Robust Video Calls for Emergency Services over IP Based Networks

Berend W.M. (Martijn) Kuipers, Ricardo N. Vaz, and Mario S. Nunes

INESC-ID*, R. Alves Redol, 9, 1000 Lisboa, Portugal
{martijn.kuipers,ricardo.vaz,mario.nunes}@inesc-id.pt

Abstract. Network links in emergency or rescue scenarios often operate under difficult circumstances, which makes live video feedback almost unusable. Received video quality is dependent on the available link rate and the packet loss ratio, which are inter-related in a congested network link. Even low packet loss ratios (PLRs) can significantly reduce the video quality. In this paper, a packet level parity Forward Error Correction (FEC) is applied to the video stream in order to reduce the video PLR. A constant gross data rate is assumed, such that adding a FEC leads to a decrease in effective video data rate. The FEC block is truncated at the end of each video frame, such that there are no inter-frame dependencies for FEC correction. An algorithm is proposed to optimize the FEC length, based on the Quality of Experience as modelled by the ITU-T R G.1070 standard. It is shown that the optimization algorithm can significantly increase the video quality, without increasing the gross data rate. The algorithm has been evaluated through simulations, which confirm the very significant increases in subjective video quality.

Keywords: video, FEC, G.1070, delay, packet networks.

1 Introduction

Past experience has shown that in emergency scenarios, such as national disasters, e.g., earthquake, terrorist attack, or other scenarios, the current telecommunications networks do not meet the needs of emergency personal. Under these circumstances network links become very unreliable, resulting in a higher packet loss ratio than usual. Search and rescue operations would benefit from a live video feedback from a surveillance robot or helmet-camera, but video transmissions have strong bandwidth, delay and loss requirements.

Video transmissions are highly affected by frame losses. Typical video encoding is based on not only compressing each frame independently, but also uses the correlation between consecutive frames in order to reach a higher compression. However, in a lossy medium some errors could slip in and if they cannot be corrected, the error will propagate throughout the stream, accumulating up to a point where the image is no longer perceptible. In order to guard against the

* This work was partially supported by FCT (INESC-ID multi-annual funding) through the PIDDAC Program funds.

J. Rodriguez, R. Tafazolli, C. Verikoukis (Eds.): MOBIMEDIA 2010, LNICST 77, pp. 383–396, 2012.

propagation error, the MPEG-4 encoder uses anchor-frames (I-frames), which are not dependent on previous frames, such that any error will not propagate beyond the reception of such a frame. Intermediate frames in the MPEG-4 encoder are P-frames and B-frames, which rely on previous and future frames, respectively. In the case of live (or real-time) video encoding, the encoder cannot afford to wait for future frames, hence the streams only consist of I-frames and P-frames. In MPEG-4 terms, a Video Object Plane (VOP) is another nomenclature used to indicate a single frame, whether that is an I, P or B-frame.

A typical Group-of-VOPs (GOV) can be composed of I and P frames, as shown in [1], where it can be seen that a P-frame is dependent on previous frames. I-frames are normally larger (in terms of storage) than P or even B frames. Adding a Forward Error Correction (FEC) scheme makes it possible to recover certain errors, which results in a decrease in packet loss ratio at the cost of a delay and transmission overhead.

More important than optimizing for QoS in terms of packet loss and bandwidth is to optimize for the perceived end-user video quality, so called Quality of Experience (QoE). An interesting approach on video quality subject is presented in [2] where the QoS objective metrics are mapped to the QoE subjective indicators. Thus, QoS thresholds can be defined, which allow to estimate the required QoE. The measurement of the quality indicators is a complex procedure and is hard to be performed in a practical manner, since it involves events related to several layers. Also, the impact of these quality indicators on QoE is not easily predicted.

Several techniques or algorithm types have been proposed to mitigate the problems in delivering video over the internet [3]. For instance in [4], an algorithm is proposed to support video transmission, where the sending video bit rate and the number of FEC packets are automatically modified through self-adapted feedback. However, it does not use information of the received video quality to adjust the video sending rate, but makes use of the amount of ACK packets to perform the mentioned adjustments only.

An adaptive channel error FEC algorithm to balance the trade-off between the QoS of video transmission and the bandwidth utilization ratio in wireless IP networks is proposed in [5]. However, it is often not possible to guarantee the QoS in the network, so that the impact on the perceived video quality is not known.

An effective feedback-free loss recovery scheme for layered video was proposed in [6], which combined FEC and a flow replication technology. However, this approach was introduced for playback streams, which can tolerate higher delays by means of buffering. Interactive real-time video services, such as video telephony, do not allow large delays, so these schemes are not adequate.

