Selective Frame Discard for Video Streaming over IP Networks

2 authors:

Ricardo N. Vaz

Mario Nunes
Inesc-ID

Proceedings of the 7th Conference on Computer Networks (CRC2004), FCCN, Leiria, Portugal, October 2004

Selective Frame Discard for Video Streaming over IP Networks

Ricardo N. Vaz[1, 2], Mário S. Nunes[1, 3]

[1]INESC/INOV/IST, R. Alves Redol, Nº9, 1000 Lisboa, Portugal

[2]ricardo.vaz@inov.pt

[3]mario.nunes@inov.pt

Abstract

In this paper we present two new selective frame discard algorithms for live video streaming over IP networks. Both algorithms consist in pre-emptively discarding frames at the sender to achieve the lowest impact possible on quality whenever congestion occurs. The video is streamed using the DCCP protocol, which has congestion control mechanism built in, as described in the paper. The performance of the algorithms was obtained using computer simulation. The results obtained show that both algorithms achieve higher PSNR and in some cases lower end-to-end delay when compared with no selective discard. It is also observed that the results obtained by the algorithms are similar for both low and high movement video content.

Keywords: Video Delivery, Congestion Control, Selective Frame Discard, Peak Signal-to-Noise Ratio, Delay Threshold.

1. Introduction

Today's demand for video availability anywhere is growing very fast. Television broadcasting, distance learning, video-on-demand, are examples of multimedia applications that need to transport the video remotely from sender to receiver. One way to provide this transport service is to use an IP network. However, current IP networks do not provide Quality of Service (QoS), which is an important obstacle to overcome in order to assure a proper delivery of real-time video, with stringent delay, bandwidth and packet loss requirements.

Currently, several efforts [1] are being made to either enhance the IP networks in order to provide QoS, or to achieve better performance with the present IP networks. To allow QoS to be provided in an IP network, Integrated Services (Intserv) [2] and Differentiated Services (Diffserv) [3] are possible solutions. However, these protocols require modifications in the routers/switches, which are intended to be as simple as possible.

Another approach is leaving to the end systems the task of adapting to the network conditions and mitigating the effects of not having a QoS-enabled network. This approach, followed in this paper, has the advantage of not needing any modification of the routers/switches and also of being independent of QoS upgrades of future IP networks.

The Datagram Congestion Control Protocol (DCCP) [4] is a new protocol currently being defined by the Internet Engineering Task Force (IETF) [5], which provides congestion-controlled flow of unreliable datagrams. Currently real-time applications use Real-Time Protocol (RTP) [6] over UDP whenever this protocol is allowed. The RTP packets are carried in UDP [7] packets and then delivered across the IP networks. Neither the RTP nor UDP has any kind of congestion control mechanisms. This can cause the flooding of packets by a miss-behaved application since there is no fairness in bandwidth share. If between sender and receiver exist firewalls that do not allow the use of UDP protocol due to security reasons there is no alternative then to use Hypertext Transport Protocol **(**HTTP)[8], over TCP [9], which has the congestion control mechanisms as defined by [10]. However, this solution can cause excessive end-to-end delay due to the retransmission mechanism of TCP, which can be unacceptable for real-time application. Note that if a packet is

delayed more than a certain amount of time, say 1s, the receiver will probably discard it since it has arrived too late to be decoded and displayed.
However congestion-controlled protocols may still not be enough to provide an acceptable QoS for video applications. In case of congestion the TCP or DCCP diminish the amount of data to be sent causing the increase in the occupation of the sender queue. Consequently end-to-end delay will rise and the packet discarding can occur in the sender queues, which have limited size. Therefore, a discard scheme could be applied to the queue in order to decrease the number of packets inside the transmitter queues, avoiding its overflow. This scheme should also take in account in the selection of the packets to be discarded, the impact that each of those could have on the perceived quality.
In this paper we present two different types of *selective frame discard* algorithms using MPEG-4 [11] video that address this problem, named *selective block discard* and *selective VOP discard*. Both of them give lower discard priority to I frames, which have no temporal dependence of precedent of future frames, and higher discard priority to dependent frames, e.g.: P frames. Dropping P frames before I frames maximizes quality and can eventually decrease end-to end delay. For similar reasons B frames have higher discard priority than P frames.
The current study was performed in the scope of the OLYMPIC project - Olympics Multimedia Personalised for the Internet Community [12], partially funded under the Information Society Technologies (IST) European Commission Programme. The OLYMPIC project intends to define, implement and integrate novel end-to-end network solutions and multimedia coding techniques, towards the realization of a decentralized system capable of capturing, encoding and distributing hundreds of personalized audio and video streams from live sources across the Web to multiple recipients.
This paper is organized as follows. Chapter 2 presents the related work. In chapter 3 the DCCP is presented. Chapter 4 presents the two types of proposed *selective frame discard* algorithms, whose evaluation results are presented and discussed in chapter 5. Finally, chapter 6, concludes the paper and presents possibilities of future work.

