CRTP, ECRTP, IPHC).
- Based on predictor.
- Microsoft Point-to-Point Compression (MPCC)
- Based on LZW (for example LZS Limpel Zif Stac.)

In many cases it is not acceptable to use compression. Limited to point-to-point connection, re-taking flow, encryption method. For these reasons, the article is an effort to design a method for compression on FPGA that would accommodate the solution to issues mentioned above and expand those methods in use.

III. GLOBAL VIEW

For research purposes it has been compiled data processing algorithm on L2 OSI layer on NetCOPE [1] platform at the laboratory of the transport network of project SIX [2]. It is intended to substitute data primary in data field for Ethernet frame, not affect the whole system processing for delivering frame through data line ignoring the highest layer processing such as for example TCP flagging on layer 4. It is effort to implement and simplify existing transformation based on principal component analysis (PCA) [3] known as Karhunen-Loeve transformation (KLT) [4] mixed with CRC recompilations and examine the effect of transformation will bring with implementation on FPGA.

A. Computing method

Consider PCA and KLT by [3] [4] simply the flow of a number of field F "data" when all the "data" are the same matrix *A* of Bit or Byte size n = Width x Height of 0 1. The first incoming and continuing the "data" always construct a matrix with the same number of rows and columns. When each element of matrix is stacked in a row, vector x_i is created x_i . The main idea of the KLT for the intended purpose is to find the vectors that best describe the distribution of the "data" throughout the space. The found vectors define the subspace of dimension *n*, which we call the area of "data" with linear combination of the original vector and forms the basis. These vectors are vectors of the covariance matrix corresponding to the original array "data". Calculating the average of the first and the following field data:

$$M = \frac{1}{F}\sum_{i=1}^{F} x_i \quad (1)$$

M as is meaning the average, x meaning vector from "data" matrix. Then we subtract from each vector an average vector:

$$\bar{x}_I = x_i - M \quad (2)$$

Thus obtained coefficients are arranged in a matrix labeled as $\overline{Q}$. Are being searched vectors referred to as eigenvectors of the covariance matrix Cm' , which is obtained as follows:

$$Cm^T = Q \times Q^T ; Cm = \frac{1}{(N-1)}\sum_{i=0}^{N-1} \bar{x}_l \times x_i^T \quad (3)$$

where N is the number of vectors. Then diophantic equation is calculated:

$$\text{left sides} \quad Cm' \times D' = \lambda \times D' \quad \text{right sides} \quad (4)$$

Wherein D' represents the matrix of eigenvectors which are arranged in rows and λ represent eigenvalues of the covariance matrix Cm' . So after solving (3) (4) are obtained eigenvalues and eigenvectors of a matrix Cm . The next step are rearranged eigenvectors of the matrix D' by the corresponding eigenvalues from the largest to the smallest using the shift algorithm and normalized to unit size for each vector x_i from matrix D' . The transformation is performed according to the relationship:

$$\overline{Q} = Q \vdots \tilde{x}_l \overset{yields}{\rightarrow} \bar{x}_l \times D'^T \quad (5)$$

where $\tilde{x}_l$ is new subset of transformed vector x_i in each row of matrix Q are filling up to matrix $\overline{Q}$.

When making rollback of transformation, algorithms based on relationship between vector base and transformed vector is used:

$$F = A \vdots \ddot{x}_l = \ddot{x}_l + (M) \quad (6)$$

$$\text{Where} \quad \ddot{x}_l[t] \overset{def}{=} \sum_{i=0}^{d-1} D'[d_i]\tilde{x}_l \quad (7)$$

1) The difference in data processing in a modified KLT based on the ideas presented in [4] is the replacement of a collection of "data" in the course of processing only the previous and next field "data" on the receiver and the transmitter side. Then it is not necessary to transmit the data field own base vectors, only D'^T which replaces the original field "data". From our idea then, this solution leads to a reconstruction for our use, because size of matrix D'^T remains consistent when resizing a vector x_i . It may be considered with error delta matrix, CRC self-correcting and must be normalized each coefficients to whole numbers with descriptive headings, because we can describe only definite group of number finite group. It can be used for this case as polynomial expansion. Finally coefficient represented by binary in "data" frame would be less than original filling frames in vector display. The difficulty of calculating and processing in time *(t)* must be verified by physically implementing algorithm. We continue with analyzing the implementation.

