Traffic Optimized FEC Control Algorithm for Multimedia Streaming Applications.

Alexander Magzumov, Wonkap Jang
Digital Media R&D Center
Samsung Electronics
E-mail : wonkapjang@samsung.com

Abstract

Packet losses in the Internet can dramatically degrade quality of multimedia streams. Forward Error Correction (FEC) is one of the best methods that can protect data from packet erasures by means of sending additional redundant information. Proposed control algorithm provides the possibility of receiving real-time multimedia streams of given quality with minimal traffic overhead. The traffic optimization is reached by adjusting packet size as well as block code parameters. Calculations and simulation results show that for non-bursty network conditions traffic optimization can lead to more than 50% bandwidth reduction.

I. Introduction

With recent break-through in network technologies, audio and video streaming tools have become extremely popular. A great deal of real-time streaming applications uses UDP as a base transport protocol because of its high throughput. In contrast with TCP, UDP is not reliable. Unfortunately, not all networks are error free. The Internet suffers from packet losses. There are also numerous types of lossy local networks and lines, such as wireless networks and point-to-point modem connections.

Several error-correction schemes were introduced in order to cope with the negative impact of network errors on final quality of multimedia stream. There are basically two well-known techniques - Automatic Repeat Request (ARQ) and Forward Error Correction (FEC). Combination of the above methods into hybrid scheme is also possible. The implementation of ARQ scheme is simple but has two drawbacks:

- in case of large multicast group ARQ significantly increases network traffic
- it requires large time delay (several round trip times), which may not be acceptable.

Thus, for applications that require little delay, such as Internet conferencing software, FEC method is an obvious choice. In utilizing such method, the behavior of the Internet, including the loss patterns, has to be understood correctly, and the FEC parameters have to be adjusted according to the particular situation.

FEC technique implies sending redundant data along with stream of useful data. There are numerous variations of FEC schemes. Some FEC schemes use data repetition and interleaving [5], others involve checksums based on bitwise "exclusive or" operations [6], the others are known as block – code schemes [4]. Schemes based on checksum calculation or data repetition are easy to implement and don't require much computational power. On the other hand, the block-code based schemes are the most robust and give the best level of data protection. An advance in development of high-speed CPU's allows the use of block-code based schemes for packet recovery.

The main principle of block-based FEC can be formulated as follows: for a group of k packets which carry useful data sender application creates l error correction packets and sends them to a receiver along with the original data stream. If total amount of received packets is more or equal to k, all groups of data packets can be restored.

In this paper FEC control algorithm is introduced in order to deliver multimedia data of predefined quality with minimum FEC packet overhead via lossy channel. This goal can be achieved by adjusting FEC parameters such as k and l values, along with data packet size, according to the loss pattern measurements. Several types of loss patterns were studied. The conditions under which the algorithm leads to significant bandwidth reduction were found. The control scheme was tested with Real Time Protocol [7] (RTP) audio / video streaming application for local network with loss simulation.

II. Loss model for the Internet

The main cause of IP-packet errors in the Internet is packet drops on routers due to congestion, while bit errors are very seldom and their fraction is negligible [1]. According to numerous observations the Internet often exhibits bursty behavior. It means that a probability of IP-packet loss depends on whether the previous packet was lost or not. The most popular and simplest model of packet loss is 2-state Gilbert model. According to this model a line can be found in two states: *good* and *bad* (Figure 1).

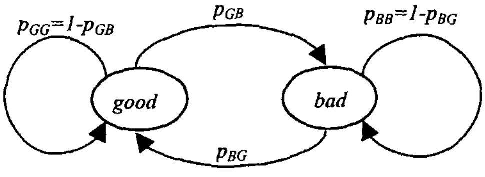

Figure 1: 2–state Gilbert loss model.

The model can be described in terms of 2 independent values: probability of transition from good state to bad one - p_{GB} and probability of transition from bad state to good one - p_{BG}. More obvious way to represent Gilbert model is to introduce total packet loss fraction p_0 – probability to find system in *bad* state, and average burst length L_b, which gives the average amount of consequent packet losses. The first two values can be found as:

$$p_{BG} = \frac{1}{L_b} \tag{2.1}$$

$$p_{GB} = \frac{p_0}{L_b(1-p_0)} \tag{2.2}$$

Numerous loss measurements show that packet losses in the Internet strongly depend on particular route [1], [2]. There is no certain dependency between packet loss fraction p_0, stream's bit rate W and IP-packet size sz. According to [1], packet loss rate seems independent of packet size and stream bandwidth. Experimental data measured in [2] show that for some routes packet loss rate is independent on size, but depends on bandwidth W, while for other routes there is a strong correlation between loss fraction and packet sending rate $f=W/sz$.

