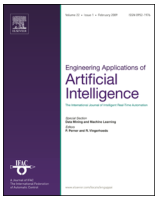
Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



The application of neural networks to improve the quality of experience of video transmission over IP networks

Carlos Eduardo Maffini Santos [a,b,1], Eduardo Parente Ribeiro [c,2], Carlos Marcelo Pedroso [c,*]

[a] Federal Institute of Parana, Technical Education Center, Curitiba, Brazil
[b] Pontificia Universidade Catolica do Parana, Campus Curitiba, TECPUC, Rua Imaculada Conceicao, 1155, Prado Velho, 80215-901 Curitiba, PR, Brazil
[c] Federal University of Parana, Department of Electrical Engineering, Curitiba, Brazil

ARTICLE INFO

Article history:
Received 8 March 2013
Received in revised form
16 August 2013
Accepted 28 August 2013
Available online 5 October 2013

Keywords:
Neural networks
Multimedia systems
Quality of experience
Selective packet discard
Traffic classification

ABSTRACT

The transmission of real-time multimedia streams requires service guarantees, such as limited packet loss, minimum bandwidth and low delay and jitter, to ensure a good quality of experience (QoE) for viewers. The spatial and temporal redundancy of videos is addressed by coding algorithms that reduce the amount of information necessary to represent the images. As a consequence, multimedia traffic commonly presents variable bit rate behavior and self-similar characteristics. Although the reduction in bandwidth requirements is highly desirable, the burstiness of traffic leads to problems in network design and performance prediction. Even a low level of packet loss could severely affect the viewer QoE. In this paper, we propose a real-time packet payload classifier, implemented with artificial neural network (ANN) to be used at network routers. A priority packet discard strategy can be implemented to avoid discarding packets that carry the most relevant information for image reconstruction, thus improving the perceived quality. This approach does not require changes at the video source to classify outgoing packets. The ANN was employed because of its good capacity in temporal series recognition and the possibility of its implementation in real-time systems due to its low computational complexity. The video traces used for training and validation were encoded with H.264/MPEG-4 Advanced Video Coding and are publicly available. The priority packet discard strategy was tested through computational simulations. The QoE was estimated comparing the peak signal-to-noise ratio (PSNR) of original and the received frames of video, and the results indicate that the proposed method improves the QoE. The implementation does not require packet payload processing and can be performed with network layer information only.

1. Introduction

The network traffic of video streaming can be self-similar (Dai et al., 2009) due to video encoding algorithms (codec), where the frame sizes vary according to the information that the frames carry. In a network architecture based on packet switching, the burstiness of network traffic can cause congestion in the router queues, leading to a possible loss of packets. The packet discard could occur even at low average utilization levels, as a consequence of traffic burstiness, leading to a temporary decrease in the quality of experience (QoE). The delivery of high-quality video is a complex issue, particularly for networks based on packet-switching techniques. Even the slightest packet loss in a video stream may result in a severe degradation of quality (Szymanski and Gilbert, 2009), and 1% or less of packet loss could severely affect the quality of image, reducing the QoE (Greengrass et al., 2009).

Numerous algorithms for video encoding have been developed. Among them, MPEG-2 and MPEG-4 are currently the most used standards. MPEG-4 is a family of open international standards that provides tools for use in multimedia applications (Van der Auwera et al., 2008a). The tools include codecs for encoding audio and video. MPEG-4 has the advantage of requiring lower transmission rates compared with its predecessors, MPEG-1 and MPEG-2. Thus, MPEG-4 allows an improvement in terms of bandwidth utilization, as well as a decrease in the amount of space for video storage (Marpe et al., 2006). The MPEG algorithm addresses the temporal redundancy of videos by representing the sequence of images with a group of pictures (GOP) that consists of a specific sequence of frames. The GOP starts with an intra frame (I-frame), which can be decoded without other frames, followed by bidirectional frames (B-frames) and predictive frames (P-frames). The P-frames depend

[*] Corresponding author. Tel.: +55 41 96653582, +55 41 3361 3256.
E-mail addresses: carlos.maffini@gmail.com (C.E. Maffini Santos), edu@eletrica.ufpr.br (E. Parente Ribeiro), pedroso@eletrica.ufpr.br, pedroso.carlos@gmail.com (C.M. Pedroso).
[1] Tel.: +55 41 3271151.
[2] Tel.: +55 41 33613227.

0952-1976/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2013.08.014

on information from the nearest previous I- or P-frames and the B-frames use both past and future I- or P-frames as references for image representation.

The H.264/MPEG-4 part 10 Advanced Video Coding (AVC) standard presents improvements in compression efficiency and is widely used in multimedia application standards and industry consortia specifications (Seeling and Reisslein, 2012; Maisonneuve et al., 2009) and for this reason it is the codec we choose to use. Hereinafter, for brevity, the H.264/MPEG-4 Part 10 AVC standard shall be referred to as “H.264”. However, although the encoding used was the H.264, we expect that the proposed method can also be applied to MPEG-4 or MPEG-2.

Fig. 1 shows the transmission sequence of I-, P- and B-frames for a video encoded with H.264. The GOP always starts with an I-frame, followed by B- and then P-frames. The sequence of frames depends on the encoding settings. The common notation uses the pair (*Y*,*Z*), where *Y* indicates the number of frames in the GOP and *Z* represents the number of B-frames between the P-frames. Fig. 1 illustrates the frame sequence for the (12,2) configuration. As the I-frames include all the information needed to decode the image without information from other frames, they are usually of a larger size than the others. Therefore, more IP packets are needed to transport an I-frame than to carry other frames, as shown in Fig. 2.

The effects of packet loss in viewer QoE are analyzed in Greengrass et al. (2009). Discarded packets carrying I-frames could result in image impairments propagated to all frames in that GOP. This could last a long time (typically from 0.5 to 1 s); video quality is recovered only when the decoder receives an unimpaired I-frame. This kind of distortion happens because the H.264 decoder uses the I-frame as reference to decode the other frames in the GOP. Depending on which packet is lost, the distortions may result in several degrees of severity, e.g. the loss of a single IP packet at the beginning of an I-frame, which contains the frame header, might have the same effect as losing a whole I-frame. Greengrass et al. (2009) also indicate that the higher the number of frames in a GOP, the greater the impairments caused by a packet loss.

To improve the viewer's QoE, Hong and Won (2010) proposed the implementation of a packet scheduler algorithm, adjusting the time intervals between packets based on their significance. The significance is defined as the importance of a packet to the image reconstruction and is obtained through analyses of the consequence of loss for each pixel transported by the packet, considering the GOP structure. This concept was applied to implement a packet scheduler called the Significance-Aware Packet Scheduler (SAPS). With SAPS the packets with higher significance will take a longer inter packet time interval than the less significant packets. From the network perspective, when the technique is applied, the resulting traffic has its burstiness modified. This allows routers to free up some space on their buffers before the next packet arrival. The most significant packets wait a longer time to be transmitted and are likely to be preserved in case of network congestion. As a result, according to the authors, the QoE perceived by the viewers is improved. SAPS can also process the Explicit Congestion Notification (Ramakrishnan et al., 2001) to collaborate with network congestion and discard the less significant packets to reduce the impairments to the QoE. The entire implementation of SAPS is done at the streaming server. The evaluation of significance requires payload processing, with high computational complexity, making it prohibitive to implement in routers.

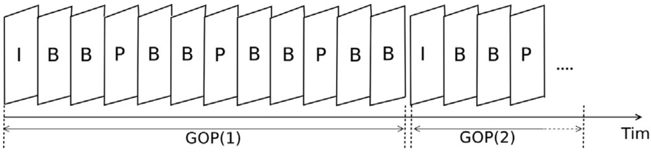


Fig. 1. Sequence of I-, P- and B-frames encoded with H.264.

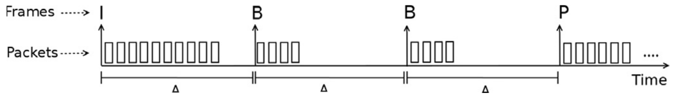


Fig. 2. Encoded packets carrying MPEG frames.

