Towards Quality Evaluation and Improvement of a MPEG Video Stream

Ricardo Ferreira Martins[1]*, Carlos Alberto Brandão Barbosa Leite[2]†, Jean-Marie Farines[3]‡

[1]Universidade do Estado de Santa Catarina - UDESC, Dpto. de Ciência da Computação - DCC
Caixa Postal 631 – 89.223-100 Joinville, SC

[2]Universidade Federal do Maranhão - UFMA, Dpto. de Engenharia de Eletricidade - DEEE
Av. dos Portugueses, S/N – 65085-580 São Luís, MA

[3]Universidade Federal de Santa Catarina - UFSC, Dpto. de Automação e Sistemas - DAS
Caixa Postal 476 – 88.040-900 Florianópolis, SC

ricardo@das.ufsc.br, cbrandao@das.ufsc.br, farines@das.ufsc.br

Abstract. *The transport of a digital video with satisfactory quality over packet networks is still a challenging issue. Video Flows have contents with different importance during the coding phase, while packet networks do not provide native mechanisms that can prioritize the more important contents. In this work, we present a metric of quality that permits the evaluation of received videos, qualitatively and quantitatively. We also analyze some different techniques conducted over differentiated service networks (packet discard schemes and data redundancy, widely used in multimedia streaming) to validate our proposed metric.*

Key-words: *"QoS", "MPEG video", "metric".*

1. Introduction

Nowadays, the use of digital video is already a reality, including distributed applications like VoD, digital TV, etc. The vast majority of these applications uses video codecs based on ISO/IEC (MPEG) or ITU-T (H.26x) patterns, using well-dimensioned proprietary networks with a well-known traffic shape. On the other hand, Internet and intranets are becoming closer and closer to the characteristics of integrated service networks, where a great number of applications run and use the same service model and network structure. However, this network model is not appropriate to deal with applications that use video codecs, generating frames with different types and importance, like MPEG. This is because there is no mechanism in the best effort model that gives this level of QoS, considering and prioritizing packets with the more relevant information. We have seen in the literature [1] that packet losses in congested networks do not correspond to frame losses in the same proportion, and this can be observed through an accentuated degradation in perceptual quality of the received video.

We found many works in the literature focusing on the improvement of video transport quality. Shin et al. [2] propose a device for video packet delivery, based on a related index of priority that reflects the effect of the loss propagation of each packet. Hemy et al. [3] propose the use of filters in the nodes of the network with the aim of dropping packets selectively, according to their importance. The use of drop priority for video packet transportation has been discussed in many articles, and Markopolou et al. [4] and Ahmad et al. [5] applied it to layered video. Ziviani et al. [6] have shown a schema for delivery based on drop priority (using one metric that imposes a video packet decoding index, taking into account the codec characteristics) and an error recovery schema.

In this article, we try to show how techniques based on the utilization of different discard levels for video frames, forwarded through a differentiated service network (DiffServ), can sensibly improve the quality of received video based on a new metric that evaluates the quality of received **GOPs** (Group Of Pictures - set of frames that defines the temporal and spatial relations, directly related with the video compression factor), only considering frames that are used during the decoding phase. This metric is also used to evaluate the transmission of a video stream containing redundant data, with the aim of tolerating some losses over the end-to-end path.

This article is organized in the following way. In section 2, we discuss the hierarchical structure, as well as questions related to packaging of video flows. Section 3 shows the adopted approaches, with the aim of improving the delivery quality of a video flow, emphasizing video packet protection mechanisms (taking into account the

*Supported by UDESC, UFSC and CAPES.
†Supported by UFMA, UFSC and CAPES.
‡Supported by UFSC and CNPq.

importance of frames during the decoding phase) and data redundancy. In section 4, we present our metric used to qualify and quantify video frames. Section 5 presents the details of the environment and topology used in our simulations, as well as the results achieved with the proposed metric. Finally, in section 6, the conclusions and perspectives of this work are presented.