According to data represented in [2] and our own observations, average burst length depends on packet sending rate. We found that Internet connection can exhibit either Bernoulli loss process behaviour with average burst length is equal to $1/p_{BG}$ or bursty Gilbert loss process behavior with burst length $L_b>2$ and strong dependency on packet rate.

Although in this paper we concentrate our attention on the Internet it is worth to mention that our FEC control scheme can be applied to any network that exhibits Gilbert loss process behavior.

III. FEC Control Algorithm.

Let $P(m,n)$ stands for probability of loosing more than m packets from n packets . For given FEC parameters k and l the $P(l,k+l)$ gives probability of final FEC block loss:

$$P(m,n) = \sum_{z=m+1}^{n} R(z,n) \tag{3.1}$$

Where $R(z,n)$ denotes probability to loose exactly z packets from n ones. $R(z,n)$ can be calculated using solution obtained in [3].

$$\begin{aligned} R(z,n) = {} & \\ & p_G\gamma^{n-2z+1}(1-\beta)(1-\gamma)\sum_{i=0}^{z-1}\tbinom{z-1}{i}\tbinom{n-z}{i+1}(\beta\gamma)^{z-1-i}\left[(1-\beta)(1-\gamma)\right]^i + \\ & p_B\gamma^{n-2z}(1-\beta)\sum_{i=0}^{z}\tbinom{z}{i}\tbinom{n-z-1}{i}(\beta\gamma)^{z-i}\left[(1-\beta)(1-\gamma)\right]^i + \\ & p_G\gamma^{n-2z+1}(1-\gamma)\sum_{i=0}^{z-1}\tbinom{z-1}{i}\tbinom{n-z}{i}(\beta\gamma)^{z-1-i}\left[(1-\beta)(1-\gamma)\right]^i + \\ & p_B\gamma^{n-2z}(1-\beta)(1-\gamma)\sum_{i=0}^{z-1}\tbinom{z}{i+1}\tbinom{n-z-1}{i}(\beta\gamma)^{z-1-i}\left[(1-\beta)(1-\gamma)\right]^i \end{aligned} \tag{3.2}$$

Where p_G=1 p_0 , p_B=p_0 , γ =p_{GG} = 1 $L_b^{\ 1}$ p_0 (1 p_0)^{1} ,and β=p_{BB}=1 $L_b^{\ 1}$ represent the probability of finding channel in good state, probabilities to find channel in bad state, *good* to *good* transition probability, and *bad* to *bad* transition probability, respectively. The advantage of this solution is that it provides a fast algorithm with $O(n)$ time complexity for finding $R(z,n)$, which makes it suitable for real time calculations[3].

Let us now formulate optimization task for FEC control algorithm. There are several conditions to be satisfied. The main requirement is to keep the quality of multimedia stream. It means that block failure probability should be small enough:

$$P(l, k+l) \leq \varepsilon \tag{3.3}$$

Here, ε is a maximum probability of block recovery failure.

Let us consider the relative bandwidth overhead, which gives the ratio of current bandwidth to the "net" multimedia stream bandwidth:

$$f_{ovh} = \frac{1+\frac{l}{k}\left(1+\frac{sh_{FEC}}{sz}\right)}{1-\frac{sh_{RTP}}{sz}} \tag{3.4}$$

Here, sh_{FEC} is FEC header size inside RTP packet, sh_{RTP} is RTP over UDP packet header size.

We choose function (3.4) as optimization criterion for FEC control algorithm.

There are also other conditions that have to be satisfied. In the case of audio or video conferencing there is very important *real time constraint*, which can be formulated as follows:

$$k <= k_{\max} = \frac{\tau W_0}{sz - sh_{RTP}} \tag{3.5}$$

Here, W_0 is "net" bandwidth of multimedia stream that is bandwidth without packet headers overhead, τ is maximum allowed delay between arrival of the first and the last packets in FEC block. It can be estimated as follows: $\tau \sim t_D$ t_{TT} t_{jitter}, here t_D stands for desired multimedia stream delay time, t_{TT} is "one way trip" time and t_{jitter} is network jitter. This real time constraint should be also taken into account when considering endpoint device with limited memory capacity. In that case, τ will be proportional to the maximum allowed buffering queue size S_{buf} for multimedia packets: $\tau \sim S_{buf}/W_0$.

