EasyChair Preprint

№ 11566

Analysis of Pattern Error Correction in Cooperation with Cyclic Redundancy Check

Tomáš Pajda

December 19, 2023

Analysis of Pattern Error Correction in Cooperation with Cyclic Redundancy Check

Ing. Tomáš Pajda, SMIEEE
0000-0002-7053-7570
Institute of Electrical and
Electronics Engineers,
Information Theory Society,
Horné Naštice 117, Uhrovec,
Slovakia
Email:
ing.tomas.pajda@gmail.com

***Abstract*—This document describes the evolution and application of Pattern Error Correction in cooperation with Cyclic Redundancy Check principle, that is based on estimation of the possible error, that might happen during the transmission of the data, especially video stream transmission.**

***Index Terms*—FedCSIS, Pattern Error Correction, Cyclic Redundancy Check.**

I. Introduction

THIS document is an overview about the theory on which the Pattern Error Correction is being based. I have got an idea, that can help us in reconstruction of the data being transmitted through the network, just by using a Pattern Error Correction. What does it mean a Pattern Error Correction? A Pattern, in this case, is a codeword that is 1 byte long and its purpose is error estimation. It can be composed of zeroes and ones. If we consider, that maximally 4 errors in terms of bits happened, what is a half of the 1-byte set that the data were divided into, we should be able to get quite a high probability of getting uncorrupted data at the end of the transmission.

A. Description of The Idea

I thought like, what happens if we divide a group of bytes of the data received by a device into a 1-byte long segments and on each byte we apply an algorithm of Pattern Error Correction.

Firts of all, we need to compute CRC before sending the data through the network towards the receivers end device, where the CRC is being computed, too. If we try to correct the data by a Pattern, that is being set up combinatorically, we can try to compute the CRC at the receivers site again after adding this Pattern to a codeword and repeat until CRC is fine.

The Pattern is being set up in the following way. In case, where we consider, that only one error in terms of bits happened, we will use all combinations of one '1' digit and seven '0' digits throughout 1-byte long set of segments of received data. Let's demonstrate it in the following example of the Pattern:

1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1

In case of 2 errors, we can use Patterns like this:

1 1 0 0 0 0 0 0
1 0 1 0 0 0 0 0
.
.
.
0 0 0 0 0 0 1 1

In case of 3 errors, we will use Patterns from this set:

1 1 1 0 0 0 0 0
1 1 0 1 0 0 0 0
.
.
.
0 0 0 0 0 1 1 1,

and in case of four errors, we are able to use Patterns as:

1 1 1 1 0 0 0 0
1 1 1 0 1 0 0 0
.
.
1 0 1 0 1 1 0 0
.
0 0 0 0 1 1 1 1.

We assume, that a higher portion af all possible errors should be eliminated in this way.

II. Analysis of PEC with CRC

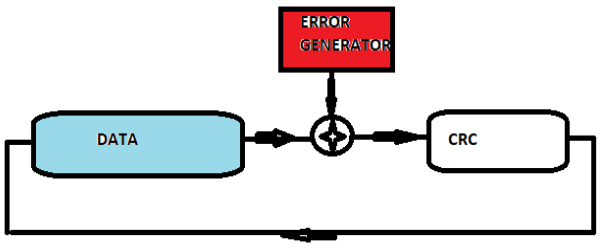
1.

For the purposes of analysis, we used a program created in C++, where we simmulated the whole process. This program contained a simmulation of CRC algorithm and a Pattern generator algorithm.

First of all, we converted hexadecimal data from a bitmap picture into binary. At the level of binary data we applied a random error from the generator without knowing what exactly the error vector was. Afterwards we wanted to know if we are able to discover what pattern to use in order to estimate the error and correct the image. In this case we know, that the error that happened will be estimated for sure, but it would be good to try it on real data transmission and especially on video stream. If we for instance compare this algorithm to Forward Error Correction codes, and for example Hamming Codes, where we are able to correct only up to t=(n-k)/2 errors, where if we have like 4-bits error code and 8-bits codeword, we would not be able to correct anyway. In this case, we also do not need the encoder in terms of FEC, we just need a CRC algorithm at the senders site and CRC algorithm at the receivers site.
The basics of the algorithm can be written as follows:

```
check(CRC);
if (CRC match)
   continue to following octet;
else
{
   while(CRC do not match)
   {
      optimise(DATA);
   }
}
```

During the proces of optimation the estimated errors from the error generator are being valuated towards the CRC match.
The advantage is, that if not more than 4 errors happen, we are always able to find the error pattern towards the CRC check to be ok.



Picture 1.

As we can see at the picture number 1, the whole proces is being cyclic with feedback.
In this case we do not expect, that more than 4 errors occure, but it would be very usefull to test it in real environment on lets say video streaming.

III. Implementation

The possible implementation could be made on Raspberry Pi, using linux distribution, where the whole algorithm should be applied. The main advantage is, that all datagrams on at the transport layer are using CRC, so we do not need an encoding algorithm at the beginning of the transmission. We just need a CRC and a Pattern Error Correction algorithm. So there is no need to change the architecture of the communication protocols at the senders site, just at the receivers site. In case, we do not find the correct pattern, we can apply the last found, because in this case it does not matter anymore.

IV. Possible Application

The possible application of this algoritgm is a Cable TV provider, where we can also test the efectivity in real environmnet. I geuss, it is easier to implement this kind of Pattern Error Correction than Forward Error Correction, because in Forward Error Correction case we need to adjust also the senders and receivers sites protocols sets.

V. Conclusion

I have gained this idea and I wanted to share my thoughts with academic world, because I think even if it is not an enhancement, it can be a new point of view on the topic of Backward Error Correction and the article can be helpful, even though it was not tested in the real environmnet. Hopefully, somebody can get inspired by the idea and it can be a topic for the further research in the field of communication and decoding techniques.

References

[1] P. Farkas, “Kódovanie a modulácie” , Slovak university of technology, 1993, pp. 1-261, ISBN 9788022705134.

[2] P. Návrat a kol., „Umelá inteligencia“, Slovak university of technology, 2015, ISBN 9788022743440.