2. The Structure and Packaging Process of a Video Stream

A video stream consists of a sequence of frames, where each frame is composed of a matrix of pixels that describes a scene. The top-level video codecs use temporal and spatial relationships to generate frames, and these frames have different types and meanings during the decoding phase. In this context, we can cite the families of codecs: MPEG - *Moving Picture Experts Group* (MPEGx) and ISO/IEC (H.26x). The structure of a MPEG stream is shown in fig. 1, where frames are generated in different types (I, P and B). Frames of type I are coded as an image describing the scene. Frames of type P are predicted from the last reference frame (it can be a frame I or P). Frames of type B are predicted based on references of previous and next frames (they can be frames I or P). Thus, fig. 1a shows the structure of a frame, while the structure of a GOP (fig. 1b) is characterized by the distance between two consecutive I frames (represented by N), and the distance between an I frame and the first P frame, in the sequence (represented by M).

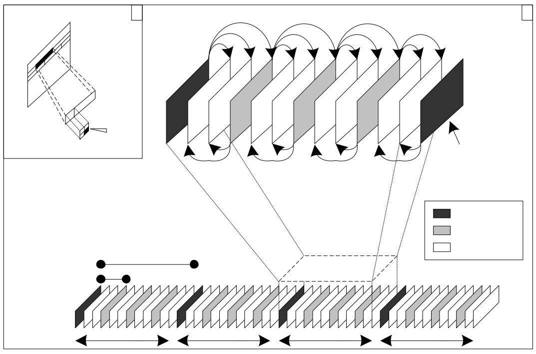


Figure 1: Hierarchical structure of a video stream: (a) frame structure; (b) dependencies between frames.

A video frame is composed as a series of **slices** (in MPEG) and **GOB** (*Group Of Blocks*, in H.26x). They are transmitted in the transport protocol payload. The way these frame slices are transmitted is called **packaging**. Generally, the packaging technique tries to maximize or minimize some metric related with losses, because each slice of different packet type (I, P or B) transports information with different meanings for the decoding phase. The corruption or loss of slices of type I compromises the rest of the slices of other types, inside a GOP. On the other hand, some packaging techniques work with error recovery. The main goal of packaging is to produce video packets as large as they can be, addressing the increase of the relation between header information and payload. However, it is not desirable to transmit more than one frame in each packet, due to the increase in the delay that this implies. Another fact that we have to take into account is the minimum size of **MTU** (*Maximum Transfer Unit*) along the end-to-end path, because the use of packets with larger sizes than the smallest MTU along the path implies fragmentation and their reassembly, generating an additional overhead. The ideal solution would be a minimum or total absence of packet fragmentation, but only in cases where there is no data redundancy.

Considering that the majority of networks, where the end systems are connected, are *ethernet* (representing a MTU of 1500 bytes), it is reasonable to think of using a packet size close to 1500 bytes, to transport a video stream. For example, if we choose packets with 1450 bytes, or 11600 bits per packet, it is feasible to presume that many frame formats can fit in this packet. Thus, in the case of a video generated with 10 frames/sec, the resulting bit rate will be 116000 bits/sec, and using the maximum size of packet payload (1450 bytes), this will be enough to accommodate a QCIF frame type (176×144 pixels). Larger frame types, like CIF (352×288 pixels) are possible, but with some degradation. These numbers suggest a rule to minimize packaging overhead, and one frame per packet is a good choice. However, with this rule, the loss of a single packet implies the loss of an entire frame.

From the point of view of error recovery, it would be more desirable to split the coded frame into a great number of packets, trying to keep the area spatially affected as small as possible, in the case of losses. Wenger et al. [7] propose a schema for packaging that uses two packets per frame, addressing a trade-off between error recovery and overhead reduction. Other packaging schemas, like transporting slices of different frame types in the same packet, are susceptible enough for losses, increasing the degradation of decoded video or even disallowing its decoding. In the context of this work, we will deal with MPEG4 coded videos, transported directly over UDP, with a variable relation between frame $\times$ packet.

3. The Approaches Adopted to Improve the Video Delivery Quality

An eventual schema for digital video transportation must be able to distinguish the more important slices of information and prioritize them against the less important slices and non-multimedia traffic. A codec structure based on MPEG and ISO/IEC patterns generates frames of different semantic importance for the decoding phase. The temporal and spatial relations used by compression techniques generate different types of frames with decreasing semantic importance for decodification of a video flow. In this way, it is interesting that the network implements mechanisms taking into consideration the importance of different types of frames and prioritize them during the forwarding process. However, such a solution may not be effective in some cases, for example, when the network is very congested, and we can adopt another technique trying to complement the first.