In this work we propose a dynamic length parity FEC, protecting only against a single packet loss in a block. Shorter blocks lead to higher protection, but more overhead, whereas longer block lead to less protection, but also less overhead. FEC schemes based on higher-order finite fields, such as Reed-Solomon (RS) codes, are computationally more complex [7]. Although they offer better burst-protection than the parity FEC, this typically comes at the cost of increased latency, which makes them unsuitable for real-time interactive streams with a strong delay requirement. Interleaving FECs are suitable when the unit size is smaller than the packet size or when the end-to-end delay is unimportant [8]. However, in this case each unit (video-frame) requires multiple packets for transmission, ruling interleaving FECs unsuitable.

This work proposes an algorithm that optimizes the protection/overhead trade-off based on the perceived video quality, obtained from the ITU-T R G.1070 [9]. Life video feedbacks from rescue or investigative missions are likely to use multicast to transmit to various control centres in the field and/or the back-office, which rules out the use of TCP due to its need of retransmissions of certain frames to specific receivers only. Another problem of TCP is that it does not perform well in interactive applications, with strict delay bounds [7]. The solution proposed in this work is to transmit the stream using UDP, allowing the stream to be multicasted and the ability to auto-correct some frame losses without the need for retransmissions. Further bandwidth control could be required and probably protocols as DCCP can be utilized, however this is out of scope for this work, which focusses on the video quality of the stream. Note that the use of a packet-level FEC using a simple marker in the packet header for the parity packets, as proposed in [10], allows the stream to be handled by receivers which do not support the error correcting mechanism.

This paper is organized as follows. In Section 2, the proposed algorithm is described. The algorithm is evaluated through simulation in Section 3. Finally, conclusions are drawn in Section 4.

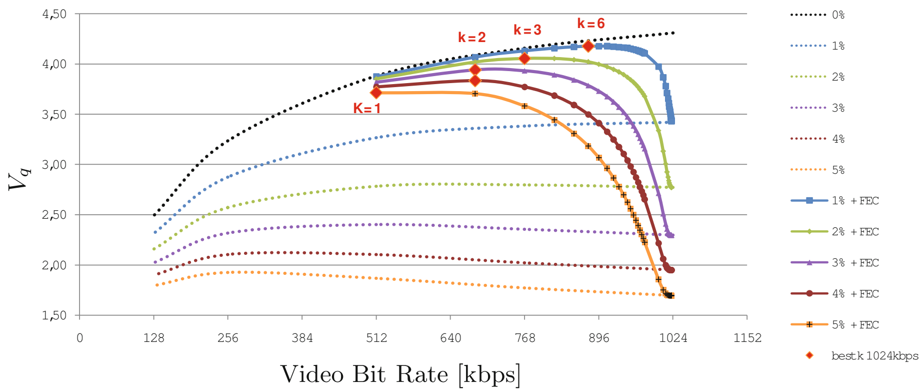
2 FEC Block Length for Best QoE

As aforementioned, minor losses can severely reduce video quality. Therefore, FEC schemes are used to reduce losses and increase received quality. Note that losses in the network occur on a per packet basis, where single packet losses are most frequent [11, 12, 13]. Therefore, this paper uses a simple packet-level parity FEC [14]. As a first step, the optimal value for the FEC block length, k, is calculated with respect to the video quality, V_q, metric from the G.1070 standard. Analysing this standard shows that one can obtain a higher V_q when using a lower data rate without losses, than with a higher data rate with losses, as packet losses significantly reduce the perceived V_q.

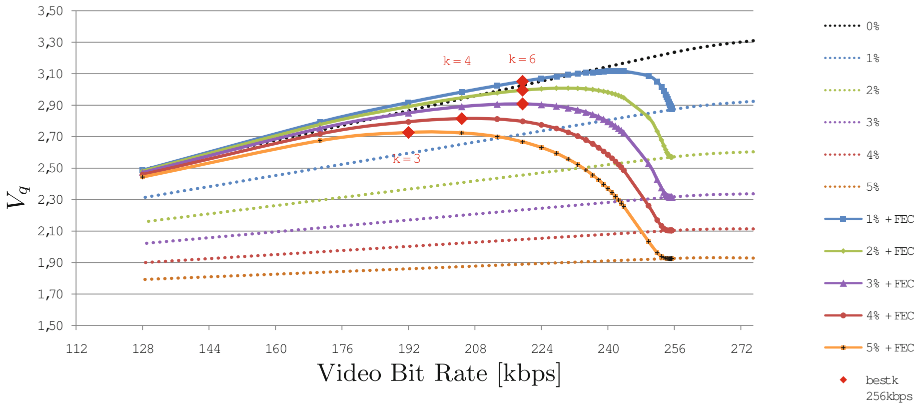
In this work a constant gross data rate is assumed, such that introducing FEC comes at the cost of lowering the effective video data rate. The relation between the gross and the video data rate is given by:

$$Br_v = \frac{B_{gross} \cdot k}{k+1} \tag{1}$$

where B_{gross} is the gross data rate. Larger values for k give less protection, but does not lower the actual video data rate much as is the case for low values of k. In the case of $k = 1$, the video data rate is halved. In the case of not applying FEC, $k \to \infty$, the gross data rate is equal to the video data rate. The