2. Related Work

As explained in [1] the heterogeneity in present IP networks can be divided in two kinds: network heterogeneity and receiver heterogeneity. The first one refers to the different sub-networks that compose the entire IP network and which could have different resources (e.g., bandwidth and packet loss ratio), resulting in a different experience in end-to-end network packet loss or delay for different users. The second kind of heterogeneity, namely receiver heterogeneity, is related with the capabilities and needs of the receiver's equipment (e.g., varying latency requirement, visual quality requirements and/or processing capability).
To address the above technical challenges, two major approaches have been proposed in the literature. The first one, the network-centric approach, assigns the task of providing Quality of Service (QoS) support to the routers/switches in the core network which have to guarantee bandwidth, maximum delay, delay jitter, and packet loss, to the video applications (e.g., Integrated Services[2], [13], or Differentiated Services [3], [14], [15], [16]). The second approach, end systems-based, does not impose any requirements on the network. The end systems employ control techniques in order to maximize the video quality without any QoS provided by the network.
The *end system-based* approach has been extensively studied and several solutions have been proposed. Those can be subdivided into two main areas: *congestion control* and *error control*.
The *error control* approach takes into account the unavoidable packets losses in the Internet, which may have significant impact on the perceptual quality. Basically, there are four types of techniques to mitigate the effects of packet losses on quality, which are: *forward error correction* (FEC), *retransmission*, *error resilience*, and *error concealment*.
The *congestion control* approach aims to mitigate the existence of bursty losses and excessive delay caused by network congestion, which has severe effects on the perceived receiver's quality. The

congestion control techniques can be described in three types. The first, known as *rate control* [17], consists in matching the rate of the video stream to the available rate provided by the network. The second type is *rate-adaptive video encoding* [18], which consists in forcing the source to send the video stream at the appropriate rate, based on a rate control algorithm. The third type of congestion control, also known as *rate shaping* [19], takes into account the available network bandwidth to apply a pre-emptively selective frame discard at the sender. In particular a selective frame discard scheme is presented in [20], which consists in pre-emptively discarding frames in the sender queue. However this scheme is designed to work with scalable video, which is not available in the majority of video servers.

In this paper we focus on the *congestion control* approach, more exactly in using DCCP [4] (a *rate control* technique) to transport the video data complemented with new selective frame discard algorithms (*rate shaping* techniques), to achieve both congestion control and higher quality of the received video. The selective discard algorithms presented in this paper can be used either in unicast or multicast. The algorithms also work with elementary video bit streams and can likely be applied to scalable video bit streams.

3. Datagram Congestion Control Protocol

The Datagram Congestion Control Protocol is a new transport protocol being defined in the IETF that provides a congestion-controlled flow of unreliable datagrams. In delay-sensitive applications, such as streaming media and telephony, timeliness to reliability is preferred. Such is the case of OLYMPIC audio and video streaming, where optimal transmission of media is the main goal.

Real time media transmission involves the generation of a packet flow, where each packet generated should be delivered as soon as possible or not delivered at all. The time constraints involved are more important than reliability, so media transmissions typically use transport protocols like UDP, where no retransmissions occurs, providing minimal packet delay. But the lack of congestion control mechanisms in UDP makes this protocol a risky choice, because there is no way to prevent a packet network flooding with a miss-behaved application and there is no fairness in bandwidth share. Presently the only alternative is to use the TCP transport protocol, to provide congestion control mechanisms, but the retransmission mechanism is a strong disadvantage due to the high end-to-end delay that it causes. DCCP combines the best of the two protocols within media transmission context, supporting congestion control mechanisms with no retransmissions.

DCCP uses Congestion Control Identifiers (CCID) to determine the congestion implemented control mechanism. Currently two identifiers are being defined: the CCID 2 [21] that implements a TCP-like Congestion Control and CCID 3 that implements a TCP-Friendly Rate Control (TFRC) [22].

CCID2 sends data using a congestion control mechanism based on the selective-acknowledgement (SACK) TCP congestion control mechanisms [23]. As stated in [21], CCID 2 is suitable for senders who can adapt to the abrupt changes in congestion window typical of AIMD (Additive Increase Multiplicative Decrease) congestion control used in TCP. It is particularly useful for senders who would like to take advantage of the available bandwidth in an environment with rapidly changing conditions.

TCP-like congestion control is intended to provide congestion control for applications that do not require fully reliable data transmission, or that desire to implement reliability on top of DCCP. It is appropriate for flows that would like to receive as much bandwidth as possible over the long term, consistent with the use of end-to-end congestion control, and that are willing to undergo halving of the congestion window in response to a congestion event. CCID 2 is suitable for streaming media applications that buffer a considerable amount of data at the application receiver before playback time, insulating the application somewhat from abrupt changes in the sending rate.

4. Selective Frame Discard based on DCCP

In most cases, when a video server sends data it has no awareness of the available network bandwidth. This causes that in congestion periods some of the data will be discarded by the network. But even if the data is not discarded inside the network it can still be discarded at the receiver when they arrive after a certain amount of time, which is crucial for playback. In this situation, having a protocol at the transport layer that reacts to the congestion periods (e.g. DCCP) plays an important role in avoiding the packet losses due to congestion inside the network.

However, even using DCCP the data can still be discarded at the sender queue. So it can be useful, in conjunction with DCCP, to have a *selective frame discard* algorithm in order to pre-emptively discard frames at the sender that would probably be discarded at some point inside the network or at the receiver. The protocol stack of this solution is depicted in Figure 1.