3.1. Mechanisms for Video Packet Protection

In congested situations, the hierarchical structure of a video stream causes promiscuous packet discard, when transported over a pure best effort network. This discard reduces severely the comprehensibility of received content, because frame fragments that have higher importance during the decoding phase have the same probability of being discarded, compared with less important frame fragments. An ideal situation would be to distinguish and give higher priority to packets that transport more important contents. A good option is to associate levels of discard priority to each packet, based on its importance (this is relatively simple) and it helps to maintain a better quality of received GOPs.

The QoS network architecture that attends the requirements indicated above can be a DiffServ network, like the one presented in Blake et al. [8]. In this kind of network, each packet is associated with an identification code (*code point*) that can be related to a procedure of discarding or forwarding. This allows, for example, the association of levels of discarding precedences or forwarding priority, for each packet. We can find in Assured Forwarding Services (AF) levels of discarding precedence associated with each packet, proposed by Heinanen et al. [9]. In this case, the service differentiation uses four priority classes, each class having three levels of discard.

In this context, DiffServ networks, like the architecture presented in Blake et al. [8], and implementing services like those proposed in Heinanen et al. [9], can configure a scenario suitable for transportation of digital video. An AF class can forward video packets following a mapping of type: Ax_1 = I frames , Ax_2 = P frames and Ax_3 = B frames. The mapping of packet type into discard priority can be performed by the application itself or by another edge device of the domain (in a video gateway, for example). Heinanen et al. [9] suggest the implementation of queue mechanisms and discard behavior as a way of trying to minimize long time congestions inside each class. This requires an active queue management algorithm (**AQM**). A **RED** mechanism (*Random Early Drop*), like that proposed in Floyd et al. [10], and implemented in multiple levels, can satisfy well this requirement. The utilization of multiple RED mechanisms was proposed by Clark et al. [11], where each RED queue controls classified packets as belonging or not to a specific profile (*in* or *out profile*). To work with video packets, we can extend the mechanism to a RED of three levels, associating each type of packet to a specific queue, as shown in fig. 2. With this mechanism, less important packets will have a higher probability of discard, while the more important packets, consequently, will have a lower probability of being discarded. The different levels of discarding precedence are also identified as *green* = Ax_1, *yellow* = Ax_2 and *red* = Ax_3, running from the minor to the major discard probability, respectively. Approaches of this nature for video transportation have been used in many works, such as [6] and [12].

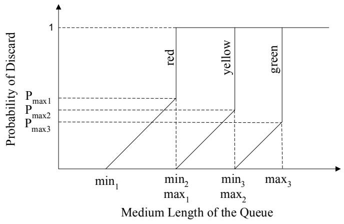


Figure 2: RED mechanism with three levels

We will show in section 5.1 some simulation results, but it is already expected that this mechanism won't solve all problems observed when the network is really overloaded. For example, received GOPs can have only frames of type I. Even with a minimum GOP loss rate, we can not guarantee that the video will be presented with a good quality. For this reason, we will complement these mechanisms with another technique, discussed in the next section.

3.2. Data Redundancy

As we said early, data losses are particularly inconvenient when we discuss applications involving video streaming, and we can use mechanisms that promote data redundancy during the coding phase, with the aim of decreasing the probability of losses from the application level.

The more common schema for data redundancy is known as **FEC** (Forward Error Correction), where all blocks of k packets are followed by $n - k$ redundant packets, composing a FEC packets block, with size of n. Thus, if at least k packets were received, whatever the order, then the information associated to this block will be decoded correctly. In the case of a multimedia flow, if a fixed minimum data percentage (dt – decodable threshold) of a frame is received, then it will be considered decodable. In [13], the overload (Ov) generated by the redundancy FEC schema is obtained by $Ov = \frac{1-dt}{dt}$.

For example, with $dt = 0.75$ and $dt = 0.5$, we will have an overload equal to $Ov = 33.3\%$ and $Ov = 100\%$, respectively. This means that, for each frame of the video flow, we will have an extra data transmitted, and the packets must be classified as the same type associated to the frame. The redundant data will be generated by the codec, and classification will be carried out by the DiffServ network mechanisms.

The combination of packet protection and data redundancy can be achieved using a monitor process that observes the losses, directly related to network overload, and decides which dt value is applicable. In fact, our intent here is to show that our proposed metric can be used to monitor the video quality.