(a) 1024 kbps



(b) 256 kbps

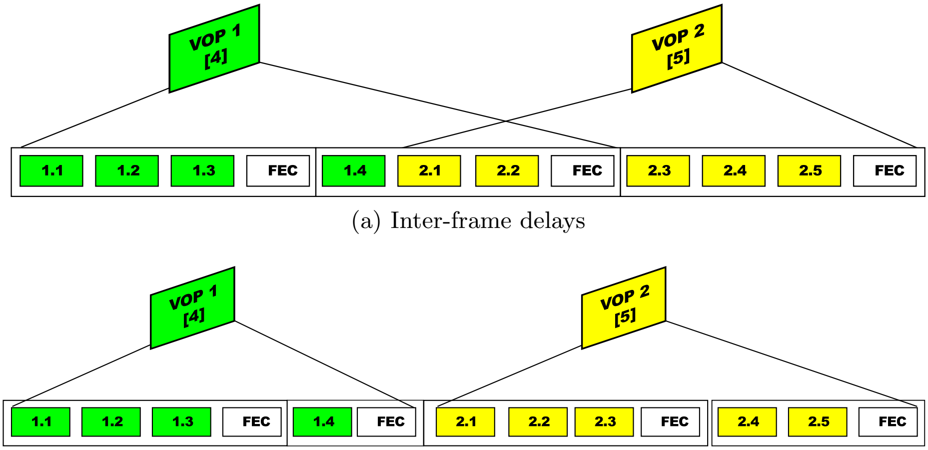
Fig. 1. V_q for the actual video rate, with and without FEC protection for 2 different gross-data rates, for various packet loss ratios and for different values for k

video quality is a trade-off between PLR and effective video data rate, where an algorithm to optimize the video quality is introduced. Since the gross data rate is unaltered, the PLR (without FEC) can be assumed to be unaltered as well.

The video quality, V_q, for the actual video rate, with and without FEC protection for various packet loss ratios, for different values for k and for gross-data rates of 1024 and 256 kbps is shown in Fig. 1. The lines without markers in fig. 1 show the V_q for a video rate without FEC protection and are the same for both figures. The lines with markers show the V_q for a video rate with FEC protection. The latter is different in each figure, as the gross-data rate is different. The lowest video rate with FEC protection is achieved when $k = 1$, corresponding to the repetition of each packet. From these figures it can be seen that with relatively low values for k, one can obtain a much higher V_q, than without using FEC using higher data rates.

In real-time traffic, such as video telephony, there are two ways that FEC introduces delay. The obvious cause of delay comes from the fact that extra packets, the FEC parity packets, are to be transmitted. Since in this work, the gross data rate is retained, this does not introduce any extra delay. The second form of delay requires a more in-depth knowledge of the FEC method applied. In this work, a simple parity FEC is applied, where k packets are protected by a single parity packet. This allows for a single lost packet to be replaced in k packets. If k increases, the overhead decreases and vice versa. Video streams encode video frames and send them over the network. This is a bursty process, as in real-time streams the period between video frame encoding is the inverse of the video frame rate. Since the payload of IP packets is limited, here it is limited to 1024 bytes, a single video frame is sent as a number of IP packets, creating the bursty IP packet stream. In order for the parity scheme to correct a single packet in k data packets, it needs first to receive $k+1$ packets. However, if k is larger than the number of IP packets for a single video frame, then the decoder will have to wait for the next burst of IP packets, originating from the following video frame. This means that an additional delay is introduced, equal to the video frame period. Smooth video is commonly encoded at a frame rate of 25 frames per second (fps), which corresponds to a 40 ms period, whereas for a frame rate of 5 fps it increases to 200 ms, which is close to a common acceptable limit used for real-time video streams.

For example, in Fig.2 , a sequence of 2 VOPs are shown. The VOPs are transmitted in 4 and 5 IP packets respectively for VOP1 and VOP2. For illustration, in Fig. 2a, a FEC-block of 3 packets is assumed, where a FEC packet is introduced after 3 IP packets. As VOP1 comprises 4 packets, two FEC blocks (data and parity packets) are need to transmit. After the reception of the 8 packets, the VOP can be decoded, which is only created O_{fr}^{-1} seconds later, causing an additional delay. This same delay is also introduced for very low PLRs, ($P_{plv} \ll 1\%$).