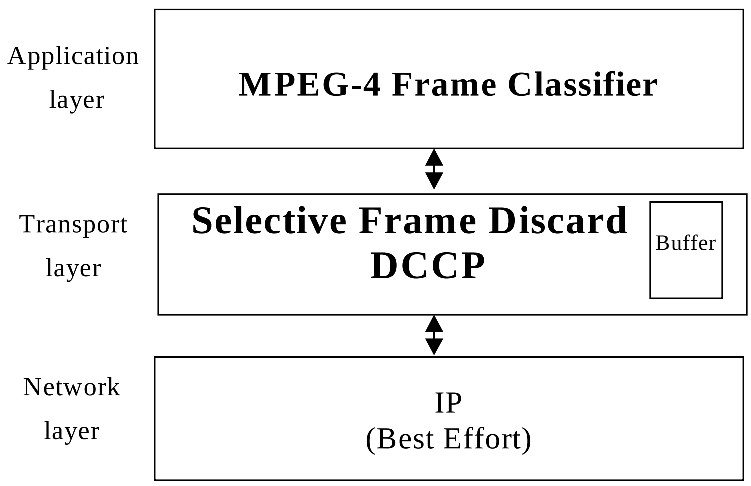


Figure 1: Protocol stack of selective drop at DCCP

Before describing the *selective frame discard* algorithms it is necessary to understand how the MPEG-4 [11] video is structured.

An MPEG-4 elementary stream usually contains one or more Groups of Video Object Planes (GOVs), which in turn are composed of several I, P or B VOP (Video Object Plane) sequences.

Although the MPEG-4 standard considers these three types of VOPs, currently almost all live encoders use only I and P VOPs, with large sequences of P VOPs between I VOPs, as shown in Figure 2.

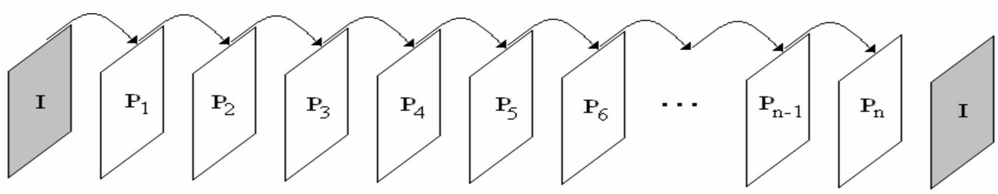


Figure 2: Typical GOV structure of a live encoded stream.

This means that in order to properly decode a P VOP, the decoder needs to completely receive the previous I VOP and all the previous P VOPs of a GOV.

When delivering this highly dependant structure through IP networks, congestion loss can seriously degrade the received quality. Therefore, a solution must be found that selectively discards frames, taking into account this VOP sequence.

This means that when congestion occurs, frame discarding should be performed on the last **m** P VOPs at a GOV, where **m** depends on the level of congestion (see Figure 3).

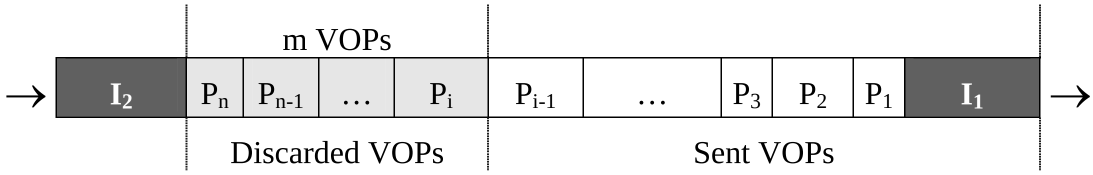


Figure 3: Sender queue showing the VOPs to be Discarded and Sent in a GOV

Note that for simplicity and because the video sequences used in this paper only have one video object layer (VOL), a VOP can also be called a frame. Either VOP or frame are used along this paper whenever considered most appropriated.

In this paper two types of *selective frame discard* algorithms are studied: *selective block discard* and *selective VOP discard*. Each one of these types can have one of two kinds of pre-determined thresholds that are used to decide when frame discarding is performed: *queue occupation level threshold* and *queue frame delay threshold*.

The first, *selective block discard* consists on discarding all of P frames inside the sender application queue, which are between two consecutive I frames starting at the head of the buffer (see Figure 4).

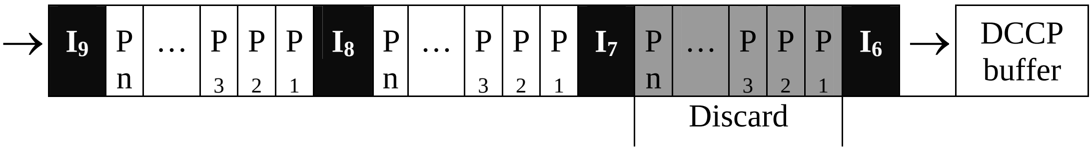


Figure 4: Sender queue showing the VOPs to be discarded using Selective Block Discard

If the *selective block discard* is used with *queue occupation level threshold*, the discarding is activated when the occupation of the sender queue is greater than the threshold. In alternative if *queue frame delay threshold* is used, the algorithm discards frames as stated previously, when any one of the packets inside the queue has exceeded the delay threshold. In Figure 5 the pseudo-code for both kinds of *selective block discard* algorithms is presented.

```
Compute Queue Occupation Level of Sender Application Queue
If Queue Occupation Level > Occupation Level Threshold
        Starting at queue head
                Search first P frame
        While (not I frame)
                Discard frame
                Next frame
        end
end
```

(a) queue occupation level threshold

```
Compute the Queue Frame Delay of frame at the Sender Application Queue head
If Queue Frame Delay > Frame Delay Threshold
        Starting at queue head
                Search first P frame
        While (not I frame)
                Discard frame
                Next frame
        end
end
```

(b) queue frame delay threshold

Figure 5: Simplified selective block discard algorithms

The second type of *selective frame discard* algorithm, called *selective VOP discard*, is similar to the *selective block discard* but less aggressive in terms of discard policy. In this case only the number of frames that do not fulfil the pre-defined threshold are discarded (see Figure 6).