We will show in section 5.2 how the utilization of data redundancy is directly related with the MTU size over the end-to-end path, and also the need for frames to be fragmented into packets.

4. A metric of Quality for Video GOPs

The utilization of packet loss rate as a metric adopted to evaluate the quality of presented video is not satisfactory, because it is not sufficient to know only how many packets were lost, associated with each type of frame. Thus, we need to evaluate which frames were received, belonging to a specific GOP, and if these frames will be used at the moment of presentation, according to frame dependence rules. For example, if a GOP made of $IBBPBBPBBPBB$ is transmitted, and the received frames were $IB - PBB - BBPBB$, then, at the moment of presentation, frames presented will be $IB - P - - - - - - - -$, where "$-$" means the absence of a frame during the presentation. In this case, a B frame and a P frame were lost (direct losses), and these losses imply the discarding of six B frames and one P frame (indirect losses), despite their being received successfully.

With the objective of evaluating video quality, at the moment of presentation, we define a metric based on what is called the *video quality factor* (*q*). The main idea is in the calculation of *q* for each video GOP, following equation 1:

$$q = \frac{a_I * x_I + a_P * x_P + a_B * x_B}{a_I * N_{TI} + a_P * N_{TP} + a_B * N_{TB}}, 0 \leq q \leq 1 \tag{1}$$

where j is the type of frames (I, P or B), x_j is the number of j frames received and displayed, N_{Tj} is the total number of j frames in a GOP and a_j is the relative coefficient (j frames in a GOP), following equation 2:

$$a_j = \frac{N_{Ij}}{N_{II} * N_{TI} + N_{IP} * N_{TP} + N_{IB} * N_{TB}}, \quad N_{Ij} : \quad \text{direct and indirect losses } (j \text{ frames}). \tag{2}$$

The value of *q* is limited by two situations: first, when all frames in a GOP are lost ($x_I = x_B = x_P = 0$), directly or indirectly, implying $q = 0$; and second, when all frames in a GOP were received ($x_I = N_{TI},\ x_B = N_{TB}\ e\ x_P = N_{TP}$), implying $q = 1$. Another relevant point is that equation 1 takes into account indirect losses, moreover, variables x_I, x_B and x_P consider as received frames only those that will be used during the decoding phase. For example, if an I frame of a GOP is lost, then we will have $x_I = x_B = x_P = 0$, independent of the number of B and P frames successfully transmitted (indirect losses).

As an example, for a GOP where $N = 12$ and $M = 3$, we will have the following values: $N_{TI} = 1$, $N_{TB} = 8$, $N_{TP} = 3$, $N_{II} = 14$ (all frames of the GOP, plus the last two B frames of the previous GOP), $N_{IB} = 8$ (all B frames) and $N_{IP} = 11$ (all B and P frames). Thus, following the previous example, the value of *q* would be equal to 0.297, because $x_I = 1$, $x_B = 1$ and $x_P = 1$.

In fact, *q* has a relative association with the quality of the presented video, mainly for B frames. In other words, a loss of a B frame will imply different qualities, depending on the position of this frame in the GOP and its importance associated to the temporal resolution. Our formula considers that B frames have the same value associated to the quality, but alternatively, we can approximate the *q* value to the real quality, giving a weighting for each B frame (based on the number of bits, for example). However, it is dependent on the way the codec movement estimator finds the movement vectors when a scene change occurs, which can imply contradictory conclusions for B frames.

5. Evaluation of the Approaches

For the verification of the transmission schema and the analysis of our proposed metric, we adopted a method based on simulations, using NS-2 [14] (with some modifications). The topology adopted for all simulations is according to fig. 3, where the video stream is sent from a source to a receiver, crossing a DiffServ network that has a bottleneck (between routers 2 and 3). In this scenario, we maintain a MTU of 1500 bytes along the entire end-to-end path.