(a) Inter-frame delays

(b) No Inter-frame delays

Fig. 2. Example of dynamic FEC length eliminating inter-frame delays

In the previous section, a fixed length FEC is applied to a number of IP packets, which can cause inter-frame jitter as explained above. In order to not have any inter-frame jitter, the FEC parity packet is always inserted after the end of each video frame. Truncating the FEC at the end of a VOP gives a slightly better FEC protection, such that the minimum required protection given by the optimal value for k can be guaranteed. The second FEC block of VOP1 is not filled entirely by VOP1 packets, so 2 IP packets from VOP2 are added. The remainder of VOP2 is transmitted in a third FEC block. In Fig. 2b, each VOP is terminated by a FEC block immediatelly. The second block of the first VOP is now followed by a FEC. VOP2 starts filling a new FEC block. As can be seen by comparing the two approaches, one more IP packet is needed for the second approach, which also has one FEC block more. However, VOP1 can immediatelly be repaired after the reception of the second FEC block, which is after 6 IP packets, in contrast with the approach in Fig. 2a, where it is only after 8 packets. The latter approach does not introduce Inter-frame jitter since, no FEC blocks contain packets from different VOPs.

However, as stated previously, the introduction of FEC packets is performed in the proposed algorithm at the cost of reducing the video bit rate because the gross bit rate is kept constant. Truncating the FEC before the complete block of IP packets has been sent introduced a slight overhead, as the FEC overhead is increased. However, this overhead is assumed to be acceptable as it completely mitigates the inter-frame jitter problem.

3 Evaluation

For evaluation of the proposed video quality protection algorithm, it has been applied to the Akiyo stream. The content of this stream is a news reader, which has a length of only 10s. Therefore, it has been repeated 24 times so a sufficiently sample base is obtained. The original Akiyo stream is YUV422 based with a frame rate of 29.7 fps and a CIF framesize. The MPEG4-2 streams analysed for this work were created with FFmpeg [15], where the original frame rate was transcoded to the optimal frame rate, O_{Fr}, which can be obtained from ITU-T R G.1070 [9] and is given by

$$O_{fr} = v_1 + v_2 \cdot Br_v, \qquad 1 <= O_{fr} \leq 30, v_1 \text{ and } v_2: \text{ const.} \tag{2}$$

The optimal framerate values for Akiyo test sequence at the various target datarates is given in Table 1.

Knowing the test sequence a-priori allows to analyse the distribution of the FEC block length and its overhead due to truncating, but it should be stressed that it is not needed to know the stream contents beforehand. The optimal block length can be derived using the approach outlined in Section 2. The test sequence is also simulated with NS2 for 2 different error models.

Table 1. Optimal framerate values for the Akiyo test sequence

bit rate [kbps]	O_{fr} [fps]
128	4.3
256	7.1
512	12.8
768	18.5
1024	24.2

3.1 FEC Block Length Distribution for the Akiyo Sequence

The number of packets per video frame is dependent on the used video encoder, the video encoder settings such as frame rate, data rate, and when using an encoder such as MPEG4, and on the actual video content. Since this paper focuses on video-telephony, where there is usually large redundancy between consecutive frames, the number of packets per frame is normally low compared to a very dynamic content. Table 2 shows the histogram for the number of IP packets per video frame for the Akiyo sequence, which can be seen as video-telephony content, encoded using MPEG4 at various target data rates, with the optimal video frame rate. The table shows that there are no video frames that require more than 8 IP packets for an encoding rate of 1024 kbps and using 1024 bytes UDP payload size. It also shows that at the highest target data rate (1024 kbps), approximately 10% of the all packets have less than 6 packets per frame. For lower data rates this number decreases and for a target data rate of 128 kbps around 15% of all packets have less than 4 packets per video frame. The limit for k is given by assuring that all video frames can be protected by a single parity packet, allowing the FEC block to be truncated in practice.

With the distribution of the number of IP packets per VOP, the distribution of FEC blocks can now be obtained. The upper triangular matrix in Table 3 shows the FEC block distribution, where on the diagonal are the number of FEC blocks with the optimal value for k. The truncated FEC blocks are accounted for in the rows above the diagonal. As an example assume an optimal value of $k = 3$ and a VOP consisting of 8 packets. This VOP is transmitted with 2 FEC blocks with the optimal length ($k = 3$) and one block with length 2. Since the truncated part is always transmitted as a single block, the sum of all the rows above the diagonal accounts for all VOPs that needed a truncated FEC. This also means, that if a FEC of fixed length had been used, the sum of all rows above the diagonal is the number of times an inter-frame delay would have occurred, which is shown in relative values in the table. Another important value that can be obtained from the upper triangular matrix is the average value of k, which is an indication of the average video quality of the stream. Other streams than Akiyo, will follow a similar trend, but with a different distribution of the number of IP packets per frame, and thus also a different distribution for the FEC block length of Tables 2 and 3, respectively.