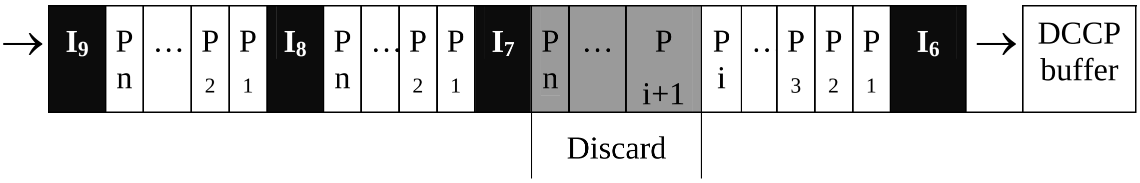


Figure 6: Sender queue showing the VOPs to be discarded using Selective VOP Discard

If the *queue occupation level threshold* is used along with *selective VOP discard* only the number of frames that have exceeded the *queue occupation level threshold* are discarded. When using *queue frame delay threshold*, the frames that have been in the sender queue longer than the *queue frame delay threshold* are discarded. Figure 7 presents the pseudo-code for the two kinds of *selective VOP discard* algorithms.

```
Compute Queue Occupation Level of Sender Application Queue
If Queue Occupation Level > Occupation Level Threshold
    Number of Frames to Discard = Queue Occupation Level - Occupation Level Threshold
    Starting at queue head
        Search first P frame
        While (not I frame)
            Next frame
        end
        Previous frame
    For (Number of Frames to Discard)
        Discard P frame
        Previous frame
    end
end
```

(a) queue occupation level threshold

```
Compute the Queue Frame Delay of frame at the Sender Application Queue head
If Queue Frame Delay > Frame Delay Threshold
    Number of Frames to Discard = Queue Occupation Level - Occupation Level Threshold
    Starting at queue head
        Search first P frame
        While (not I frame)
            Next frame
        end
        Previous frame
    For (Number of Frames to Discard)
        Discard P frame
        Previous frame
    end
end
```

(b) queue frame delay threshold

Figure 7: Simplified selective VOP discard algorithm

In conclusion, the *selective block discard* algorithms are more severe regarding the discard policy, than *selective VOP discard* algorithms. The first one can for example discard an entire GOV in the presence of congestion when only the discard of fewer frames could be sufficient to meet the threshold imposed. In contrast, the *selective VOP discard* only discards the exact amount of frames necessary to meet the threshold requirement.

5. Simulation Results

In this chapter a comparison is made between the previously presented: *selective block discard, selective VOP discard* both using DCCP, and DCCP without any kind of *selective frame discard*. The results were obtained using a simulation framework developed in MATLAB [24], which

allowed the simulation of video streaming using a DCCP CCID2 sender and receiver modules. The simulation framework is represented in Figure 8.
The MPEG-4/RTP Generator block reads the VOPs from a real MPEG-4 video file and segments them in packets with the maximum size of 1500 bytes. The DCCP queue has a First In First Out (FIFO) policy, storing the packets until the DCCP CCID2 Sender extracts them, or are discarded based on an algorithm applied by the Selective Frame Discard block.

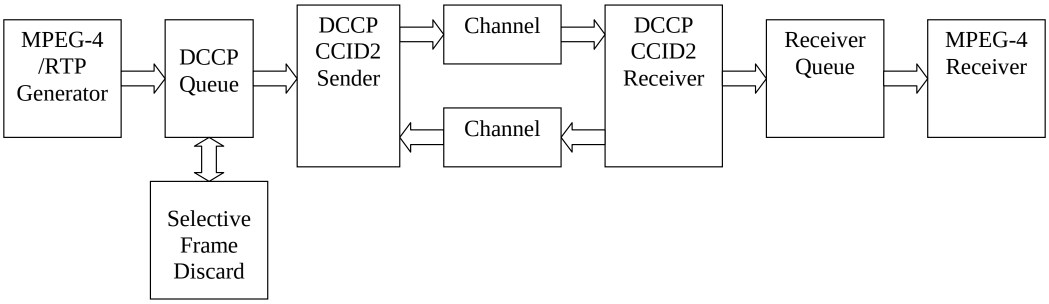


Figure 8: Simulation model block diagram

The packets that reach the DCCP Sender are then transmitted over the Channel to the DCCP CCID2 Receiver. The Channel block has the possibility of altering the available bit rate, which allows the simulation of different levels of congestion. After the DCCP Receiver acknowledges the valid packets through the reverse Channel, they are inserted into the Receiver Queue and then finally passed on to the MPEG-4 Receiver. This block saves the data related to the received packets and stores them on a MPEG-4 file format. This information will then be used for computing the PSNR and average VOP delay values.
All algorithms were tested with two MPEG-4 video sequences: one with low movement content (Akiyo sequence) and the other with high movement content (Stefan sequence). These sequences were coded with an average target rate of 256Kbps and with 25 frames per second. Only I and P frames were used and one I frame was inserted every 1 second with 24 P frames between two consecutive I frames.
The prime objective was to compare the video quality and average VOP delay under different Congestion Rates to evaluate the algorithms. The Peak Signal-to-Noise Ratio (PSNR) was used as the video quality metric, which can be obtained in the following manner:

$$PSNR = 20 \cdot \log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \tag{5.1}$$

where MSE is calculated as follows:

$$MSE = \frac{\sum\left[f(i,j) - F(i,j)\right]^2}{N^2} \tag{5.2}$$

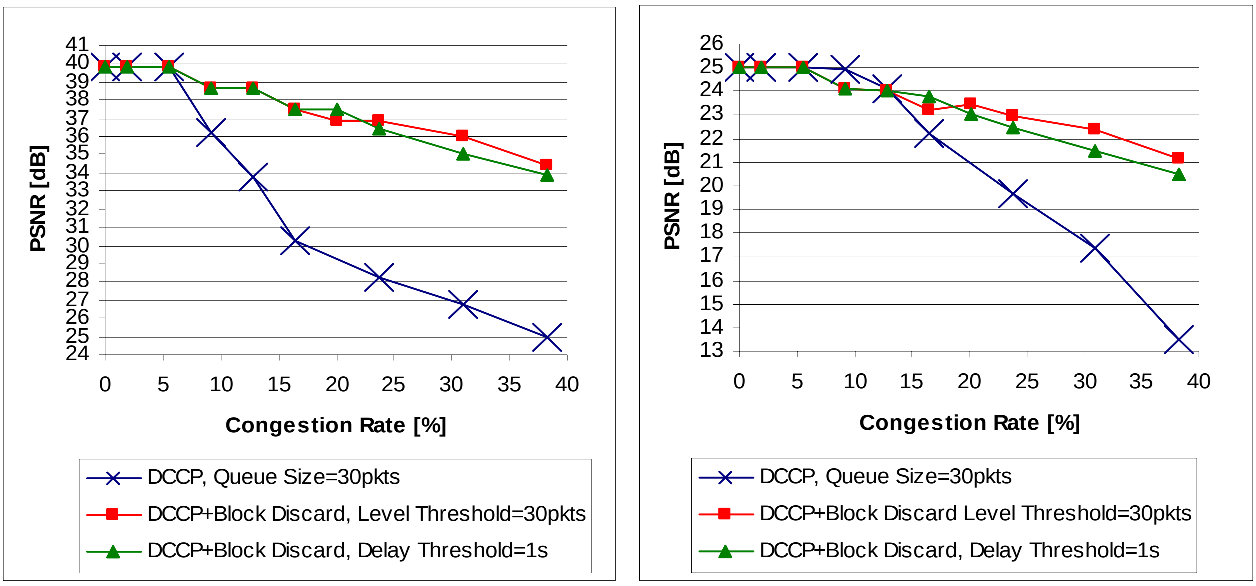
where $f(i,j)$ is the source image, $F(i,j)$ is the reconstructed image and N^2 is the number of pixels of the source image.
The average VOP delay was computed subtracting the reception time of the frame by its generation time .
The Congestion Rate in percentage was calculated in the following way:

$$CongestionRate[\%] = \frac{VideoAverageDataRate - ChannelDataRate}{VideoAverageDataRate} \times 100 \tag{5.3}$$

where ChannelDataRate is the data rate available in the channel and VideoAverageDataRate is the average bit rate of the video after coding it in MPEG-4 format. Note that the different values of congestion rate are obtained by modifying the ChannelDataRate value.
In the following results the queue thresholds used by the algorithms were 30 packets for the DCCP and for DCCP using *selective block* or *VOP discard* with *queue occupation level threshold*. For the algorithms using *queue frame delay threshold*, the threshold of one second was chosen in order to be more consistent with the results obtained by the other algorithms.

5.1. Impact of Congestion on PSNR

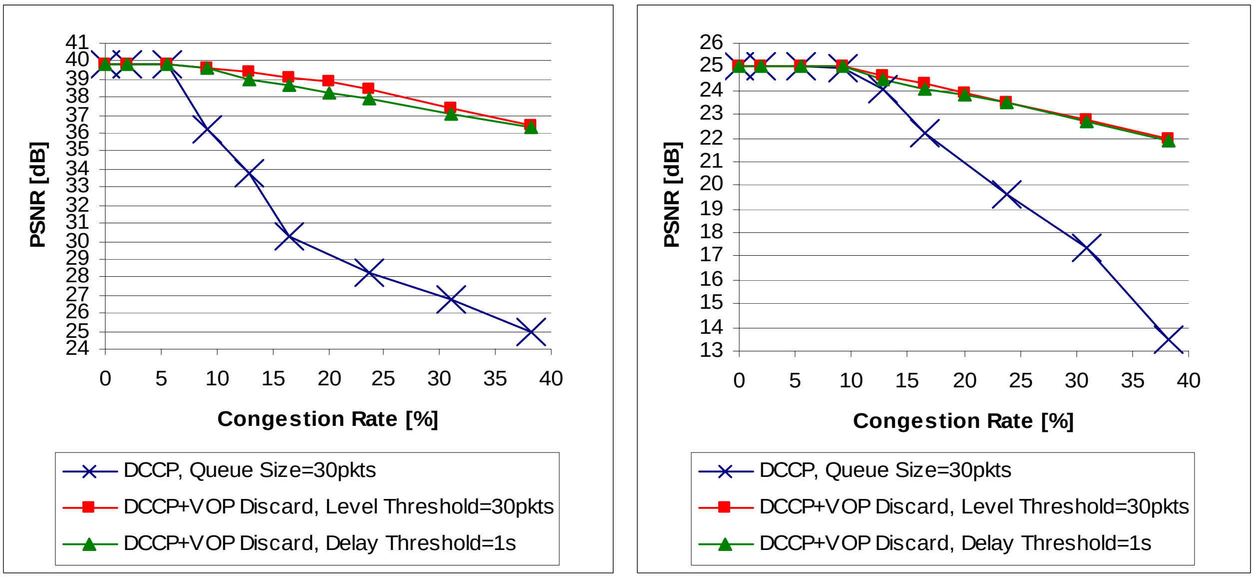
To objectively compare the impact on video quality of the *selective block* and *VOP discard* algorithms in the presence of congestion, several simulations were conducted. For each simulation a different value of Congestion Rate was imposed and the respective average PSNR of the video sequence calculated.
In Figure 9 (a) and (b) can be seen that the *selective block discard* algorithms achieve significantly higher PSNR values. The difference on PSNR between the use of *selective block discard* and no discard at all is even higher when the Congestion Rate increases.
Another important conclusion is that the effectiveness of the algorithm is apparently independent of the video sequence content, since the discrepancy of PSNR values is similar for both Akiyo and Stefan sequences.