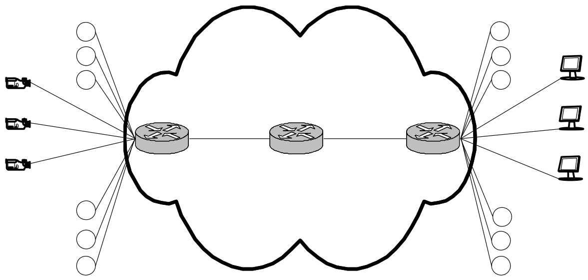


Figure 3: **Topology used in simulations.**

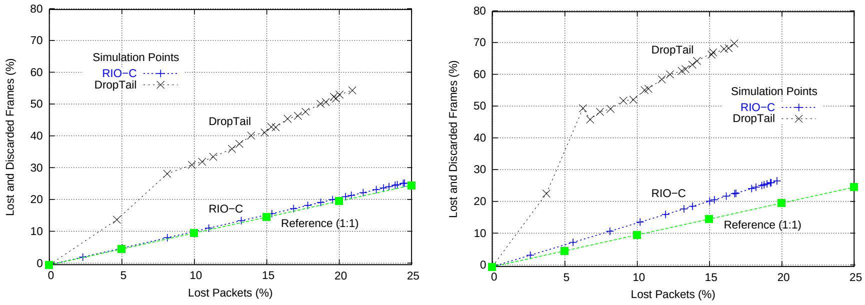
We use video streams coded in MPEG-4 based on trace files available in [15], comprising basically ASCII files with information (generation time, type and packet size). We also use trace files of two distinct videos, and the difference is in its coding quality (lower and higher quality). The coding approach for low video quality fixes the quantization parameters to 10 for I frames, 14 for P frames and 18 for B frames. For high video quality, the quantization parameters are fixed to 10 for all types of frames. In both videos, the GOP size is fixed and equal to 12. The maximum size of video frames generated is 4686 bytes and 9370 bytes for lower and higher quality, respectively. Consequently, this implies that there is a possibility of fragmentations occurring over the network. Furthermore, for a frame to be considered interpretable, all data associated to it must be received, and if not, it will be considered lost (without error recovery).

In the DiffServ network, the configuration used has only one real queue, to which is applied a specific policy through an active queue management (AQM), implementing a protection over packets (using a RED mechanism with three levels), known as MRED, and based on a RIO-C algorithm [16]. Each level is associated with a discard precedence and has a virtual queue. This configuration allows a treatment of short duration congestions, situated at specific points of the network (and in this particular case, at the bottleneck link).

Two types of background traffic were used: elastic applications based on FTP flows over TCP, and video flows over UDP (with the same characteristics of the main video flow), focusing an evaluation of main video stream quality against the presence of competitive flows, in addition to the characteristics of video used and queue policy adopted. The FTP/TCP traffic was injected in the same virtual queue, where all B packets were forwarded, thus having a higher discard probability. For all simulation results, the values for the confidence interval were 97% and 94%, respectively.

The characteristics of trace files (GOPs with $N = 12$ and $M = 3$) and technology used provided in all simulations direct packet losses always associated to distinct frames, meaning that the ideal relation between lost packets percentage and lost frames percentage is in the order of 1:1 (each lost packet implies a loss of only one frame). However, such a relation will be observed only in situations where we do not have indirect losses, in other words, losses will be only of B frames. Fig. 4 shows this relation, for simulations with FTP/TCP background traffic up to 20 competitive flows, where we evaluate two queue policies (DropTail and RIO-C) and two qualities of transmitted video.

In fact, if the behavior of a flow presented by a sender approximates to the ideal relation (continuous line indicated in the graphics), it implies that indirect losses were minimum, which is the case of simulations with



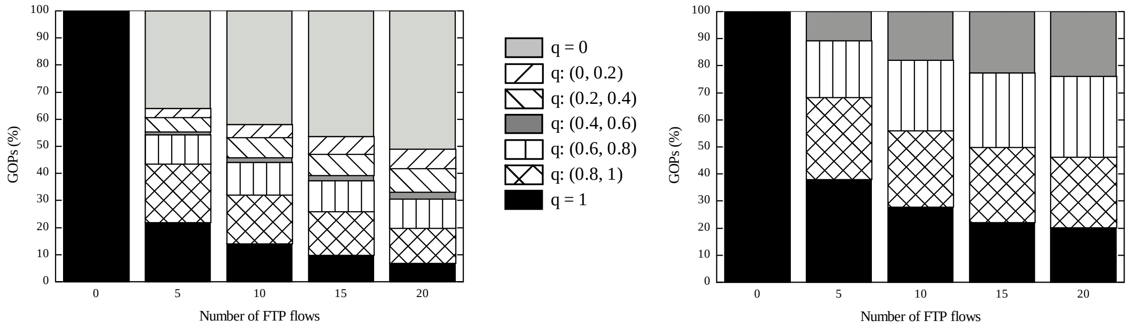
(a) Low Quality Video (b) High Quality Video

Figure 4: Relation between lost packets percentage and undecodable frames percentage, varying the number of competitive flows (FTP).