B. Logical programmable method and hardware description

2) First, method have effort to taken to the software Matlab [5] processing and then the following problem is solved: how it will be implemented to the FPGA with making it synthesizable and pipelining it. It can be described by Matlab and must be solved on FPGA. Mainly solution processing of Matlab function:

1. *Mean()* function for (1)
2. *Sort()* function for (5)
3. *Diag()* function for (4)
4. *Eig()* function for (4)
5. *Poly()* function of 2).
6. Matrix multiplication (7)-(1)

C. The considered first basic symbolic communication

PROG. 1. Matlab symbolic KLT process sender A, receiver B,

```
% set transceiver A, receiver B = static
% process sender A
input = [static,newdata];
% set data size
[~, N] = size(input)
% mean factor for each vector                    math(1)
```

```
M = mean(input,2) %if M > M-1 send it
% calculate the covariance matrix              math(3)
rinput = input - repmat(M,1,N)
C = 1 / (N-1) * rinput *input'
% solve the evectors and evalues               math(4)
[V, D] = eig(C)
% where det(C)= prod(D)
% extract and sort by vector
D = diag(D)
[item, index] = sort(-1*D)
V2 = V(:,index)
% KLT only processing
KLT = V2 * input %                    math(5)
% if is energy in the first vector of KLT
% process CRC and then
% send only V2 + KLT(1,:)to receiver B
% then process on receiver B
KLT2 = [KLT(1,:);zeros(1,N)]
receive = KLT2' * V2 %    + if „M“ math(6)
% process CRC and then
output = abs(round(receive))
% else ~
```

Input data “newdata” in Prog. 1 are tagged in accordance with eigenvalues and eigenvector Table I according to the following algorithm based on values of the four elemental matrixes 0 and 1:

TABLE I: MARKING DATA

Binary		Evector	Evalue
0#00			
00	0	#1001	6#0000
0#00			
01	1	#1001	2#0001
0#00			
10	2	#1001	1#0010
0#00			
11	3	#1001	7#0011
0#11	1		
00	2	#1001	5#1100
0#11	1		
01	3	#1001	3#1101
0#11	1		
10	4	#1001	0#1110
0#11	1		
11	5	#1001	4#1111
1#01			
00	4	#0110	0#0001
1#10			
00	8	#0110	1#0010
1#01			
11	7	#0110	3#1101
1#10	1		
11	1	#0110	0#1110
2#01			
01	5	(-0,7/0,7)	10#000(1,4)
2#10	1		
10	0	(-0,7/0,7)	9#00(1,4)0
3#01		(-0,7/-	11#(-0,7/-0,7/0,7/-
10	6	0,7)	0,7)
3#10		(-0,7/-	8(-0,7/-0,7/-
01	9	0,7)	0,7/0,7)

If we have vector x_i of data: v = […..11100011100111111111110001] it will mean according to Table 1 of the following table input sequence s = […7 8 4 4 2] for the field "newdata" Prog. 1. In this example, the number zero will be filled after decoding due to the fixed size of vector. The number is differentiated using the eigenvector and this table corrects power of new matrix. Then the receiver after doing Prog.1 replaces solved output in conformity with Tab.1. Ultimately, one can imagine that, if will be accepted fact that the subspace of the vector x_i can grow without limit, will be attained 2:1 of the rate of substitution. However, there is not an accurate result after step 6 of Prog. 1. In this case, there is studied the possibility of additionally introduce Cayley-Hamilton theorem [6] for Prog. 1 math (5) "KLT" and edit the aforementioned symbolic computation by mentioned polynomial structures and in the end to pass the least information necessary to reconstruct the field.

IV. HARDWARE CODE

For this purpose VHDL code is considered as the implementation programming language to the FPGA chip in platform NetCOPE. There must be solving the set up of each mathematical step as a proprietary logical block. Basic Fig. 1. showing involvement of lab. First is set up of the separation data from the field data of data frame block /1/.