(a) Akiyo **(b) Stefan**

Figure 9: Comparison of PSNR versus Congestion Rate of Akiyo and Stefan video sequences for Selective Block Discard Algorithms

Regarding the *selective VOP discard*, the same conclusions made for *selective block discard* apply, as shown in Figure 10 (a) and (b). However comparing the two types of algorithms, *block* and *VOP discard*, the later achieves higher PSNR (maximum of 1dB better), which can be explained with the lesser aggressiveness of the *selective VOP discard* in discarding frames.

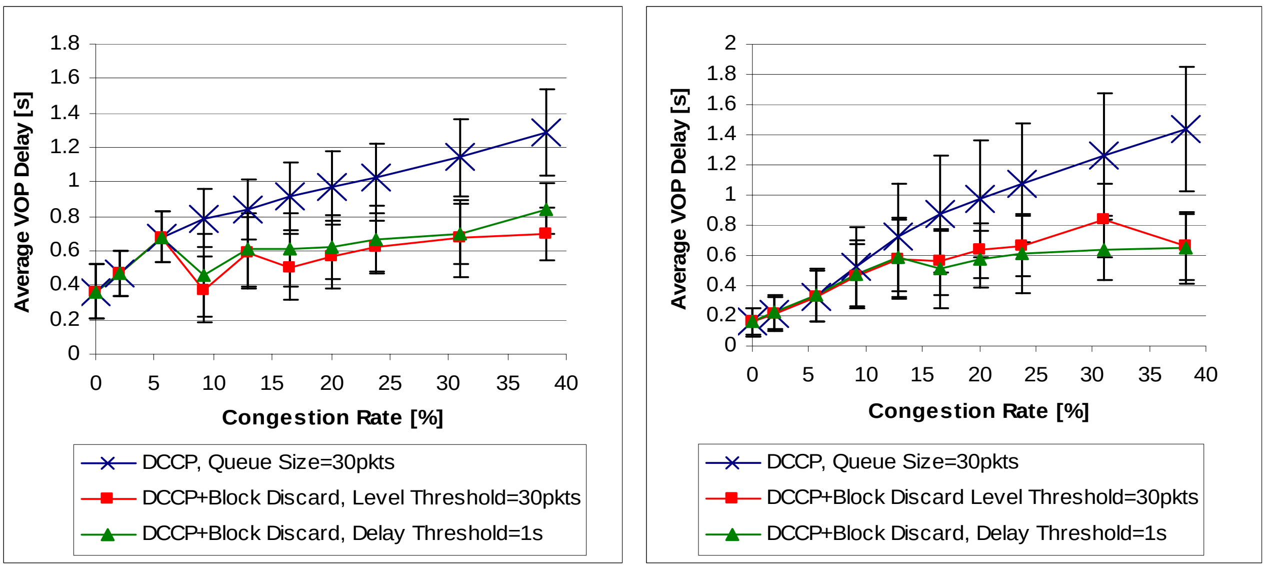


(a) Akiyo **(b) Stefan**

Figure 10: Comparison of PSNR versus Congestion Rate of Akiyo and Stefan video sequences for Selective VOP Discard Algorithms

5.2. Impact of Congestion on Average VOP Delay

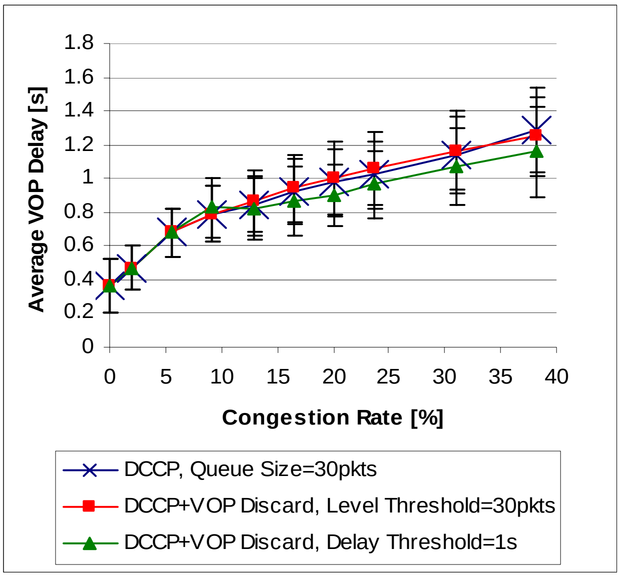
In Figure 11 the Average and Standard Deviation of VOP Delay are shown for the two movie sequences referred previously, again using the two kinds of *selective block discard*: *queue occupation level threshold* and *queue frame delay threshold.* As before, the *selective block discard* algorithms achieve better results. The average VOP delay obtained for the *block discard* is much lower than with no selective discard at all, and is similar for the two video sequences, indicating that this behaviour is likely to be independent of the movie content.