RIO-C policies. Analyzing the graphics, it is possible to observe that the results for high quality video have a significant difference, compared with other experiments, and this is because of the packet fragmentation rate, that has influence over indirect losses (there are losses of I and P frames).

5.1. Simulation Results Related to Packet Protection

The graphics presented in fig. 5 and 6 show the percentage of GOPs received, with their respective qualities, expressed through our metric (q), for different types of background traffic, queue policy and quality of main video. We present in the graphics only some values of the entire simulation (for 0, 5, 10, 15 and 20 competitive flows), but we will comment the behavior considering all the interval (from 0 to 20).



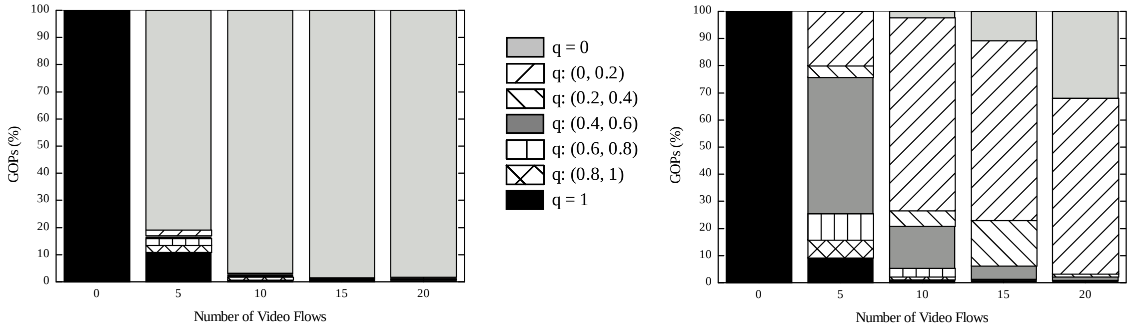
(a) DropTail (b) RIO-C

Figure 5: Simulation of a video transmission, varying the queue policy (bottleneck link) and the number of competitive flows (FTP) – high quality video.

To analyze the q value for each GOP, we made the same simulations described previously. Fig. 5 shows the results for competitive FTP/TCP flows, where the graphics represents, for high video quality and different queue policies, the relation between the GOPs percentage with maximum quality ($q = 1$) and the GOPs percentage with null quality ($q = 0$), and the other results, classifying the rest of GOPs inside each interval, with a variation of 0.2. We also made the same simulations with low video quality, obtaining similar results. For DropTail simulations, up to 28% (low quality video) and 50% (high quality video) of all GOPs were completely lost ($q = 0$), approximately. It is also possible to observe that, in the worst case and in all simulations with RIO-C, 20% of all GOPs were received with total quality ($q = 1$), and the minimum quality observed in the rest of GOPs was above 0.4. Besides this, when the number of GOPs with $q = 1$ starts decreasing, almost a half of the rest of GOPs has the q value greater than 0.7. Such results show that the protection mechanism adopted is efficient, mainly because there is no GOP loss ($q = 0$).

As the protection mechanism promoted the absence of P frames losses, the received GOPs during the simulations with RIO-C had all I and P frames, characterizing a minimum quality achieved by the video transmission, and in this case, independent of the quality of transmitted video.

The results presented in fig. 6 show how competitive video flows affect the quality during the presentation of video, and the behavior of the main flow is similar to the other flows. The results obtained were presented in the same manner explained previously, and although the competitive flows were transmitted over UDP (reducing part of the load along the end-to-end path, because of acknowledgement packets), the equivalent treatment of all flows by the queue policy adopted had a big impact over the quality of the main video, and also over the other flows.