In the case of capture data a simple finite state machine is used. In the case of detection of an incoming frame retrieves an array of data into memory. Details are not important for the purposes of article content. Blocks /2/ and /4/ are under development. Block /2/ contains the KLT process itself should contain the aforementioned functions *B.* 3) 1-6. Block /3/ compiles the entire frame incl. CRC determination and then the frame is sent. Block /4/ should provide normalization and parameterization of coefficients.

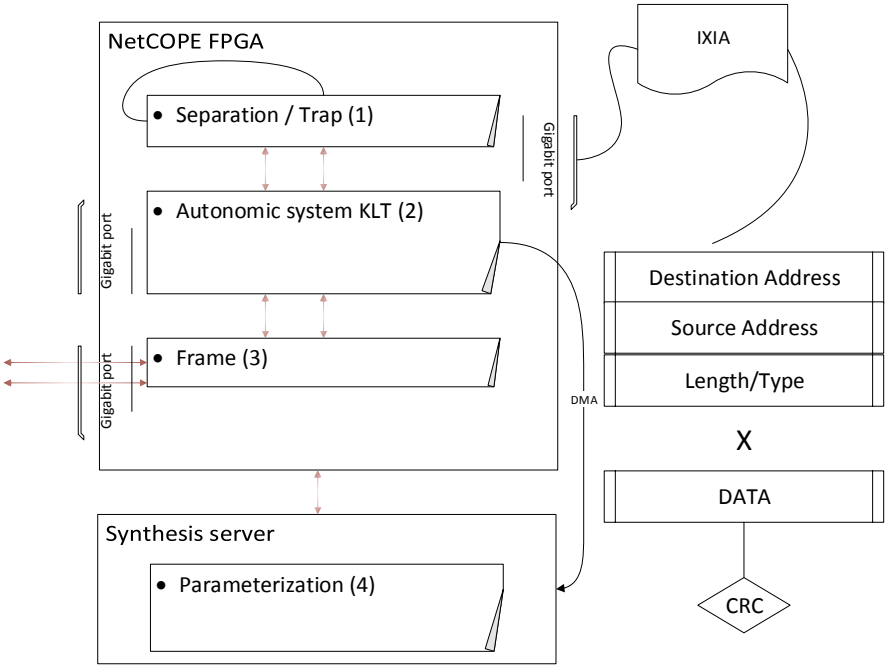

Fig. 1. The block diagram of capturing and processing data

V. CONCLUSION

The subject of this paper was introduction of the conducted research with the intention to develop the current compression method used in data networks of systems transformation. As one of the methods was chosen "KLT" method in combination with "CRC" checksum for data contained in the data frame. There is an attempt to mathematically advanced analysis of possible transformations implemented in hardware data processing such as "FPGA". There were established the main objectives of a body of mathematical operations needed to transfer and implementation using "VHDL". Completion and a comparison of the effectiveness and ability to execute will follow.

ACKNOWLEDGMENT

This work was supported by the grant OPVK No CZ.1.07/2.2.00/28.0062, "Joint activities of BUT and VSB-TUO whereas creating the content of accredited technical courses in ICT" and by the SIX project No. CZ.1.05/2.1.00/03.0072, Centre of sensor information and communication systems.

REFERENCES

[1] "*NetCOPE Development Framework - INVEA-TECH*.". INVEA-TECH a.s., Accessed 2014-01-21. https://www.invea.com/en/products-and-services/fpga-development-kit/netcope

[2] *SIX research centre: Sensor, Information and Communication Systems*. Accessed 2014-01-22. http://www.six.feec.vutbr.cz/
[3] Shlens, Jonathon. Center for Neural Science, New York University. *A Tutorial on Principal Component Analysis*. New York City, 2009. Accessed 2014-01-21. http://www.snl.salk.edu/~shlens/pca.pdf
[4] Davlla, Carlos E. *Method and system for blind Karhunen-Loeve transform coding*. US. Application Data, US7103101 B1. http://www.google.com.br/patents/US7103101
[5] *MATLAB - The Language of Technical Computing*. Accessed 2014-01-27. http://www.mathworks.com/products/matlab/
[6] Ikenaga, Bruce. "*The Cayley-Hamilton Theorem*.". Accessed 2014-01-26. http://www.millersville.edu/~bikenaga/linear-algebra/cayley-hamilton/cayley-hamilton.html