An algorithm that combines packet scheduling and queue management is proposed in Huang et al. (2006). The algorithm improves the transmission of video streams over networks with bandwidth constraints. Called Active Drop Queue (ADQ), the algorithm implements three distinct queues: one for the traffic within the bandwidth limit (conformant queue), one for the traffic exceeding the bandwidth limit (excess queue), and the last one for the best-effort traffic. Each packet in queue is associated with a time stamp. This allows the evaluation of an excess delay beyond a specified deadline where the transmission of packet is useless. In case of network congestion, the ADQ removes, from excess queue, the packets with its deadline expired, freeing the queue to receive new packets.

An active queue management algorithm based on priority dropping (PD) and a proportional-integral-derivative (PID) controller is proposed in Xiaogang et al. (2007) using the control theory, called PID_PD, which first drops the least important packets when network congestion arises. The packet is marked by the application layer, writing the priority number to the priority field of the IP packet. The B-frames receive the least priority and I-frames receive the higher priority. The results show that the schema can prevent the high-priority layer or frame from dropping, thus preserving viewer QoE in case of network congestion.

Another strategy to preserve viewer QoE in situations of network congestion is proposed in Schier and Welzl (2009). The packet classification is based on macroblock distortion estimation, requires superficial decoding of the video bitstream, and takes in account important indicators such as the macroblock composition of frames, temporal dependencies and potential scene cuts. The results show an increase of perceived video quality by an average of 3–4 dB in terms of peak signal-to-noise ratio (PSNR).

A survey on methods for Internet traffic identification, using information available in the network layer to classify the payload of packets is presented in Callado et al. (2009) and Nguyen and Armitage (2008). The goal of these methods is to identify the application protocol without relying on well-known TCP or UDP port numbers or processing the packet payload. The main applications are in quality of service (i.e., different applications get different service from the network), filtering (i.e., threats/attacks can be blocked), and billing (i.e., per-application charging rates) (Callado et al., 2009). The methods available for classification are based on heuristics and use the time interval between packets, the size of packets, and the length of session as input. The effectiveness of classification varies greatly, depending on the application protocols and the real-time capacity of the method.

In this paper, we analyze the use of a selective packet discard (SPD) strategy to preserve the QoE in H.264 transmission over congested IP networks. The idea is to preserve the packets that carry more relevant information for image reconstruction in case of network router congestion. To implement the SPD, the packets of video traffic should be classified. The first and natural alternative is to perform the packet marking jointly by the video encoder and packetization components at the streaming server. This information can be stored as DiffServ Code Point (DSCP) in the IP header, eliminating the need to classify packets in the routers at real time. This strategy is possible if the network facilities and streaming server can be configured jointly to collaborate. In some cases, however, the streaming server administrator cannot control the network facilities and vice-versa. In this case, one can implement a strategy in the server or in the network routers, independently. SAPS was designed to be implemented at the server side, without the collaboration of network routers. Thus, in this paper we analyze two alternatives: (i) performing the packet classification and SPD at network routers using only information available at network layer – the goal is to investigate if this can be done without the cooperation of the servers, using artificial neural networks (ANN) due to their low computational complexity, thus allowing their implementation in routers; and (ii) a collaborative setting, with servers marking the packets and routers implementing the SPD. The latter alternative was implemented as a performance reference and will be referred to as the *golden standard*. The main benefit is the preservation of the viewer's QoE in case of network congestion.

The rest of this paper is structured as follows. Section 2 describes the ANN topologies used for the packet payload classification, the origin of the data set under study, and the classification results. Section 3 presents the proposed packet discard strategy. Section 4 shows the method evaluation for several network congestion scenarios. Section 5 presents the conclusions.

2. Packet classification

We choose to employ ANN to perform the packet payload classification, because ANNs are noted for being capable of solving complex problems of forecasting and recognition of time series and can be implemented in real-time systems because of their low computational complexity. According to Basu et al. (2010), ANNs provide a suite of nonlinear algorithms for feature extraction and classification, and can be efficiently implemented in hardware, including the implementation of sigmoid activation functions (Szabó and Horváth, 2004; Mishra et al., 2007).

Particularly in the case of video traffic encoded with H.264, with variable bit-rate characteristics due the variation of frame size, we expect that the training of the ANN is capable of capturing the characteristics of packet flow and relating these with the type of frame they carry.

As a basic premise, the video streams should be pre-classified in separate queues, as illustrated in Fig. 3. This pre-classification could be implemented based on IP address and port numbers. The effects of misclassification at this stage are not analyzed in this paper. However, a misclassification at this stage would negatively affect the proposed method. We consider the development of methods to prevent this problem a topic of future work, because we are interested in finding out whether it is possible to implement a packet classifier with ANN associated with an SPD mechanism exclusively on routers.

Several approaches have been proposed to model MPEG traffic (Dai et al., 2009; Van der Auwera et al., 2008a, 2008b; Klein Junior and Pedroso, 2013). The high variability of the video traffic, the short- and long-range correlations, and the sudden scene changes make it difficult to perform the payload classification with traditional methods. In this scenario, the use of ANN may be appropriate, because the training process can capture the characteristics of the system without the use of a particular traffic model.

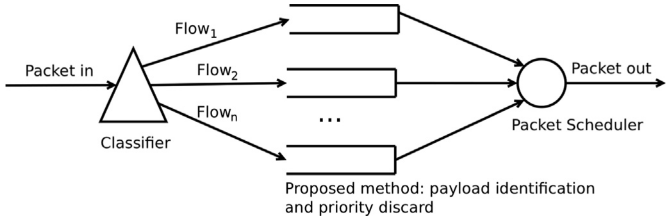


Fig. 3. Router configuration: each video stream should be pre-classified in an independent queue.

A common way to characterize the accuracy of a classifier is through metrics known as false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN) (Nguyen and Armitage, 2008). TP is defined as the percentage of members of class X correctly classified as belonging to class X and TN is the percentage of members of other classes correctly classified as not belonging to class X. FN and FP are given respectively by 1-TP and 1-TN.

Two approaches to define the class X were considered: (a) X= packets carrying I-frames, avoiding to discard them and (b) X=packets carrying B-frames, which would be the first to be discarded in case of network congestion. With approach (b), the false negatives could be packets carrying I-frames or P-frames, and they would be prioritized to be discarded. With approach (a), the false negatives could be the packets carrying the P-frames or B-frames, and they would be preserved from discarding. As our goal is to preserve I-frames because their importance to image decoding, we choose approach (a). Thus, X represents the packets carrying I-frames.

The input parameters used in the ANNs were the time interval between packets, $\delta_k \in \mathbb{R}$, and the packet sizes, $\rho_k \in \mathbb{N}^*$, observed within a past time window with N observations. The index $k \in \mathbb{N}^*$, represents a particular observation. Therefore, the input consists of $\delta_k, \delta_{k-1}, \ldots, \delta_{k-N}, \rho_k, \rho_{k-1}, \ldots, \rho_{k-N}$. Thus, the number of inputs is $2N$ considering the two input variables and the window size.

The output of the ANN is given solely by $x \in \mathbb{R}$, $0 \le x \le 1$, where $x=1$ and $x=0$ represent respectively the presence and the non-presence of a packet carrying an I-frame in the input window. As x is a real number, it will be used as a confidence level of the ANN output.

The chosen approach was not designed to classify packets individually, and one can observe that, with our proposal, achieving a true positive classification of 100% would be almost impossible due to the uncertainty about the type of all packets in the window. In a first approach, we attempted to identify the packets individually. We realized, however, that it would be very difficult to achieve a good success rate this way. Nonetheless, considering the application, the queue sizes, and the maximum packet loss to the video decoding still being possible, we realized that a false positive identification was acceptable, as long as a sufficient number of packets transporting B- and P-frames were marked as priority candidates to discard. Thus, the ANN topology was designed to improve the chances of success in the identification of packets carrying I-frames, without much preoccupation with false negatives, since this approach improves the percentage of true positives.

2.1. On the use of ANN to packet classification

A number of ANN topologies are commonly applied for the prediction and identification of time series: feed-forward (FF), cascade-forward (CF), feed-forward with tapped delay (FFTD), radial basis (RB), general regression (GR) and Elman recurrent with tapped delay (ERTD) (Principe et al., 1999). We choose to employ two neural networks topologies: (i) FFTD and (ii) ERTD, mainly due to the simplicity of FFTD and the good results reported for ERTD in time series recognition (Abdennour, 2006).