There also exists performance-related limitation that should be imposed when endpoint device has inadequate processing power. Our FEC algorithm uses block codes described in [4]. For this algorithm decoding and encoding speed are proportional to l- parameter, that's why:

$$l \le l_{max} \quad (3.6)$$

For this FEC scheme the (k,l) parameters must satisfy the following requirement:

$$k + l \le 255 \quad (3.7)$$

The RTP packet size must be in the following range:

$$sh_{RTP} + sh_{FEC} < sz \le MTU, \quad (3.8)$$

Where *MTU* stands for Maximum Transfer Unit i.e. the maximum size of packet, which can be transferred without fragmentation.

Now the traffic optimization problem for FEC-protected stream can be formulated as follows: for given multimedia stream with "net" bandwidth W_0, and for given final FEC block recovery failure fraction ε, find such a FEC parameters (k,l) and such an IP packet size sz, that conditions (3.3), (3.5)-(3.8) are satisfied and relative bandwidth function f_{ovh} reaches its minimum .

To solve this problem we also assume, that packet loss probability and burst length are functions of packet size, stream bandwidth and time:

$$p_0=\pi(sz, W, time) \quad (3.9)$$
$$L_b=\lambda(sz, W, time)$$

Here, $W=W_0(1+sh_{RTP})(1+sh_{FEC}*l/k)$ represents the channel bandwidth.

The goal of FEC control algorithm is to solve aforementioned optimization task using feedback information about functions (3.9).

As a measure of traffic optimization the following gain function can be taken:

$$G=\frac{f_{OVH}\big|_{MTU}}{f_{OVH}\big|_{MIN}}-1 \quad (3.10)$$

Where numerator designates overhead for *MTU* packet size and denominator designates the minimum overhead.

IV. Results for different loss patterns.

As mentioned above, the main cause of packet loss in the Internet is congestion of routers. Let us consider the most frequent situation when packet drops are due to one *bottleneck* router and analyze different types of possible loss patterns (3.9) generated on such a router. All calculation results presented here were obtained for quality parameter $\varepsilon=10^{-4}$, which gives very good final stream quality: for example if video stream has 30 fps, only 1 frame per 5 minutes is lost. Calculations for other values of ε in range $[10^{-6},10^{-2}]$ give qualitatively same results, while absolute values are different.

We will begin analysis with the simplest case of Bernoulli loss process for which $p_{GB}=p_{BB}$ and therefore $L_b=(1\ p_0)^{1}$. Let us also start with constant value of loss fraction $p_0=const$. It is easy to show that such loss pattern can be found if the bottleneck router has FIFO [8] queuing policy or Waited Fair Queuing [8] (WFQ) policy with queue, which size is set in *bytes* rather than in *packets*. Figure 2 represents dependency of relative overhead function (3.4) from packet size for different values of τW_0. The loss probability p_0 is equal to 0.05. We can see strong dependency of traffic overhead function on τW_0 parameter. For small values of this parameter the gain (3.10) reaches almost 0.4, while for big values it is moderate - about 0.05.

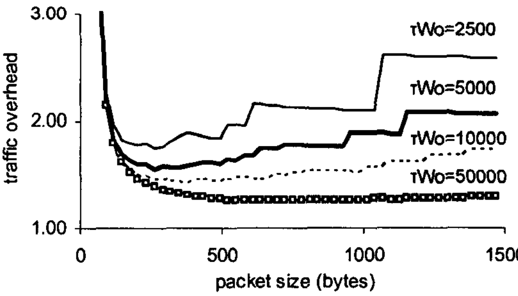

Figure 2: Relative overhead versus packet size for Bernoulli loss process with $p_0=0.05$ and $\varepsilon=10^{-4}$ for different τW_0 values.

Figure 3 shows overhead function versus packet size for different packet loss rates for $\tau W_0=5000$. We can see that gain parameter G is growing with increase of p_0 from 0.3 at $p_0=0.01$ to 0.6 at $p_0=0.2$.

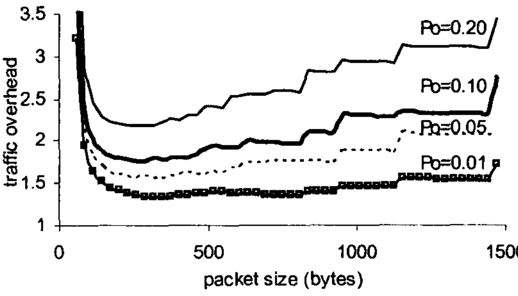

Figure 3: Relative overhead versus packet size for Bernoulli loss process with $\varepsilon=10^{-4}$, $\tau W_0=5000$ and different loss fractions p_0.