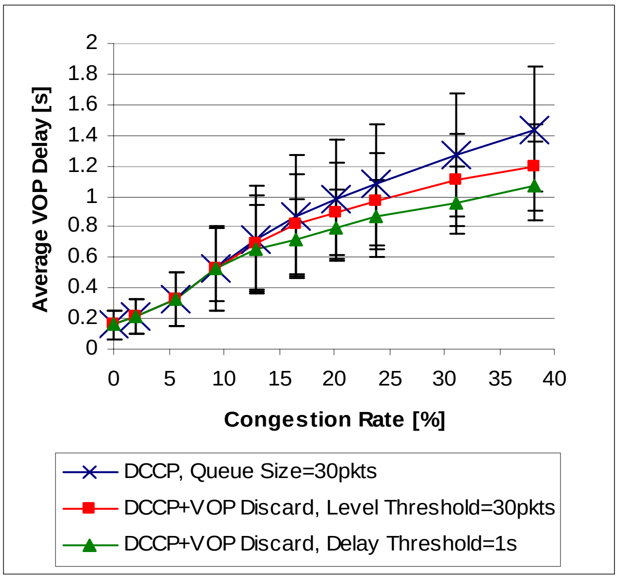
(a) Akiyo **(b) Stefan**

Figure 11: Comparison of Average VOP Delay versus Congestion Rate of Akiyo and Stefan video sequences for Selective Block Discard Algorithms

For the *selective VOP discard* the differences in average VOP delay when compared to not having selective discard are practically inexistent or slightly favouring the *selective VOP discard* algorithms as is depicted in Figure 12. This can be explained looking at the occupation of the sender queue. In the case of *selective VOP discard* only the necessary amount of frames will be discarded so that the threshold is respected. Therefore, the average occupation of the DCCP Queue will be similar to the occupation of the FIFO queue used when no selective discard is applied. Consequently, the Average and Standard VOP Delay will be similar for both *selective VOP discard* based DCCP and just DCCP.



(a) Akiyo



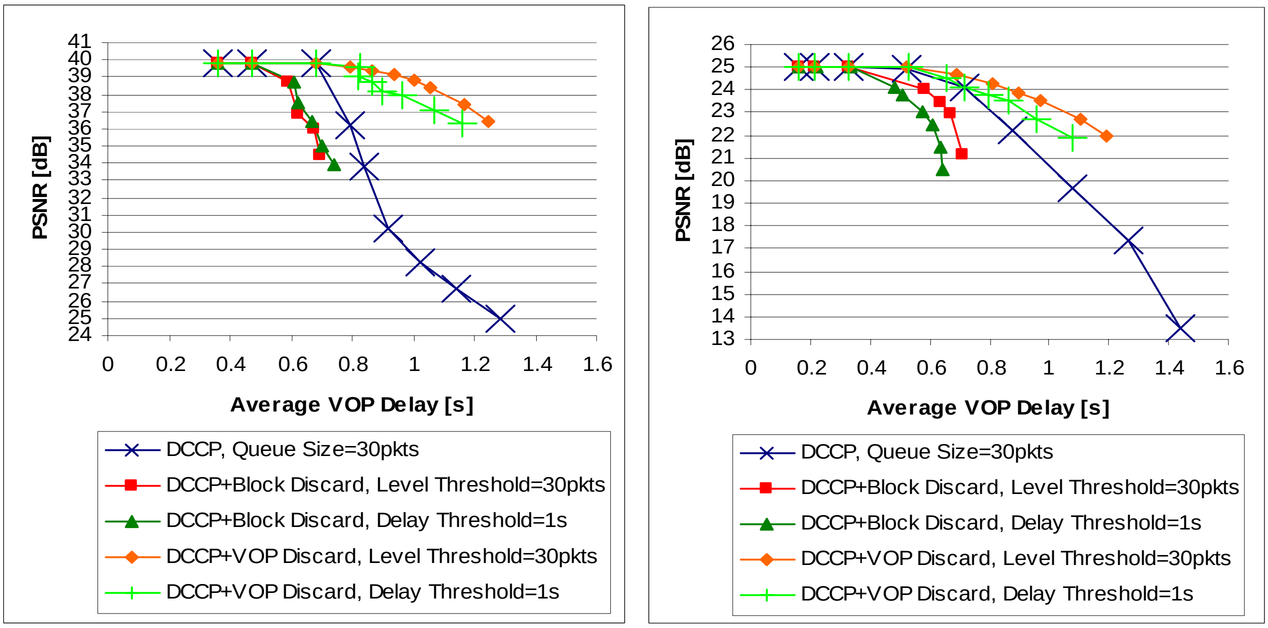
(b) Stefan

Figure 12: Comparison of Average VOP Delay versus Congestion Rate of Akiyo and Stefan video sequences for Selective VOP Discard Algorithms

Comparing Figure 11 with Figure 12 allows concluding that the *selective block discard* algorithms achieve lower values of average VOP delay than *selective VOP discard*. However, for PSNR values the *selective VOP discard* has a slight advantage, as can be seen when comparing Figure 9 with Figure 10.