Fig. 4(a) and (b) shows respectively the structure of the neural networks FFTD and ERTD. Both architectures have $2N$ inputs, one hidden layer and one output layer with one neuron whose output reports if the packets within window N carry I-frames or not. Additionally, ERTD has a context layer with the same number of neurons of the hidden layer. The number of neurons in the hidden layer was established by the arithmetic average between the number of inputs and outputs, $\lfloor (2N+1)/2 \rfloor = N$.

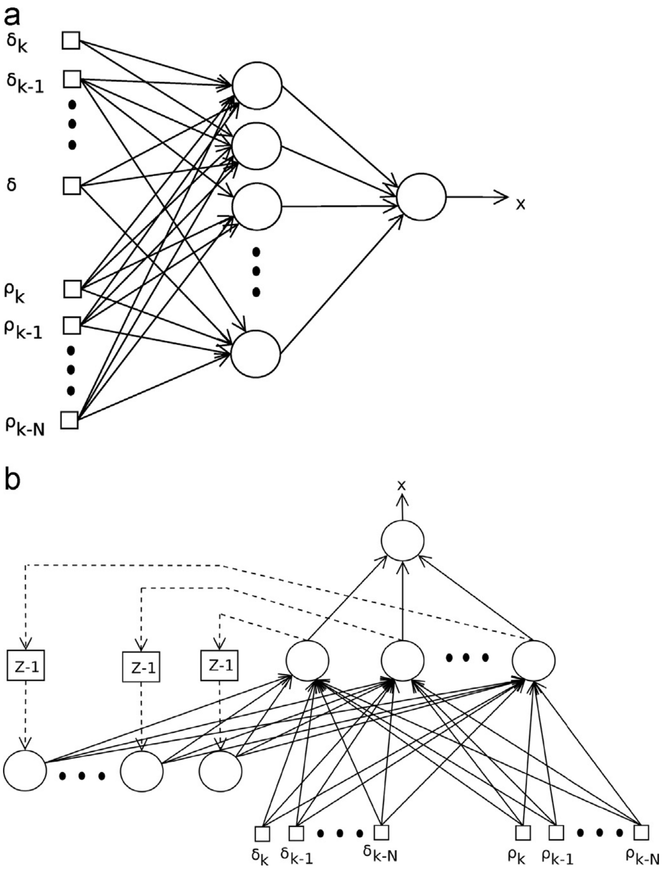


Fig. 4. Topologies of ANN: (a) feed-forward with tapped delay (FFTD) and (b) Elman recurrent with tapped delay (ERTD).

For the neural network training, the video coded data were split into two sets. The first consisting of 70% of the total, was used for training; the remaining 30% was used for the validation. According to Haykin (1998), in practice, the training can achieve a good generalization with training set T given by $T=O(W/E)$, where W is the total number of free parameters (i.e., synaptic weights) of the network, E denotes the fraction of classification errors allowed, and $O(\cdot)$ denotes the order enclosed within. Considering the worst case for neural network topology, for the ERTD network, the number of free parameters is given by $3N^2+2N$, where N represents the window size. Considering the use of $N=12$, which is the maximum GOP length for this video and the maximum value for N, the training set has 6543 situations and the fraction of errors allowed for Salesmen video can be calculated as $E=6.9\%$. The error for all other videos is lower than this, what ensure a good generalization for all training.

The window size N is fundamental to the success of the classification. If N is lower than the number of packets necessary to transport an I-frame, the neural network cannot recognize the presence of an I-frame due to the lack of input data. If N is greater than the GOP size, the window will necessarily contain an I-frame, making the planned approach worthless, because the output of the neural network would be always 1.

Thus, tests were performed using N greater than the minimal number of packets to carry an I-frame and lower than the average number of packets of the GOP. With these restrictions, we seek the smallest possible window size N.

2.2. *Data source*

The videos employed for the tests are publicly available at Video Trace Library (2012), with resolution of chrominance sub sampling of 4:4:4, resolution of 352 × 288 pixels and 25 fps. The GOP configuration was (12,2). Other authors have often used these videos in the study of image and transmission systems, as in Greengrass et al. (2009), Van der Auwera et al. (2008b), Abdennour (2006) and Bouras et al. (2009). All videos were encoded with H.264 through the use of the *ffmepg* (Niedermayer, 2012) tool, which is also publicly available. The *ffmpeg* tool allows the adjustment of several parameters, such as the GOP configuration, image size, quality/compression, and frame rate, among other settings.

Table 1 summarizes the main video characteristics, showing total and average frame sizes, and total and average packet sizes. The videos were chosen because of their characteristics, ranging from static to dynamic images, resulting in several levels of traffic burstiness. The movies *Star Wars Ep. IV* (SW), *Jurassic Park* (JP) and *Silence of the Lambs* (SL) were included to provide a range of different behaviors, caused by scene changes.

As the movies have a large length, they were separated into smaller sets, typically two minutes long, to facilitate the analysis and reduce the computational effort to evaluate the results. The subsets were named JP, JL, SW-1, SW-2, and SW-3; JP, JL, SW-2, and SW-3, represent scenes with moderate motion whereas SW-1 has sudden scene changes, which is the worst case for the packet classifier. Fig. 5 illustrates the first images for the scene sets JP, JL, SW-1, SW-2, and SW-3.

To collect δ_k and ρ_k, the videos were transmitted over a non-congested Ethernet network, and the data were captured with traffic monitoring tools *tcpdump* (Richardson and Fenner, 2012; Wireshark, 2012).

2.3. *Training and validation of classifier*

The training and the validation of ANNs were done with the Java Neural Network Simulator (javaNNS), developed by *Wilhelm-Schickard-Institute for Computer Science* (Fischer et al., 2001). The javaNNS was chosen because of its reliability, the large number of available topologies and training algorithms supported, and the ability to generate source code in C language, facilitating the later implementation of the queue simulator.

The neural networks were trained with the *Backpropagation* algorithm. During the training process using this algorithm, the network operates a sequence of two steps. The first consists in presenting a set of patterns to the network input. Data are processed and flow through the network, layer by layer, until the response is propagated to the output layer. This procedure is called the forward propagation phase. In the second step, the ANN output is compared with the training data set. If the outputs values are not equal, the error is computed and propagated from the output layer to the input layer, changing the values of the connection weights of internal layers of the network.

Table 1
Summary of basic statistics of the videos.

Video	Number of frames	Average frame size (bytes)	Number of packets	Average packet size (bytes)
Highway	2001	13,016	18,810	1416
Bridge Far	2101	12,247	18,637	1403
Coast Guard	300	20,514	4360	1448
Paris	1065	11,413	8845	1408
Soccer	300	15,575	3345	1431
Salesman	450	1863	779	1075
JP	3720	5772	16,158	1329
SL	3600	2811	8515	1189
SW-1	3719	5708	16,096	1319
SW-2	3719	4181	12,324	1262
SW-3	3599	3345	9999	1205

Fig. 5. First images for the scene sets JP, SL, SW-1, SW-2, and SW-3, respectively, from the left to the right.

Table 2
Percentage of true positives using the topology FFTD for the videos *Coast-Guard*, *Highway*, *Bridge Far*, *Paris*, and *Soccer* for the *training* sets.

N	*Highway* (%)	*Bridge* (%)	*Coast* (%)	*Paris* (%)	*Soccer* (%)	Average (%)
15	17	32	22	95	50	43.2
25	99	100	92	100	100	98.2
35	100	100	100	100	100	100
45	100	100	100	100	100	100
55	100	100	96.4	100	100	99.3

Table 3
Percentage of true positives using the topology ERTD for the videos *Coast-Guard*, *Highway*, *Bridge Far*, *Paris*, and *Soccer* for the *training* sets.

N	*Highway* (%)	*Bridge* (%)	*Coast* (%)	*Paris* (%)	*Soccer* (%)	Average (%)
15	86	81	65	98	90	83.9
25	96	100	85	99	91	94
35	98	99	85	100	96	95.7
45	99	99	93	100	100	98.2
55	98.5	100	93	100	100	98.3

Table 4
Percentage of true positives using the topology FFTD for the videos *Coast-Guard*, *Highway*, *Bridge Far*, *Paris* and *Soccer* for the *validation* sets.