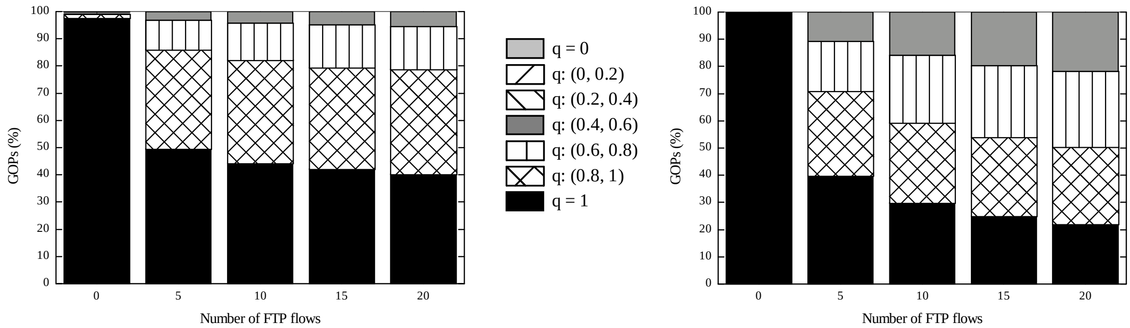
(a) DropTail (b) RIO-C

Figure 6: Simulation of a video transmission, varying the queue policy (bottleneck link) and the number of competitive flows (videos) – high quality video.

We also made some simulations for low quality video, and the impact of the adopted queue policy was less than for high quality video. With 10 background video flows, almost 85% of GOPs had $q = 1$, but with 20 flows, this value decreased to 13%. In addition, all results for competitive video flows also indicate one situation that was not observed in simulations with competitive FTP flows – the saturation point. In other words, it is possible to observe that when we achieve the number of 10 competitive video flows, it starts a considerable decreasing in GOP's quality, even with the protection mechanism. This is because of two reasons: first, all the flows have packets with the same discard priorities; and second, the less elastic characteristic of the transport protocol used (UDP), since TCP (used in FTP flows) allows a better smooth degradation in quality, referring to the control of transmission rate.

5.2. Simulation Results Complemented with Data Redundancy

We run the same simulations described in previous section, but now complementing with data redundancy (FEC). Two situations were considered: first, when video flow has $dt = 0.75$, and second, for $dt = 0.5$. Analyzing the results for a high quality video and RIO-C queue policy, presented in fig. 7, and comparing them with fig. 5b, we can observe that there is a minimum value achieved for q during all the simulations ($q > 0.4$) and we can conclude that the network was not saturated. More over, it could be possible to use a FEC schema, but the dt value also depends on the frame size and MTU along end-to-end path. For a frame size lesser than MTU (even including FEC overload), then each packet will carry one frame (this is the case of fig. 7b and also for simulations with low quality video), and the use of this mechanism does not improve the quality, generating only an unneeded network overload. On the other hand, for a frame size greater than MTU (fig. 7a), the FEC schema improves the quality.



(a) FEC: $dt = 0.5$ (b) FEC: $dt = 0.75$

Figure 7: Simulation of a video transmission, using a RIO-C queue policy and FTP competitive flows (with data redundancy).

We can conclude that if the network is congested without the use of a FEC schema, then we can easily obtain worst results applying such a schema because the introduction of redundant data will only increase the overload, and this was the case in our simulations using video flows as background traffic. However, with our metric, we can monitor the quality of received video, and analyzing its results together with MTU and frames sizes, it is possible to decide the use or not of a FEC schema.

6. Conclusions

In this work, we propose a metric that allows us to qualify and quantify the GOPs quality, signaling in the way of how to obtain a better perceptual video quality. In a general way, one approach like this can work in a synergetic way, together with other devices (reactives or not), trying to improve the quality of video streams transmitted over packet networks. We also show the evaluation of the transport of a digital video over DiffServ networks, assigning discard priority according to the importance that each frame has during the decoding phase, using the proposed metric, as well as the use of a data redundancy schema, focusing on a lower frame loss probability.

From the point of view of adaptation, associated with network load variation, we believe that our metric can be used as a feedback variable, indicating to the sender how the transmitted video has been received, as an approximation to the quality perceived by end-users, compared with a single analysis of packet loss rate. The adaptation could be achieved as much through changes on transmitted video quality (for example, changing the quantization factor), as through a new definition of the enhancement layers to be transmitted (layered video), or using error recovery schemas. In all cases, we must observe that if the composition of GOPs (N and M) has changed, then it should have a compensation mechanism that operates during the transition phase.