Table 2. Distribution of the number of IP packets per Video Frame for the Akiyo sequence at various data rates in absolute and relative values

Packets per video frame	Distribution [kbps]									
	1024		768		512		256		128	
	N_f	[%]	N_f	[%]	N_f	[%]	N_f	[%]	N_f	[%]
1	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
2	0	0.0	0	0.0	0	0.0	0	0.0	1	2.3
3	2	0.8	1	0.5	3	2.3	8	11.0	6	13.6
4	12	4.9	10	5.4	14	10.8	22	30.1	19	43.2
5	10	4.1	17	9.1	92	70.8	16	21.9	17	38.6
6	206	84.8	146	78.5	7	5.4	18	24.7	1	2.3
7	7	2.9	6	3.2	9	6.9	9	12.3	0	0.0
8	6	2.5	6	3.2	5	3.8	0	0.0	0	0.0
$\geq$9	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
Total	243	100.0	186	100.0	130	100.0	73	100.0	44	100.0

Table 3. Distribution of FEC block length for the Akiyo sequence at 1024 kbps

Packets per video frame	Optimal k								
	1	2	3	4	5	6	7	8	9
1	1437	19	19	10	206	7	6	0	0
2		709	16	206	7	6	0	0	0
3			462	9	8	2	2	2	2
4				247	12	12	12	12	12
5					229	10	10	10	10
6						219	206	206	206
7							13	7	7
8								6	6
$\geq$ 9									0
Mean FEC length	1.00	2	2.9	3.0	3.1	5.6	5.8	5.9	5.9
δ_{inter} [%]	0.0	7.8	14.4	92.6	95.9	15.2	97.1	97.5	100.0

The improvements in V_q by using the proposed algorithm, that are obtained by lowering the actual data rate but adding FEC protection is measured by ΔV_q, which is given by:

$$\Delta V_q = V_{q,FEC} - V_{q,w/oFEC} \tag{3}$$

The ΔV_q for various data rates is shown in Fig. 3, where it can be seen that the largest improvements can be obtained for the 5% PLR value and for the highest data rates. Note that the video bit rate in the abscissa for the $\Delta V(qFEC)$ is the video quality where the video bit rate represents the sum of FEC bit rate and video bit rate, i.e., the gross bit rate. The optimal value for k is obtained for the different PLRs and corresponding gross bit rates.

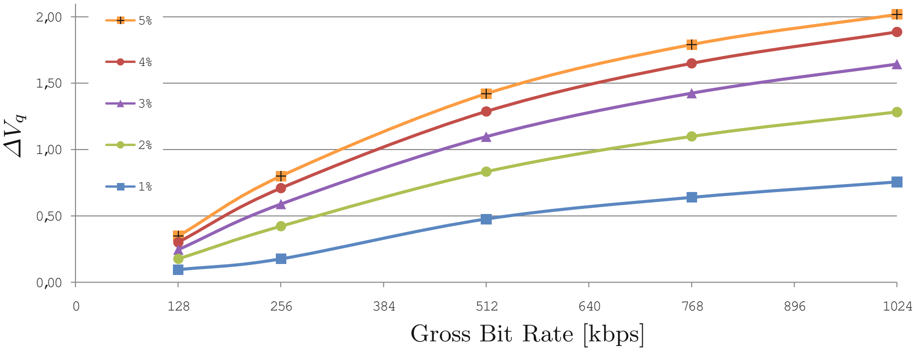
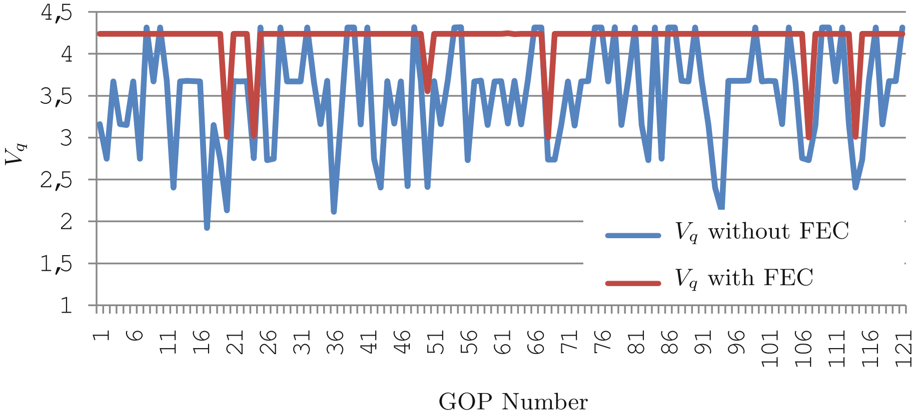
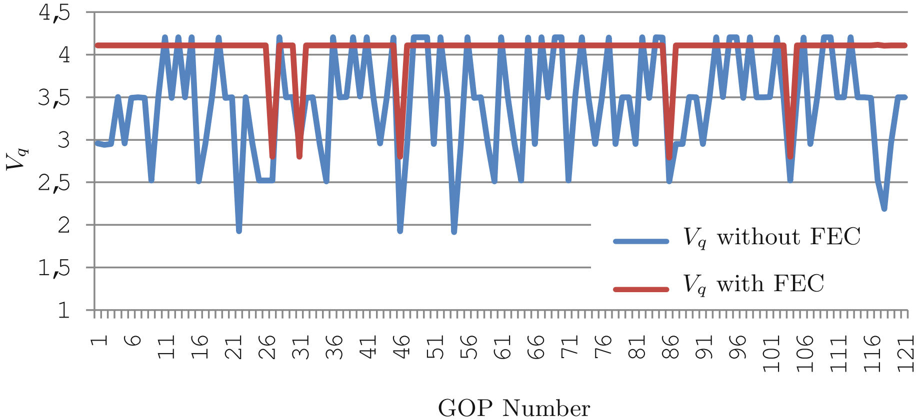