N	*Bridge* (%)	*Coast* (%)	*High* (%)	*Paris* (%)	*Soccer* (%)	Average (%)
15	55	52	51	66	51	55
25	65	42	59	48	49	53
35	61	49	50	54	18	46
45	62	32	45	47	27	43
55	64	39	56	72	33	53

Table 5
Percentage of true positives using the topology ERTD for the videos *Coast-Guard*, *Highway*, *Bridge Far*, *Paris* and *Soccer* for the *validation* sets.

N	*Bridge* (%)	*Coast* (%)	*High* (%)	*Paris* (%)	*Soccer* (%)	Average (%)
15	77	73	73	74	63	72
25	72	42	65	59	54	58
35	69	62	62	59	29	56
45	62	39	51	75	41	54
55	62	96	53	70	44	65

This procedure is known as the backward propagation. The Error Backpropagation algorithm is the most famous among the learning algorithms, being particularly useful in cases of large training sets with many similar examples (Zell et al., 2011). The parameters of the training algorithm are *dmax*, i.e., the maximum difference between target value and the value obtained by the output of the neuron, and η, i.e., the learning rate. Typically, *dmax* should range from 0 to 0.2, according to the desired error, and was set at 0.01 to get a small error. The η parameter indicates the step size for the adjustment of synaptic weights between neurons connections for each training cycle. The lower the learning rate, the lower the adjustment of synaptic weights, which will provide a gradual update of the weights, but a considerably longer time of training. Thus, increasing the learning rate will result in the acceleration of the training time, but the adjustment in the weights between connections will be more significant. The learning rate parameter was set to 0.1. This was done because the time for training is not critical for the application under consideration and the training is performed offline. The amount of training cycles was set to 50,000 due to the observation of a significant reduction in error after 5000 training cycles. All neurons were configured with the sigmoid activation function, which has many interesting features, including its capacity to capture the non-linear characteristics of the process (Principe et al., 1999).

2.4. Classification results

This section shows the results of the tests employing the ANN topologies described in Section 2.1. The window size was configured with size of $N=15$, 25, 35, 45, and 55, for *Coast-Guard*, *Highway*, *Bridge Far*, *Paris*, and *Soccer*. For *SW*-1, *SW*-2, *SL*, and *JP*, the window size was configured with $N=10$, 13, 16, 21, 24, and 27. The window size N was chosen due the GOP structure of the videos, the average number of packets to carry an I-frame and the GOP, and considering the maximum transfer unit (MTU) of 1500 bytes. Suppose the average number of packets for transport I-, P- and B-frames are denoted by ϕ_I, ϕ_P, and ϕ_B, respectively. *Coast-Guard*, *Highway*, *Bridge Far*, *Paris*, and *Soccer* present $\phi_I=15$, $\phi_P=10$ and $\phi_B=10$. *SW*-1, *SW*-2, *SL*, and *JP* present $\phi_I=10$, $\phi_P=5$ and $\phi_B=3$.

Tables 2 and 3 show the percentage of true positives for the videos *Coast-Guard*, *Highway*, *Bridge Far*, *Paris* and *Soccer*, for the training sets, for the FFTD and ERTD topologies, respectively. These results show that the FFTD and ERTD topologies could be trained with a good degree of accuracy. It is possible to see the growing of true positive percentage as N increases. The poor performance of $N=15$ can be seen as well, because it is the average number of packets necessary to carry an I-frame in those videos, and the neural network does not have an enough number of parameters to identify the transition between frames. With $N=25$, the training achieves a true positive percentage average of 98.2% and 94%, for FFTD and ERTD, respectively. The $N=25$ should be enough to obtain a sufficient number of packets eligible to discard, as the average packet number of the GOP is 125 packets. The use of a larger window size implies a reduction of the number of packets identified as non-I, decreasing the number of packets eligible for discard in case of network congestion.

Tables 4 and 5 show the percentage of true positives for the videos *Coast-Guard*, *Highway*, *Bridge Far*, *Paris*, and *Soccer*, for the validation sets for the FFTD and ERTD topologies, respectively. For $N=25$, FFTD achieves an average of 55% and ERTD 58%. However, considering the larger windows, the performance of ERTD was consistently better than FFTD. The hit rate in the validation set is compatible with the classification results reported for the IP traffic classification methods.

The data sets SW-1, SW-2, SL, and JP were used the same way as the videos presented before, with 70% for training and 30% for validation. For the SW-3, however, we took another strategy: SW-3 were used only for validation, with the ANN trained with the SW-2 set. We are interested in whether a trained ANN can produce good results in the classification of an arbitrary movie subset. Tables 6 and 7 present the true positive percentage for training *SW*-1, *SW*-2, *SL*, and *JP* for the topologies FFTD and ERTD, respectively. From $N=16$, both ANN topologies could be trained with good results: the averages of true positive were 99% and 97%, respectively. The results show a better performance for larger window size, as in the videos before.

Tables 8 and 9 present the true positive percentage for the validation sets of movies, as well as the true positive percentage for SW-3 submitted to an ANN trained with SW-2. SW-1 presents sudden scene changes, and the classifier achieves 58% of success for $N=13$ for both FFTD and ERTD. For the more regular scene pattern of SW-2, SL, and JP, the true positives using FFTD for $N=13$ were respectively 68%, 64%, and 58%. Using ERTD with $N=13$, the true positives were respectively 71%, 63%, and 67%. The average performance of the classifier for SW-3 with $N=13$ was 64% and 71%, for the FFTD and ERTD, respectively. For SW-3, with $N=27$, FFTD and ERTD achieve 95% and 100%, respectively.

Table 6
Percentage of true positives using the FFTD topology for video sets SW-1, SW-2, JP, and SL for the *training* set.

N	*SW*1 (%)	*SW*2 (%)	*JP* (%)	*SL* (%)	Average (%)
10	51	95.1	72	71.4	72
13	90.4	99.4	99.7	92.7	96
16	98.4	99.7	99.7	100	99
21	98	100	100	100	100
24	99	100	100	99.7	100
27	98.5	98.1	100	99.3	99

Table 7
Percentage of true positives using the ERTD topology for video sets SW-1, SW-2, JP, and SL for the *training* set.

N	*SW*1 (%)	*SW*2 (%)	*JP* (%)	*SL* (%)	Average(%)
10	57.3	94.9	96.8	75.9	81
13	91.2	98.6	99.7	96.1	96
16	95.3	95.3	100	97.5	97
21	99.3	100	100	100	100
24	98.6	100	100	99.7	100
27	100	99.6	100	100	100

Table 8
Percentage of true positives using the FFTD topology for the videos SW-1, SW-2, SW-3, JP, and SL for the *validation* set.

N	SW1 (%)	SW2 (%)	SW3 (%)	SL (%)	JP (%)	Average (%)
10	32	72	68	65	55	58
13	58	68	64	64	58	62
16	49	81	76	65	57	65
21	48	83	86	73	64	71
24	54	86	94	75	58	74
27	52	87	95	84	73	78

Table 9
Percentage of true positives using the ERTD topology for the videos SW-1, SW-2, SW-3, JP, and SL for the *validation* set.

N	SW1 (%)	SW2 (%)	SW3 (%)	SL (%)	JP (%)	Average (%)
10	36	64	57	68	63	58
13	58	71	71	63	67	66
16	51	76	75	64	58	65
21	48	81	86	68	54	68
24	53	85	98	77	58	74
27	50	86	100	82	71	78

Table 10
Training and validation for the video *salesman*.

N	Training (%)		Validation (%)	
	FFTD	ERTD	FFTD	ERTD
7	100	95	76	67
8	100	100	97	93
9	100	100	85	81
11	100	100	71	81
12	100	100	77	81

It is important to notice the increasing hit rate with larger window sizes. This behavior was observed in all validation sets. One can observe in Tables 8 and 9 that the averages considering all movies are quite similar for FFTD and ERTD, varying from 58% with $N=10$ to 78% with $N=27$.

As the minimum size of the window is specified by the number of packets carrying I-frames and the maximum size is limited by the number of packets in the GOP, we suggest the use of the following relation for determining the window size:

$$N=\phi_I+\alpha\cdot(A\cdot\phi_P+B\cdot\phi_B),\quad 0<\alpha<1 \tag{1}$$

where ϕ_I, ϕ_P and ϕ_B represents the number of packets, on average, to carry I-, P- and B-frames, respectively. *A* and *B* represent, respectively, the number of P- and B-frames of GOP. The results suggest good performance of classification with $\alpha=0.1$.