We can compare our metric with others found in literature, but only the results presented by Ziviani et al. [13] are similar. Although both metrics are based on decodable frames, there is a significant difference. For example, if we have two different sequences of GOPs (first, $IBBP--PB-P--$, $------------$, $IBBP--PB-P--$, and the second, $IBBP--P--P--$, $IBBP--P--P--$, $I--P---------$), their metric analyzes the decodable frames percentage, and the results will be the same for all GOPs in both situations, because there is the same number of decodable frames either in the first sequence or in the second. In fact, the quality is not the same, mainly if we have many scene changes, while our metric can detect different GOPs qualities, and its evaluation has a better approximation with real video quality.

Future works will be focused on the validation of our metric based on end-user perceptions. We will also study the use of our metric as a feedback variable for control mechanisms, interacting with adaptive applications.

References

[1] J. Boyce and R. Giglianello, "*Packet loss effects on MPEG video sent over public Internet*," in *Proc. of ACM Multimedia*, A. Press, Ed., 1998, pp. 181–190.

[2] J. Shin, J. Kim, and C. C. J. Kuo, "*Content-Based Packet Video Forwarding Mechanism in Differentiated Service Networks*," *Proc. of Packet Video Workshop*, May 2000, Sardinia, Italy.

[3] M. Hemy, U. Hengartner, P. Steenkiste, and T. Gross, "*MPEG System Streams in Best-Effort Networks*," *Proc. Packet Video Workshop'99*, May 1999.

[4] A. Markopoulou and S. Han, "*Transmitting Scalable Video over a DiffServ Network*," Internet article, March 2001, http://ise0.stanford.edu/class/ee368c/Projects/project06/,(May2003).

[5] T. Ahmed, G. Butidant, and A. Mehaoua, "*Encapsulation and Marking of MPEG-4 Video over IP Differentiated Services Commentaire*," *IEEE Symposium on Computers and Communications - ISCC2001*, 2001.

[6] A. Ziviani, J. F. de Rezende, O. C. M. B. Duarte, and S. Fdida, "*Improving the Delivery Quality of MPEG Video Streams by Using Differentiated Services*," *2nd European Conference on Universal Multiservice Networks - ECUMN'02*, pp. 107–115, April 2002.

[7] S. Wenger, G. Knorr, J. Ott, and F. Kossentini, "*Error Resilience Support in H.263+*," *IEEE Transaction on Circuit and Systems for Video Technology*, vol. 8, no. 7, November 1998.

[8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "*An Architecture for Differentiated Services*," Internet RFC, December 1998, RFC 2475.

[9] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "*Assured Forwarding PHB Group*," Internet RFC, June 1999, RFC 2597.

[10] S. Floyd and V. Jacobson, "*Random Early Detection gateways for Congestion Avoidance*," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.

[11] D. D. Clark and W. Fang, "*Explicit Allocation of Best Effort Packet Delivery Service*," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, August 1998.

[12] A. Ziviani, J. F. de Rezende, O. C. M. B. Duarte, and S. Fdida, "*Vídeo MPEG sobre serviços diferenciados*," *Anais do IX Simpósio Brasileiro de Telecomunicações*, Setembro 2001.

[13] A. Ziviani, B. E. Wolfinger, J. F. de Rezende, O. C. M. B. Duarte, and S. Fdida, "*Joint Adoption of QoS Schemes for MPEG Streams*," *Multimedia Tools and Applications*, October 2002.

[14] UCB/LBNL/VINT, "*The Network Simulator - ns (version 2)*," http://www.isi.edu/nsnam/ns/,(May2003), 2001.

[15] F. H. Fitzek and M. Reisslein, "*MPEG-4 and H.263 Video Traces for Network Performance Evaluation*," http://www-tkn.ee.tu-berlin.de/research/trace/trace.html,(May2003), 2000.

[16] R. Makkar, I. Lambadaris, J. H. Salim, N. Seddigh, B. Nandy, and J. Babiarz, "*Empirical Study of Buffer Management Scheme for DiffServ Assured Forwarding PHB*," *IEEE ICCCN2000 - International Conference on Computer Communications and Networks*, October 2000.