Fig. 3. ΔV_q due to FEC protection



(a) 1024 kbps.



(b) 768 kbps.

Fig. 4. V_q per GOP with a 1% Bernoulli loss model for various data rates

3.2 Simulation Results

The results describe above are based on analytical analysis, assuming a uniformly distributed error for the calculation of PLR after FEC protection. In this section the video protection algorithm is analysed based on simulation results with a Bernoulli error model and the simplified Gilbert model [16].

The simulations are performed with NS2 [17], with the myEvalvid [18] extension. The FEC algorithm is implemented in the myEvalvid framework, where the maximum value for k is set via TCL-bindings. The FEC is adapted to the length of the video frame in order to avoid inter frame delay. For example, assuming a maximum value for $k = 3$ and a video-frame of 4548 bytes. The maximum payload for UDP was set to 1024 bytes, so that 4 completely filled IP packets and one partly filled packet of 500 bytes is needed. One FEC packet is added to the first 3 IP packets of 1024 bytes each. The remaining data of 1524 bytes is too large to fit in a single packet, so 2 IP packets are needed. Since the FEC needs to protect the largest IP packet it needs to be of the same size as the largest packet. However, since the 2$^{\text{nd}}$ packet is not completely filled, a little bit can be gained (size-wise) by equally dividing the remaining data over the 2 packets. The remaining data is transmitted in 2 packets of 762 bytes each, and a FEC packet of the same size is added for protection. In this example the video frame is protected by one FEC block of length 3 and one of length 2. In this example a mere 262 bytes ($1024 - 762 = 262$) are gained in the video frame, but at zero cost.

The network is a simple direct lossless link on top of which an error model is defined. The error model works directly on the packets, in order to compare with the analytical results presented above.

Bernoulli Loss Model. The Bernoulli loss model is memory-less and has a fixed packet loss probability, which is in line with the analytical method for the calculation of the PLR after FEC protection. The red line in Fig. 4a represents the V_q with the FEC protection scheme, whereas the blue line shows the V_q of the stream using a data rate equal to the gross data rate of 1024 kbps and a PLR of 1%.

It should be noted that the FEC protected stream has a much more constant performance than the non-protected one. Only in cases that a single VOP has been received without any loss, the non-protected stream gets slightly better results, which can be explained by the fact that the non-protected stream uses a higher video data rate (in the stream without FEC protection the video data rate is equal to the gross data rate). Similar results are obtained for 768 kbps and 1% PLR, see Fig. 4b. For higher PLRs, the FEC protected stream outperforms the non-protected streams, but becomes less constant as the PLR increases. The simulated results with the Bernoulli loss model are compared with the analytical results for ΔV_q in Table 4, where it can be seen that the analytical results are a little optimistic compared to the simulated results, but both results show similar gains with respect to unprotected streams.

Table 4. Comparison of ΔV_q for various data rates with Bernoulli loss model, analytical and simulated

$Br_v[kbps]$		P_{plv} [%]				
		1	2	3	4	5
1024	$V_{q,theor}$[%]	22	46	72	97	119
	$V_{q,sim}$[%]	19.5	40.9	63.0	86.8	104.8
768	$V_{q,theor}$[%]	19	39	61	82	101
	$V_{q,sim}$[%]	17.9	37.5	53.2	72.6	90.3

As noted previously, the truncating of the FEC blocks at the end of a VOP in order not to add any delay leads to a slight overhead in data rate, see Table 5. Note that the exact value is dependent on the used video stream, but the differences are usually small. The largest overhead measured was for a PLR of 5% and a gross data rate of 768 kbps. The lowest overhead was obtained for a gross data rate of 1024 kbps with a PLR of 5%. However, the latter had an optimal value of $k = 1$, in which case there is no need to truncate the FEC as a $k = 1$ cannot introduce inter-frame delays.

Table 5. Overhead due to truncating the FEC block at the end of a VOP

Gross data rate [kbps]	P_{plv} [%]				
	1	2	3	4	5
1024	2.3	1.1	0.5	0.5	*0.0*
768	4.0	3.7	4.9	4.5	5.0

Simplified Gilbert Model. The second error model used is the simplified Gilbert model [16], which is modelled by the average packet loss ratio and the average burst error length (ABEL). Although it is possible to see burst-errors in the Bernoulli model, the ABEL cannot be controlled. The simplified Gilbert model is shown in Fig. 5, where no packets are dropped in the *Good* state and all packets are dropped in the *Bad* state.