The video *salesman* was employed to compare the results with a different quality and GOP configuration. The video coding was made in the same way as performed in Hong and Won (2010). We employed a window size of 7, 8, 9, 11, and 12. Table 10 presents the results for training and validation, which indicate that the classification could be performed with a good true positive percentage - for instance, with $N=8$, the true positive is 97% for FFTD and 93% for ERTD.

3. Selective packet discard

The Packet Discard Algorithm (PDA) manages the queues of a network element and is responsible for discarding the packets in case of queue congestion. Among the available queue management methods, the most known is the drop tail. The drop tail is a simple queue management algorithm: when the queue is occupied to its maximum capacity, the newly arriving packets are discarded. Other popular options are Random Early Detection (RED) and Weighted RED (WRED), which could drop packets even before the queue is totally filled, as a warning to the congestion control mechanisms on the traffic sources, to reduce their transmission rate and help the network. However, only protocols like TCP (Transmission Control Protocol) and SCTP (Stream Control Transmission Protocol) are able to dynamically adjust its transmission rate based on the packet loss rate. Time-sensitive applications, as video streaming, often use UDP (User Datagram Protocol) because dropping packets is preferable to waiting for delayed packets. None of these can perform a selective packet discard based on information of application. The performance of H.264 with active queue management (AQM) was investigated in Torres et al. (2012).

We propose a priority packet discard mechanism to drop packets according to the classification made by the ANN. The method has three steps:

1. The time between successive arrivals of packets and the size of the last *N* packets received are stored and used as input of the ANN.

2. Packet classification is performed according to the output of neural network, x_k. If $x_k > Lim_1$ the presence of packets carrying information about an I-frame is assumed, and in this case the packets will be marked as *green*. The mark is done in an auxiliary data structure, without changing any values in the IP header. If $x_k < Lim_2$, it is assumed that the packets in the window do not carry information about I-Frames and they will be marked as *red*. If the output is in the range $Lim_2 \le x_k \le Lim_1$, the packets will be marked as *yellow*.
3. If the queue capacity reaches its limit, the proposed method discards the red packets first, then the yellow, and finally the green. In the tests, Lim_1 and Lim_2 were configured, respectively, with 0.1 and 0.9.

In the method we propose, the computational complexity depends almost on the packet classifier, which has the worst-case computational complexity given by $O(n)$, n being the window size.

4. Results

The efficiency of the proposed method was measured through a queue simulator, developed in C language, to evaluate the performance of a queue fed by real traffic. The queue simulator was carefully validated by comparing the results with known analytical models, as indicated in Banks et al. (2001), accordingly the $M/M/1/\infty$ and $M/M/1/B$ queue systems, and the simulated results are consistent with the analytical models.

In the first test, the bottleneck link rate was configured at 90% of queue utilization and the maximum queue size was varied. In this case, the burstiness of video could cause packet discard for a limited length of time. In the second test the queue depth was kept constant and the queue utilization varied. For all tests, the number of discarded packets carrying I-frames was observed and compared with drop tail and the golden standard. The golden standard represents the best possible performance in a given network congestion situation.

The mean opinion score (MOS) is a subjective evaluation of the quality of a video transmission; it depends on the impression a human observer has on the delivered video, as described in ITU-T recommendation BT 500 (Recommendation ITU-T BT.500, 2012). The MOS is one of the most commonly used metrics to estimate QoE and is expressed by a number, 1 being the worst and 5 the best perceived quality. In contrast, objective video quality metrics are calculated by computers. The most relevant metrics in the area of video quality assessment are PSNR and Structural Similarity (SSIM) (Serral-Gracià et al., 2010). Recent results show that SSIM presents better approximation of human subjective evaluation of quality (Silpa and Mastani, 2012). However, we need to evaluate the relative quality, allowing comparison of between *golden standard*, *drop tail*, and the proposed method. A study presented in Hore and Ziou (2010) indicates that both PSNR and SSIM are capable to capture the degradation of video quality, and that a simple analytical link exists between the PSNR and the SSIM. Thus, the simple metric of PSNR is still adequate to compare the efficiency of proposed method and will reduce the computational overhead of quality evaluation.

Thereby, MOS was estimated employing the Evalvid tool (Klaue et al., 2003). The Evalvid compares the original video image with the video received to estimate MOS, through the evaluation of PSNR. The PSNR is calculated frame by frame using the mean squared error (MSE) given by

$$MSE = \frac{1}{rc} \sum_{i=1}^{r} \sum_{j=1}^{c} [Y_o(i,j) - Y_r(i,j)]^2 \qquad (2)$$

Table 11
PSNR to MOS conversion (Klaue et al., 2003).

PSNR (dB)	MOS
> 37	5 (excellent)
31–37	4 (good)
25–31	3 (fair)
20–25	2 (poor)
< 20	1 (bad)

where r and c represent, respectively, the number of rows and columns of image, $Y_o(i,j)$ and $Y_r(i,j)$ represent the luminance of pixel (i,j) of original and received frame, respectively. The PSNR can be obtained using

$$PSNR = 20 \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \qquad (3)$$

where MAX_I represents the maximum value of pixel intensity. For the videos in consideration, $MAX_I = 255$.

The conversion between PSNR and MOS was done using the convention presented in Table 11. The PSNR of video was computed with an average of PSNR of all video images.

As mentioned previously, a strategy of marking traffic on the server, in cooperation with the application, was also implemented. In case of network congestion, the SPD first drop packets carrying B-frames, followed by packets carrying P-frames finally packets carrying I-frames. This strategy was implemented as a reference and is called the golden standard.

Fig. 6(a) and (b) shows the percentage of packets carrying I-frames dropped as a function of maximum queue size, with link utilization of 90%, for the videos *Highway* and *Bridge Far*. For each figure, three lines are presented: for the proposed method, drop tail, and golden standard. The proposed method presents a better performance than drop tail for both videos. Fig. 6(c) and (d) presents the MOS evaluation for the same videos. It can be seen that the method prevents QoE degradation, e.g. drop tail achieves a MOS of 2.8 and the proposed method 4.5 when queue size is 6000 bytes for the video *Bridge Far*. For the golden standard, the packets carrying I-frames were preserved and the degradation of MOS is consequence of delay and jitter. Additionally, when the queue depth increases, the performance of the proposed method approaches the golden standard.

For the second test, Fig. 7(a) and (b) presents, respectively, the percentage of packets carrying I-frames discarded and the MOS evaluation for several queue utilization levels, with a fixed queue size, for the video *Highway*. The proposed method outperforms drop tail in all cases, preventing I-packets from being dropped and thus increasing the MOS.

The performance of the proposed method was also analyzed for the movies subsets SW-1, SW-2, SW-3, SL, and JP. These movies have different characteristics from the other videos analyzed earlier. The sudden scene changes of movies cause variation on the frame sizes, decreasing the efficiency of the classifier. The subsets were submitted to the queue simulator. Fig. 8 shows the percentage of packets carrying I-frames discarded for each subset with several values of queue maximum queue depth, for utilization of 90%. In all cases, the ANN achieves a good success rate in preserving the packets carrying I-frames, with a performance similar to that of the golden standard for the higher values of queue depth. Fig. 9 shows the MOS estimation. The MOS improved for the SW-1, SL, and SP, respectively, presented in Fig. 9(b)–(d). For the SW-2, with sudden scene changes, the method seems to have a worse MOS than drop tail, as shown in Fig. 9(a). However, the ANN was actually able to identify the packets carrying I-frames, as presented in Fig. 8(a). With the sudden scene changes, the P- and B-frames become important because the aggressive