5.3. Relation between PSNR and Average VOP Delay

In order to observe more easily the impact of the different *selective frame discard* algorithms presented in this paper, the PSNR and average VOP delay values were grouped together and displayed in Figure 13. The results show that in terms of PSNR the *selective VOP discard* achieves generally higher values than any of the other algorithms studied. However, in terms of average VOP delay the lowest values are obtained by the *selective block discard* algorithms.



(a) Akiyo **(b) Stefan**

Figure 13: Comparison of PSNR versus Average VOP Delay of Akiyo and Stefan video sequences for Selective Block and VOP Discard Algorithms

The independence of the *selective frame discard* algorithms in relation to the movement content that characterizes the sequences can also be inferred from Figure 13. For both sequences, Akiyo and Stefan that have completely different movement content the results are quite similar. This proves the importance of using either *selective block* or *VOP discard* algorithms to achieve better performances in PSNR and end-to-end delay when video is transported across an IP network.

6. Conclusions and Future Developments

In this paper two new *selective frame discard* algorithms were presented and studied in terms of PSNR and average VOP delay. The results obtained show that in the presence of congestion, the *selective block* and *VOP discard* algorithms achieve higher PSNR and the same or lower average VOP delay when compared to not using any *selective frame discard* algorithm.

Other important conclusions are that the *selective VOP discard* algorithms achieve the highest PSNR values and that the *selective block discard* algorithms obtain the lowest values of Average VOP Delay. Therefore, an interesting future development based on the work presented in this paper would be the design of a new type of *selective frame discard* algorithm that could attain the better of these two selective discard approaches: highest PSNR values and lowest average VOP delay.

It is also clear in this paper that for the two different types of movement content used: Akiyo (low movement) and Stefan (high movement), the above conclusions are still valid. In other words, the improvement in PSNR and average VOP delay provided by the *selective block* and *VOP discard* algorithms is independent from the movement content in each of the video sequences.

Finally, the DCCP protocol, which is under consideration by the IETF still raises some important questions, e.g.: its coexistence with TCP that could be explored in future work.

7. References

[1] D. Wu, Y. Hou, Y. Zhang, "Transporting Real-Time Video over the Internet: Challenges and Approaches", Proceedings of the IEEE, Vol. 88, No. 12, December 2000, pp.1855-1874.

[2] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: An

overview", RFC 1633, Internet Engineering Task Force, July 1994.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services", RFC 2475, Internet Engineering Task Force, December 1998.

[4] Eddie Kohler, *et al*, "Datagram Congestion Control Protocol (DCCP)", Internet draft "draft-ietf-dccp-spec-07.txt", 18 July 2004.

[5] IETF home page: http://www.ietf.org

[6] H. Schulzrinne, *et al*, "RTP: A Transport Protocol for Real-Time Applications", RFC1889, Internet Engineering Task Force, January 1996.

[7] P. Postel, "User Datagram Protocol", RFC 768, Internet Engineering Task Force, August 1980.

[8] R. Fielding, J. Gettys, J. Mogul, H. Nielsen, T. Berners-Lee, "Hypertext transfer protocol - HTTP/1.1", RFC2068, Internet Engineering Task Force, January 1997.

[9] P. Postel, "Transmission Control Protocol", RFC 793, Internet Engineering Task Force, September 1981.

[10] M. Allman, *et al*, "TCP Congestion Control", RFC 2581, Internet Engineering Task Force, April 1999.

[11] ISO/IEC International Standard 14496; "Information Technology – Coding of Audio-Visual Objects", 2001.

[12] OLYMPIC Project home page: http://olympic.sema.es

[13] Y. Bernet *et al*, "A framework for Integrated Services Operation over DiffServ Networks," RFC2998, Internet Engineering Task Force, November 2000.

[14] K. Nichols et al, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC2474, Internet Engineering Task Force, December 1998.

[15] V. Jacobson et al, "An Expedited Forwarding PHB Group," RFC2598, Internet Engineering Task Force, June 1999.

[16] Heinanen et al, "Assured Forwarding PHB Group," RFC2597, Internet Engineering Task Force, June 1999.

[17] J-C. Bolot, T. Turletti, "Experience with control mechanisms for packet video in the Internet", *ACM Computer Commun. Rev.*, vol. 28, Jan. 1998.

[18] D. Wu, Y. T. Hou, W. Zhu, H.-J. Lee, T. Chiang, Y.-Q. Zhang, and H. J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 923–941, Sept. 2000.

[19] A. Eleftheriadis and D. Anastassiou, "Meeting arbitrary QoS constraints using dynamic rate shaping of coded digital video," in *Proc. IEEE Int.Workshop Network and Operating System Support for Digital Audio and Video (NOSSDAV'95)*, Apr. 1995, pp. 95–106.

[20] E. Gurses, G. Bozdagi Akar, N. Akar, " Selective Frame Discarding for Video Streaming in TCP/IP Networks," Packet Video Workshop April 2003, Nantes, France.

[21] Sally Floyd, *et al*, "Profile for DCCP Congestion Control ID 2: TCP-like Congestion Control", Internet draft "draft-ietf-dccp-ccid2-06.txt", 18 July 2004.

[22] Sally Floyd, *et al*, "Profile for DCCP Congestion Control ID 3: TFRC Congestion Control", Internet draft "draft-ietf-dccp-ccid3-06.txt", 18 July 2004.

[23] E. Blanton, *et al*, "A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP", RFC 3517, Internet Engineering Task Force, April 2003.

[24] MATLAB home page: http://www.mathworks.com