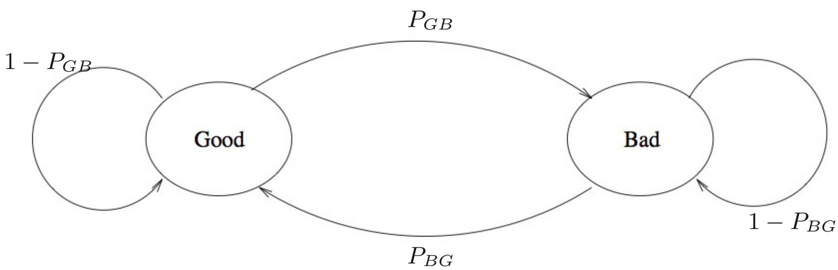


Fig. 5. Simplified Gilbert Error Model

In this case the average burst error length is equal to mean time the model remains in *Bad* state and the PLR can be obtained by:

$$\mathrm{PLR} = \frac{Bad}{Bad + Good} \tag{4}$$

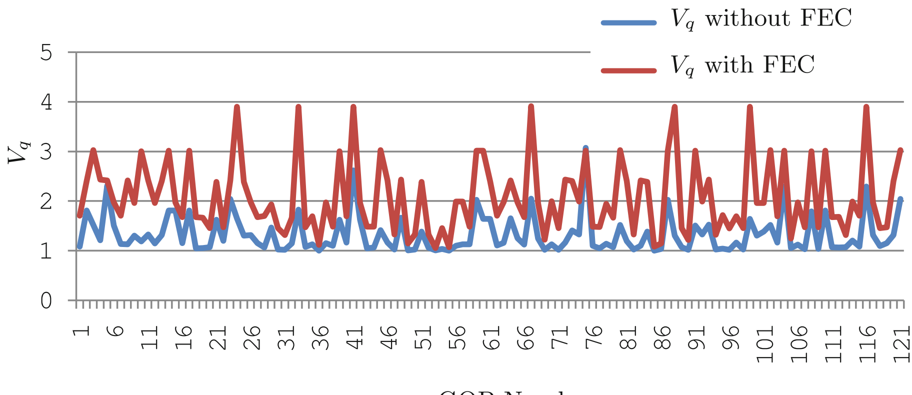


Fig. 6. V_q per GOP for a gross data rate of 1024 kbps and 10% Gilbert Error model with ABEL of 1

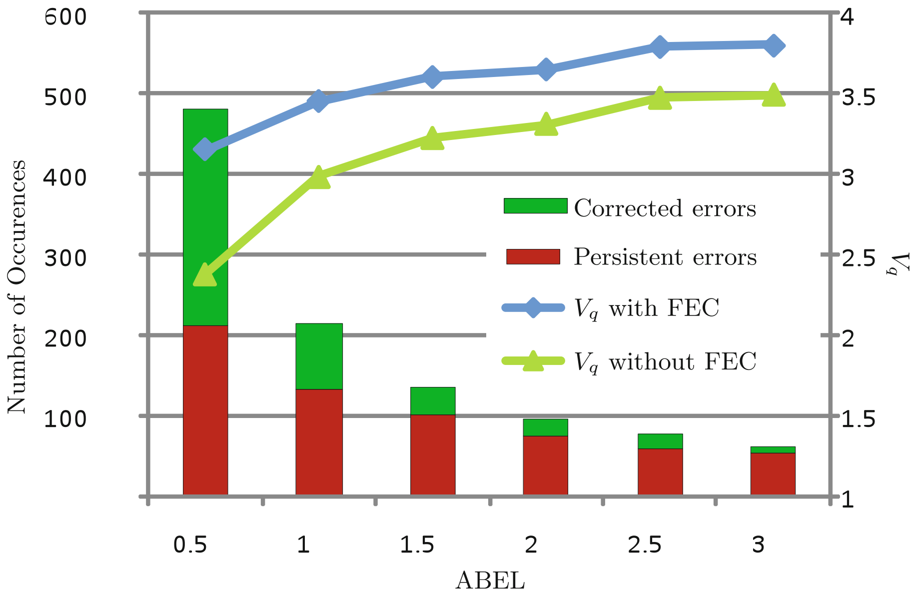


Fig. 7. V_q for a gross data rate of 1024 kbps and 1% Gilbert Error model with different ABEL

It can be expected that when the ABEL increases, the PLR after protection decreases, as the proposed parity FEC only protects a single packet loss in a block length. However, as can be seen in Fig. 6, even under bad conditions with 10% packet loss and an ABEL of 1, the protected stream outperforms the non-protected one. The effect of the ABEL on V_q are shown in Fig. 7, where the simulated curve outperforms the unprotected stream in all cases. It is interesting to note that while the ABEL is increasing and less burst can be protected, the V_q also increases. However, this is due to the fixed PLR for the simulations. In a simulation of 120s with PLR of 1%, the number of lost packets is more or less fixed. This effectively means that when increasing the ABEL, the number of occurrences of a burst error becomes smaller, i.e., less burst of errors, but of longer duration. This effect is also shown in Fig. 7, where the green bars, representing the corrected errors, becomes much smaller in comparison with the persistent errors, represented by the red bars.