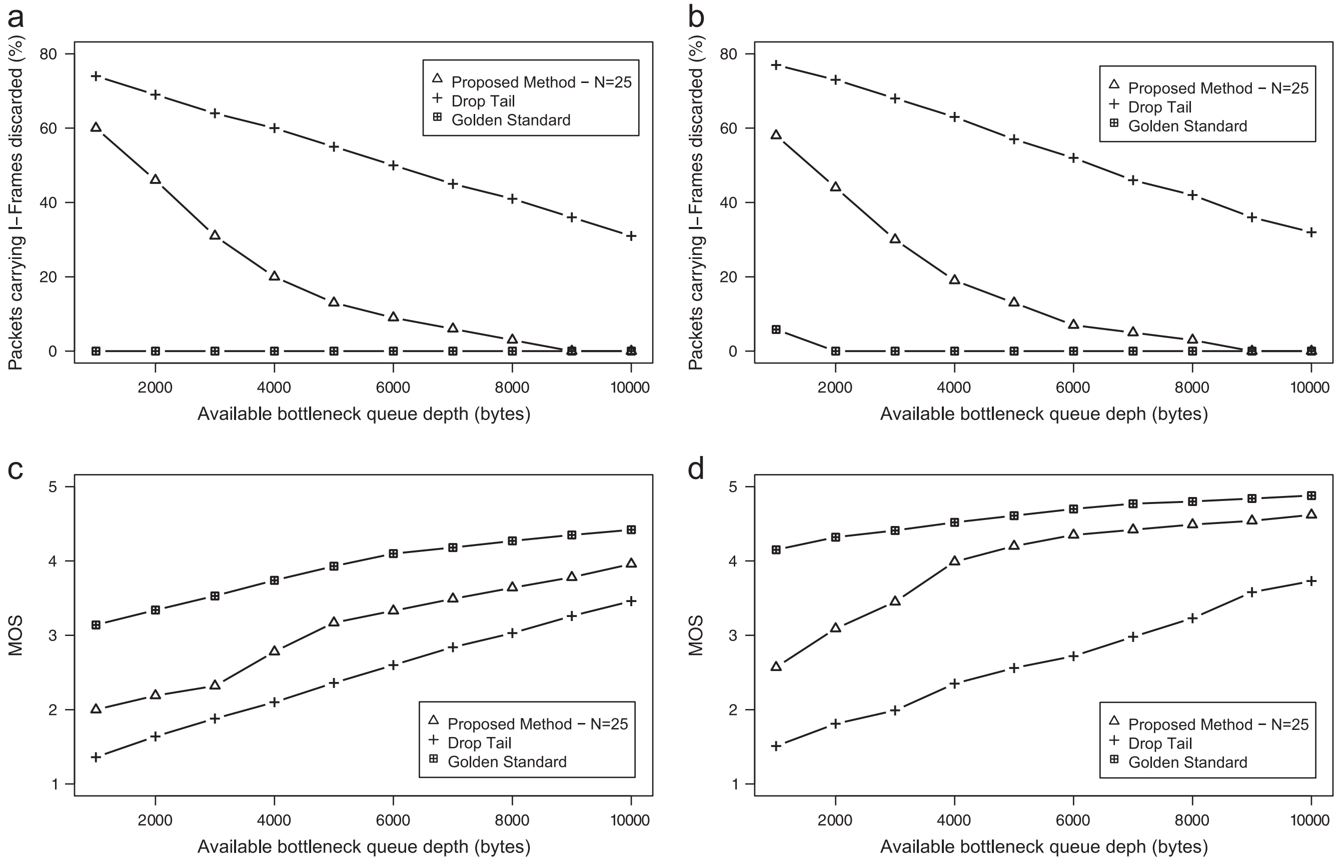


Fig. 6. Percentage of packets carrying I-frames discarded for the videos Highway (a) and Bridge Far (b), and MOS measure for the videos Highway (c) and Bridge Far (d).

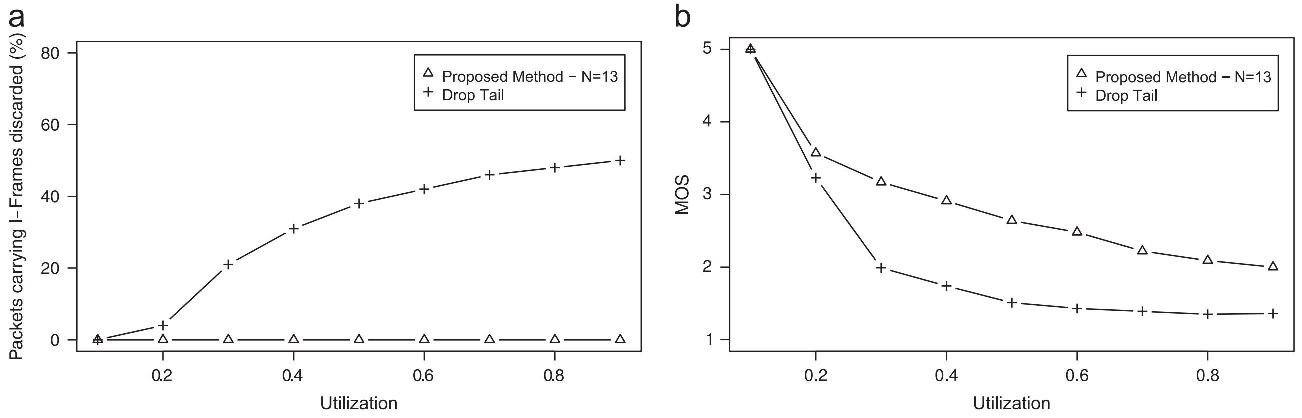


Fig. 7. Percentage of packets carrying I-frames discarded for the videos (a) and MOS evaluation for the video Highway at several queue utilization levels (b).

changes in scenes decorrelate the first image of GOP with subsequent images, and in this case, the proposed SPD strategy does not benefit the quality of experience.

The ANN previously trained with SW-2 was employed to verify the feasibility of using a standard trained ANN to traffic classification. The transmission of SW-3 subset was simulated. Fig. 10 (a) presents the percentage of packets carrying I-frames discarded and Fig. 10(b) shows the MOS evaluation for this case. The results show an improvement of MOS compared with drop tail, with good results for $N = 13$. In order to extend this test we randomly selected two sequences from Star Wars Ep. IV. Both sequences are 2 min long. The percentage of packets carrying I-frames discarded is shown in Fig. 11, respectively, for the two sequences. Its possible to see that ANN successfully preserves from discarding a good number of packets carrying I-frames, and this results in an improvement of QoE. This indicates that it is possible to employ a standard trained ANN to implement the strategy in routers.

Even if the buffers of routers are large enough to avoid dropping packets, a deadline could be establish for the transmission of each packet in queue, and the packets that exceed this deadline could be considered lost. From this point, the proposed method can be applied.

Another important result is the performance of golden standard. For all scenarios, the estimated MOS was greatly improved for golden standard, and this indicates that the use of this method is possible to provide a good quality with higher network utilization levels. This implies in lower infrastructure costs for video transmission networks if compared to the scenario where the proposed method is not used. Additionally, the use of the golden standard should be preferred compared to the method using

