Broadcast video streaming in Internet

Dmytro Shylovskyi

Student, Software Systems, Lviv Polytechnic National
University, UKRAINE, Lviv, S. Bandery street 12,
E-mail: dimas09@ukr.net

***Abstract* – the paper is dedicated to optimization quality of video streaming in network with low bandwidth which isn't insufficiently to broadcast video stream in real time. To improve method Forward Error Correction (FEC) through introducing varying redundancy ratio for each data element.**

Key words – streaming video, Forward Error Correction, FEC, lossy network, packets scheduling.

I. Introduction

Broadcast video streaming has some technical problems associated with unknown Internet's bandwidth and packet delivery time variation, so that may lead to a loss of data.

Error protection in broadcast video streaming could be reached on two levels: at the data link layer and the data decoding level. In the first case, error protection will be reached by changing a method of transfer or protocol of encoded video. In the second case, protection protection will be reached by changing encoding or decoding method of video and does not care about data transfer method.

Basic approaches at the data link layer are retransmission, forward error correction and diversification of channels. In approaches based on retransmission, receiver informs sender about which packets were delivered and which were not, and sender according to that messages resends packets which have been either damaged or lost. In approaches based on forward error correction, during broadcasting video the sender adds redundant data, which could be used to restore losses of data. For example, in the method of Reed-Solomon [1], during sending k packets sends n-k redundant packets and when no more than n-k any packets is lost (not necessarily extra packets) then all the k information packets could be always recovered. Diversification of channels in broadcasting video means sending data to multiple independent channels.

II. Video streaming optimization

The network which has non-guaranteed packets delivery, dynamically variable and limited bandwidth, limited size data packets was used in my research. Supposed using simplex communication, which is typical for broadcast streaming video on the Internet.

In paper [2] proposes solution for these problems. This solution uses the method of Reed-Solomon with constant redundancy coefficient for each level of the video. The aim of my research is development a new method for scheduling packets transfer that provides a high quality and aren't tied to a method of encoding video.

Forward error correction method[1] uses in this paper, based on redundant packets transferring, and the objective is approximated by objective of dynamic programming:

$$\sum_{i \in k} (D_i(\vec{x}) - \lambda x_i) \to \max \tag{1}$$

where x – integer vector of amount of packets for data elements transferring, D_i – expected quality improvement of all video data in the case of the i-th data element transferring, λ – multiplier changes dynamically depending on the network bandwidth and the virtual buffer size on the sender's side . The paper presents an interactive solution which consists of 3 steps in each iteration. In the first step new data elements will be added to schedule to increase the function value (1). If the first step doesn't get necessary result, then follows to the second step, where packets set searched by depth-first search with limited step in dependency tree of video data elements. The third steps will be added redundant packets for data element which had already been added to schedule.

Let's consider the first step of the iteration. At this step, the set A data elements is considered. And each element of set satisfies 2 conditions:

- packet have never been transferred;
- all packets, which depend on this packet, has transferred at least once.

The main idea of the first step is that data elements which have positive M_i Lagrangian for i-th data element are selected from the set A.

$$M_i(\vec{x}) = D_i(\vec{x}) - \lambda x_i \tag{2}$$

In this case, during transferring packets of that data elements, function (1) is increased and thats why solution improves. It is easy to show that the effective data element distortion which satisfies conditions the set A is

$$D_i(\vec{x}) = d_i p_{x_i}^{k_i} \prod_{\forall j : j \to i} p_{x_j}^{k_j} \tag{3}$$

where $p_{x_i}^{k_i}$ - the probability of i-th data element recovery, which consists of the basic k_i and redundant x_i-k_i packets. The probability is expressed by the formula

$$p_n^k = \sum_{i=k}^{n} C_n^k (1-\varepsilon)^i \varepsilon^{n-i} \tag{4}$$

Let

$$F_{x_i} = \frac{\sum_{l=k_i}^{x_i} C_l^{k_j} (1-\varepsilon)^l \varepsilon^{x_i - l}}{x_i} \tag{5}$$

If we substitute (4) in (3) and (3) in (3) then we will get maximizing condition of function (5) which is reached by maximizing x_i function (7). Only if this condition is positively. Let that x_i is as x_i', a corresponding value F is as F_i'. In this case, M function value equals to

$$M'_i(\vec{x}) = (d_i F'_i \prod_{\forall j : j \to i} p_{x_j}^{k_j} - \lambda) x_i' \tag{6}$$

x_i', M'_i value is calculated for each packet of the set A. We put a data elements with M'_i positive to subset A^* from the set A.