According to our observations of real Internet loss process the is also "anti-burst" situation for which $L_b \quad 1 \le (1 \quad p_0)^{1}$. It can be found very often for routers with Random Error Detection[8] (RED) drop policy. Our calculations show that qualitative behavior of FEC control algorithm is similar to the case of pure Bernoulli process described above.

Let us now consider bursty network conditions. Let p_0 will be still constant function while $L_b = L_{MTU} \, W_{sz} \, MTU \, / \, (W_{MTU} \, sz)$, where W_{sz} and W_{MTU} are equal to bandwidth for packet size sz and MTU correspondingly. It is easy to show that this situation can be found for FIFO routers with *tail drop* or *head drop* policies. Indeed, burst means that router is closed for some time Δt. The amount of consequent packet drops will be equal to Δt multiplied by beaconing frequency. At the same time loss fraction is still independent on sending rate. Figure 4 represents results for such a situation. We can see that for bursty network condition control algorithm has trivial solution at $sz=MTU$. This result can be explained by strong dependency of block loss probability function (3.1) from burst length.

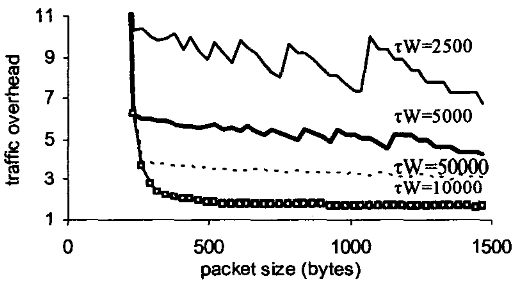

Figure 4: Relative overhead versus packet size for Gilbert loss process with p_0=0.05 and $\varepsilon=10^{-4}$ and L_b=2*MTU/sz for different τW_0 values.

The last frequent loss pattern occurs in situation when WFQ router has queue length in *packets* rather then in *bytes*. In this case, it is easy to show that $p_0 = 1 \quad (1 \quad p_{MTU}) W_{MTU} sz \, / (W_{sz} \, MTU)$. For such loss pattern, the control algorithm also gives trivial solution, but overhead curve goes down much more steeply than in the previous case.

The FEC control algorithm presented here has been tested in local network with packet loss simulations. Our application sends and receives MPEG4 stream via RTP. FEC packets are sent using additional RTP connections as described in [6]. Feedback data is delivered by means of RTCP receiver reports. We found that the calculation results are in good agreement with the measurements made in these experiments.

V. Conclusions and future work.

In this paper the FEC traffic optimization task for multimedia streams with real-time constraints was formulated. The calculations and simulation results show the ability of FEC control algorithm to save significant bandwidth by means of finding the optimum packet size. The best results obtained for non-bursty network conditions, when proposed algorithm can dramatically decrease required bandwidth. It is worth mentioning that modern routers with RED drop policy exhibit "anti-bursty" behavior when burst length parameter is very close to 1. For such routers our optimization scheme gives very good results. However for bursty networks there is no gain in adjusting packet size.

In the future, we are planning to apply our scheme to the real internet conditions. The control algorithm in this case has to constantly gather connection statistics for several packet sizes to be able to identify queuing policy on bottleneck router and correctly predict the dependency of loss parameters on bandwidth and packet size.

References

[1] Stephan Wegner, "Proposed error patterns for Internet experiments", ITU-T Q.15/16 Video Coding Experts Group, Document Q15-I-16, Oct 19-22 1999.

[2] B.Girod, K.Shtuhlmuller, M.Link and U.Horn., "Packet Loss Resilient Internet Video Streaming", SPIE Visual Communications and Image Processing 99, San Jose CA.

[3] Homayoun Yousefi'zadeh, Hamid Jafarkhani, "Analytical Modeling of Burst Loss: A Study of the Gilbert Model".http://www.ece.uci.edu/~hyousefi/pub.html. Will be published.

[4] L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols", ACM Computer Communication Review, Vol.27, n.2, Apr.97, pp.24-36.

[5] J-C.Bolot, D. Towsely. "Adaptive FEC-Based Error Control for Internet Telephony", Proc. IEEE Infocom'99, NY, March 1999.

[6] J. Rosenberg, H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733.

[7] H. Schulzrinne at al."RTP: A Transport Protocol for Real-Time Applications", RFC 1889.

[8] Metz C., "IP QOS: traveling in first class on the Internet". Internet Computing, IEEE,Vol. 3, Issue: 2 ,1999 , pp. 84 -88.