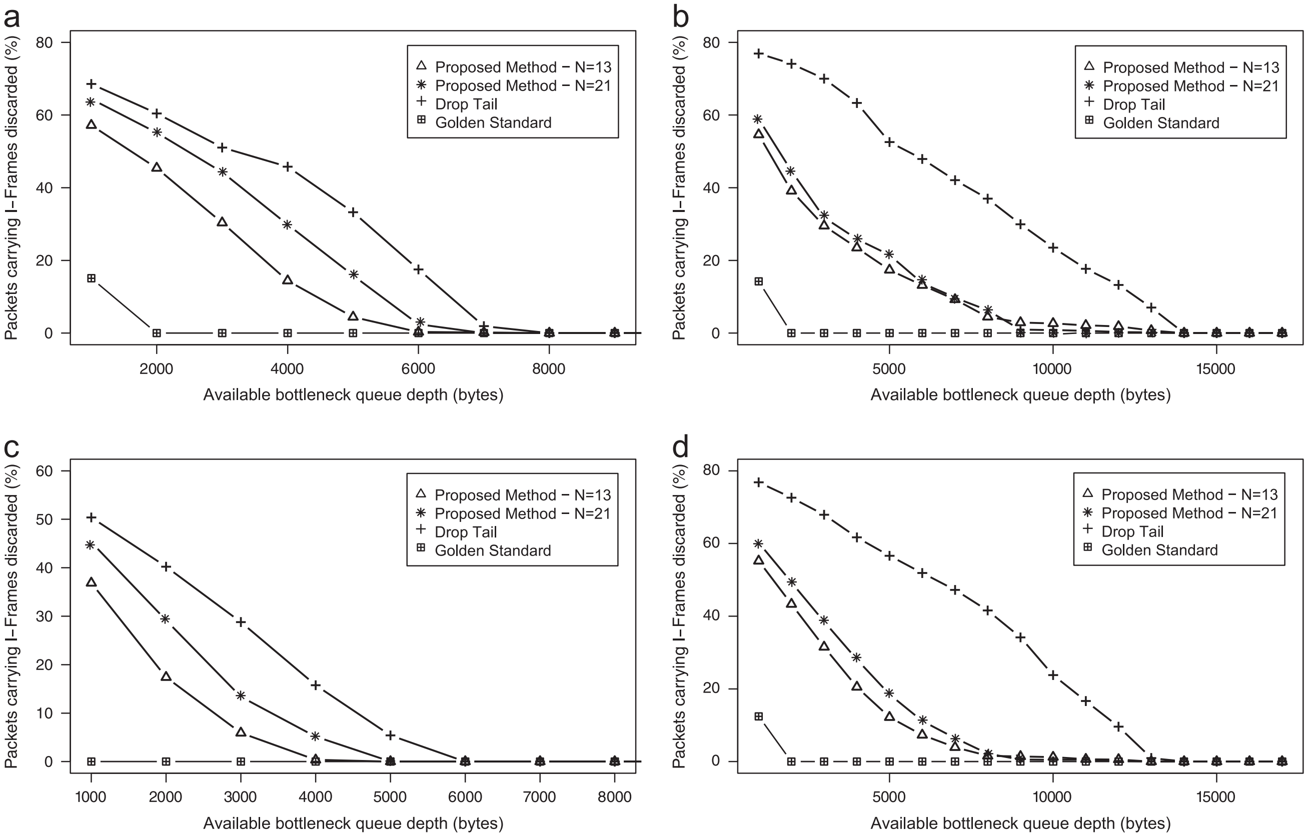


Fig. 8. Percentage of packets carrying I-frames discarded for SW-1 (a), SW-2 (b), SL (c), and JP (d).

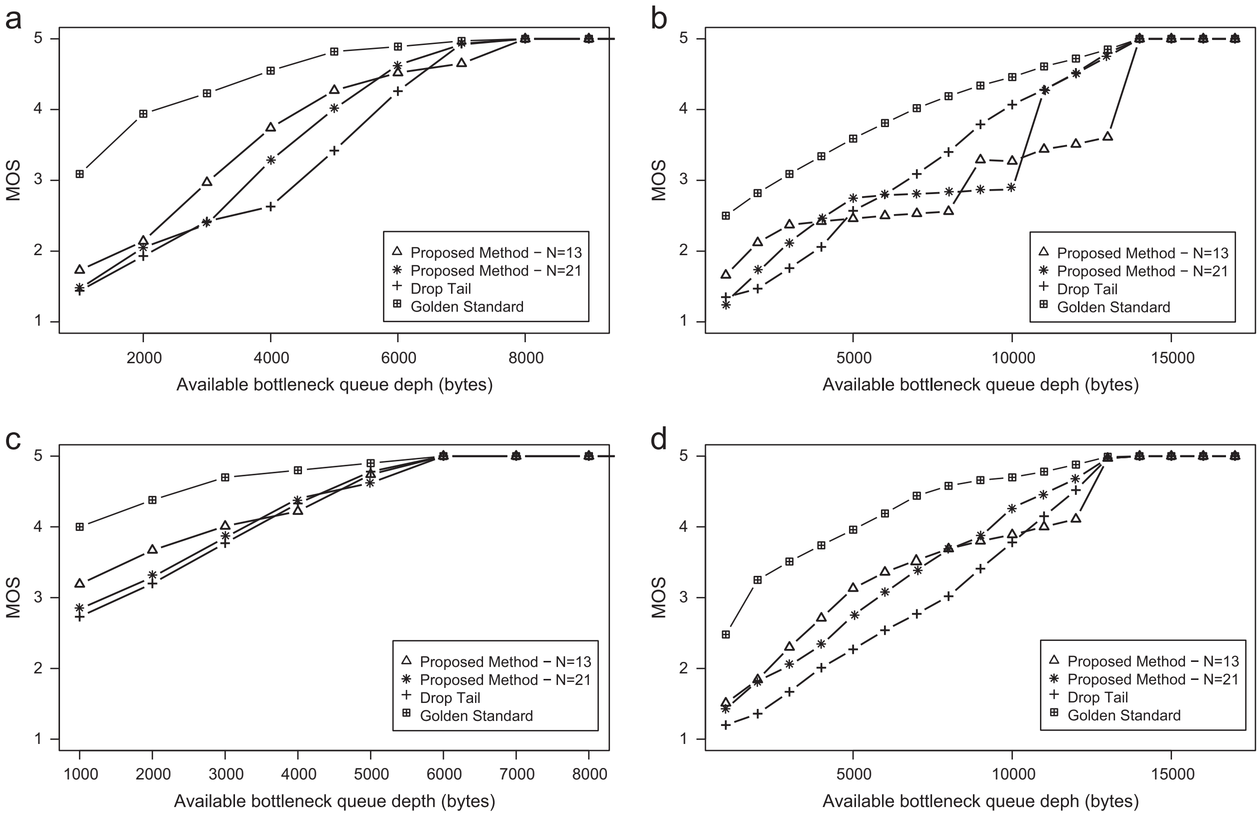


Fig. 9. MOS estimation for SW-1 (a), SW-2 (b), SL (c), and JP (d).

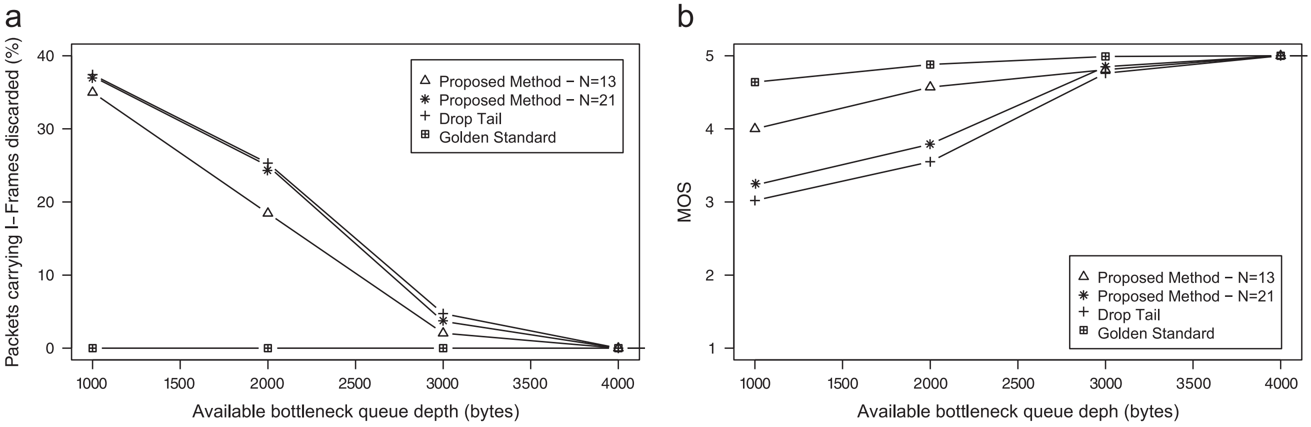


Fig. 10. Percentage of packets carrying I-frames discarded for SW-3 (a) and MOS evaluation for SW-3(b).