Let's try to improve the Lagrangian for the set A^*. Let's define Lagrangian increment for i-th data element as: subtraction the Lagrangian function from function M initial value. Only if i-th data element is transferred one time again.

$$\Delta M_i(x_i) = M(x_i + 1) - M(x_i) \tag{7}$$

For each packet of the set *A**, we consistently increase since x_i from x_i' to x_i'' which will satisfy two inequalities

$$\Delta M_i(x_i'' - 1) > 0, \ \Delta M_i(x_i'') \le 0 \qquad (8)$$

If the subset *A** is not empty, each *i* data element of the subset *A** will be consistently added to packets schedule x_i''. At this moment first step of iteration is finished.

If the subset *A** is empty, then the second step of iteration will be proceeded, otherwise the second step will be skipped and the third step will proceeded. Let's consider the second step of the iteration. At this step, we do the depth-first search with limited depth [3] of the set *B* data elements which have the following properties:

- all data elements of the set *B* are not yet added to packets schedule;
- data elements of the set *B* are numbered to condition that any data element doesn't depend on data element follows after;
- amount of data elements of the set *B* is not more than search depth *N*;
- there is assertion for any data element of the set *B*: "All data elements which depends on data element of the set *B*, belong to *B*, or have already been added to packets schedule".

The set B satisfies inequality

$$\sum_{i \in B} D'_i > 0 \qquad (9)$$

The search stops when the first set that satisfies these conditions was found. In this case, each *i* data element from the set *B* we series add to packets schedule x_i', and we guess that the second step of iteration is completed. Note that in a depth-first search depth depend on computing power, so the search gives a more accurate result on more powerful computers.

When non-empty set of packets will be added to the schedule on the first or the second step, then we proceed to the third step. On the 3rd step, we consider the set *C* of data elements, each one satisfies the following conditions:

- data element packets have been added to the schedule;
- packets have been added to the schedule at least for one of data elements , current element depends on it.

The main idea of the third step is successive calculating of Lagrangian increment for each data element of the set *C* and packets amount x_i, x_i+1, etc., until packets amount satisfying the conditions (9) will be found. If this amount x_i'' more than x_i, then redundant x_i''-x_i packets will be added to the schedule for *i*-th data element. Lagrangian increment for the *i*-th data element will calculated by the formula

$$\Delta D_i(\vec{x}) = \left(\frac{p^{k_i}_{x_i+1}}{p^{k_i}_{x_i}} - 1\right)\left(D_i^s + \sum_{\forall j : i \to q} D_j^s\right) - \lambda \qquad (10)$$

Conclusion

In the paper suggested new improving of forward error correction method, which allows to implement FEC effectively in broadcast video streaming over lossy data networks. This method is not tied to a method of encoding video with variety redundancy coefficient one level packet and has enough computing speed to run in real time on modern computers.

References

[1] D. Mandelbaum. An adaptive-feedback coding scheme using incremental redundancy // IEEE Transactions on Information Theory - Piscataway, N.J., USA: IEEE, May 1974. – Volume 20 – № 3 – P. 388 – 397.

[2] W. Tan and A. Zakhor. Video multicast using layered FEC and scalable compression // IEEE Trans. Circuits Syst. Video Technol. , Mar. 2001 - vol. 11, no. 3, P. 373–386.

[3] S. Skiena. The Algorithm Design Manual // Springer; 2nd ed. 2008 edition. – 2010. – P. 169 -184.