4 Conclusions

The Quality of Experience (QoE) of a Video Call in emergency networks is severely degraded due to packet-loss. In this work a dynamic FEC length scheme is proposed that mitigates most of the packet losses and maximizes the QoE according to the G.1070 recommendation. The FEC scheme proposed in this work is a simple packet level FEC, which protects up to a single packet in a block. Longer FEC blocks give lower protection and data rate overhead than short blocks. In order not to introduce extra delays, the FEC block cannot contain packets from different video frames, which is guaranteed by truncating the FEC at the end of a VOP. For links with a packet loss ratio near 5%, the received video quality can often be doubled. For more reliable links, with a packet loss ratio of 1% can still achieve over 20% improvement for high data rates (1024 kbps) and almost 10% for lower data rates (256 kbps). The algorithm has been evaluated by means of simulations for the Akiyo test sequence in NS2. It was shown that the truncating of the FEC block at the end of a GOP, introduces a slight overhead in terms of data rate ($< 5\%$). It has also been shown that even when using an error model with an ABEL > 1, the proposed algorithm still outperforms the unprotected stream.

References

1. Vaz, R.N., Nunes, M.S.: Selective Frame Discard for Video Streaming over IP Networks. In: Proc. of the 7th Conference on Computer Networks (CRC 2004), Leiria, Portugal (October 2004)
2. Williams, J.: IPTV QoS/QoE Quality Model. JDSU, Tech. Rep. (2008)
3. Wu, D., Hou, Y.T., Zhang, Y.-Q.: Transporting real-time video over the internet: Challenges and approaches. Proceedings of the IEEE 88(12), 1855–1877 (2000)
4. Tan, Y., Wang, H., Wang, X., Zhang, Q.: A Video Transmission Algorithm over the Internet based on FEC and Kalman. In: IEEE International Symposium on IT in Medicine and Education, ITME 2008, pp. 263–267 (December 2008)

5. Feng, J., Xuefen, C., Li, P., Yining, W., Guan, L.: Adaptive FEC Algorithm based on Prediction of Video Quality and Bandwidth Utilization Ratio. In: International Conference on Advanced Information Networking and Applications, AINA 2009, pp. 182–188 (May 2009)
6. Chan, S.-H., Zheng, X., Zhang, Q., Zhu, W.-W., Zhang, Y.-Q.: Video Loss Recovery with FEC and Stream Replication. IEEE Transactions on Multimedia 8(2), 370–381 (2006)
7. Perkins, C., Hodson, O.: Options for Repair of Streaming Media, RFC 2354 (Informational), Internet Engineering Task Force (June 1998), http://www.ietf.org/rfc/rfc2354.txt
8. Ramsey, J.: Realization of optimum interleavers. IEEE Trans. on Information Theory 16, 338–345 (1970)
9. ITU-T, Rec. G.1070 Opinion Model for Video-Telephony Applications, ITU-T, Tech. Rep. (April 2007)
10. Li, A.: RTP Payload Format for Generic Forward Error Correction, RFC 5109 (Proposed Standard), Internet Engineering Task Force (December 2007), http://www.ietf.org/rfc/rfc5109.txt
11. Bolot, J.-C., Vega-Garcia, A.: The case for FEC based error control for packet audio in the Internet. ACM Multimedia Systems (1997)
12. Handley, M.: An examination of mbone performance, USC/ISI Research Report, Tech. Rep. ISI/RR-97-450 (April 1997)
13. Yajnik, M., Kurose, J., Towsley, D.: Packet loss correlation in the Mbone multicast network. In: Proc. IEEE Global Internet Conference (November 1996)
14. Huitema, C.: The Case for Packet Level FEC. In: Protocols for High-Speed Networks, Sophia Antipolis, France, October 28-30, pp. 109–120 (1996)
15. FFmpeg Project, Ffmpeg encoder (January 2010), http://www.ffmpeg.org
16. Gilbert, E.N.: Capacity of a Burst-Noise Channel. Bell Systems Tech. J. 39, 1253–1265 (1960)
17. Fall, K., Varadhan, K.: The ns-manual. VINT Project, University of California, Berkely, CA (2002)
18. Ke, C.-H., Shieh, C.-K., Hwang, W.-S., Ziviani, A.: An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission. J. Inf. Sci. Eng. 24(2), 425–440 (2008)