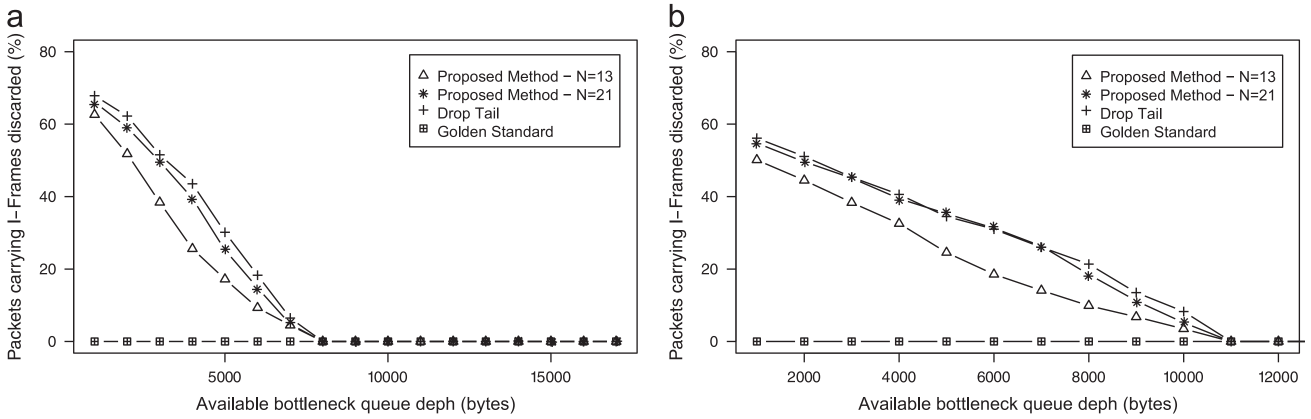


Fig. 11. Percentage of packets carrying I-frames discarded for two sequences of 2 min randomly selected from Star Wars Ep.IV.

classification with ANNs. However, in the situations where it is not possible to perform the jointly configuration required to implement the golden standard, this alternative results in a performance improvement for the system.

5. Conclusions

In this paper we propose a selective packet discard strategy, at network routers, to preserve user QoE in case of network congestion. Even if the transmission network is well planned, packet loss can occur due the burstiness of video traffic. A packet that has exceeded its delay in a maximum threshold may also be considered a loss.

The impact of packet loss on QoE can have several degrees of severity, depending on which packet is discarded. The most relevant packets are those that carry information about I-frames, because they are used by the decoder as references for decoding other frames. Thus, if the queue becomes congested and packet discard is unavoidable, the preservation of these packets leads to a better QoE. The standard method for packet discard is drop tail, but this method is not aware of the payload of the discarded packet.

We proposed the use of neural networks to classify the packets using only information about their size and time interval. We simulated a priority discard queue where packets classified as I-frames are preserved and packages classified as P- and B-frames are preferably discarded. This resulted in an improvement of user perceived QoE. We showed how much improvement can be obtained in many videos used as examples. We also compared the video quality improvement due to neural network classification with the ideal scenario where all packets were correctly classified at the source. The proposed classification scheme can be performed with available information in network layer protocol, without the decoding of higher protocol layers.

References

Advanced Video Coding for Generic Audiovisual Services. Recommendation ITU-T H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), January 2009.

Abdennour, A., 2006. Evaluation of neural network architectures for MPEG-4 video traffic prediction. IEEE Transactions on Broadcasting 52 (2), 184–192.

Banks, J., Carson II, S.J., Nelson, L.B., Nicol, M.D., 2001. Discrete-Event System Simulation, 3rd edition Prentice Hall.

Basu, J.K., Bhattacharyya, D., hoon Kim, T., 2010. Use of artificial neural network in pattern recognition. International Journal of Software Engineering and its Applications 4 (2), 23–34.

Bouras, C., Gkamas, A., Kioumourtzis, G., 2009. Evaluation of single rate multicast congestion control schemes for mpeg-4 video transmission. In: Proceedings of the 5th Euro-NGI Conference on Next Generation Internet Networks, NGI'09. IEEE Press, Piscataway, NJ, USA, pp. 32–39.

Callado, A., Kamienski, C., Szabo, G., Gero, B., Kelner, J., Fernandes, S., Sadok, D., 2009. A survey on Internet traffic identification. IEEE Communications Surveys Tutorials 11 (3), 37–52.

Dai, M., Zhang, Y., Loguinov, D., 2009. A unified traffic model for MPEG-4 and H2.64 video traces. IEEE Transactions on Multimedia 11 (5), 1010–1023.

Fischer, I., Hennecke, F., Bannes, C., Zell, A., 2001. Java Neural Network Simulator – User Manual, Version 1.1. Wilhelm-Schickard-Institute for Computer Science, University of Tubingen.

Greengrass, J., Evans, J., Begen, A.C., 2009. Not all packets are equal. Part II. The impact of network packet loss on video quality. IEEE Internet Computing 13, 74–82.

Haykin, S., 1998. Neural Networks: A Comprehensive Foundation, 2nd edition Prentice Hall PTR, Upper Saddle River, NJ, USA.

Hong, S., Won, Y., 2010. Incorporating packet semantics in scheduling of real-time multimedia streaming. Multimedia Tools Applications 46, 463–492.

Hore, A., Ziou, D., 2010. Image quality metrics: PSNR vs. SSIM. In: 20th International Conference on Pattern Recognition (ICPR), 2010, pp. 2366–2369.

Huang, Y., Guérin, R., Gupta, P., 2006. Supporting excess real-time traffic with active drop queue. IEEE/ACM Transactions on Networking 14, 965–977.

Klaue, J., Rathke, B., Wolisz, A., 2003. Evalvid – a framework for video transmission and quality evaluation. In: Proceedings of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, pp. 255–272.

Klein Jr., V., Pedroso, C., 2013. SAVI – A model for video workload generation based on scene length. In: Advanced Infocomm Technology, Lecture Notes in Computer Science, vol. 7593. Springer, Berlin, Heidelberg, pp. 116–127.

Maisonneuve, J., Deschanel, M., Heiles, J., Li, W., Liu, H., Sharpe, R., Wu, Y., 2009. An overview of IPTV standards development. IEEE Transactions on Broadcasting 55 (2), 315–328.

Marpe, D., Wiegand, T., Sullivan, G., 2006. The H.264/MPEG4 advanced video coding standard and its applications. IEEE Communications Magazine 44 (8), 134–143.

Mishra, A., Zaheeruddin, Raj, K., 2007. Implementation of a digital neuron with nonlinear activation function using piecewise linear approximation technique. In: International Conference on Microelectronics, 2007 (ICM 2007), pp. 69–72.

Nguyen, T., Armitage, G., 2008. A survey of techniques for Internet traffic classification using machine learning. IEEE Communications Surveys Tutorials 10 (4), 56–76.

Niedermayer, M., February 2012. FFmpeg, Online. URL 〈http://ffmpeg.org/〉.

Principe, J.C., Euliano, N.R., Lefebvre, W.C., 1999. Neural and Adaptive Systems: Fundamentals Through Simulations, John Wiley and Sons.

Ramakrishnan, K., Floyd, S., Black, D., September 2001. The Addition of Explicit Congestion Notification (ECN) to IP, RFC 3168 (Proposed Standard).

Recommendation ITU-T BT.500, January 2012.

Richardson, M., Fenner, B., February 2012. TCPDUMP and LIBCAP, Online. URL 〈http://www.tcpdump.org/〉.

Schier, M., Welzl, M., 2009. Selective packet discard in mobile video delivery based on macroblock-based distortion estimation. In: Proceedings of the 28th IEEE International Conference on Computer Communications Workshops, INFOCOM'09. IEEE Press, Piscataway, NJ, USA, pp. 212–217.

Seeling, P., Reisslein, M., 2012. Video transport evaluation with H.264 video traces. IEEE Communications Surveys Tutorials PP (99), pp. 1–24.

Serral-Gracià, R., Cerqueira, E., Curado, M., Yannuzzi, M., Monteiro, E., Masip-Bruin, X., 2010. An overview of quality of experience measurement challenges for video applications in ip networks. In: Osipov, E., Kassler, A., Bohnert, T., Masip-Bruin, X. (Eds.), Wired/Wireless Internet Communications. Lecture Notes in Computer Science, vol. 6074. Springer, Berlin, Heidelberg, pp. 252–263, URL http://dx.doi.org/10.1007/978-3-642-13315-2_21.

Silpa, K., Aruna Mastani., 2012. Comparison of image quality metrics. International Journal of Engineering Research and Technology (IJERT) 1(4). URL < http://www.ijert.org/browse/june-2012-edition?start=20 > .

Szabó, T., Horváth, G., 2004. An efficient hardware implementation of feed-forward neural networks. Applied Intelligence 21 (2), 143–158.

Szymanski, T., Gilbert, D., 2009. Internet multicasting of IPTV with essentially-zero delay jitter. IEEE Transactions on Broadcasting 55 (1), 20–30.

Torres, D.P., Fernandez, E.M.G., Ribeiro, E.P., Reguera, V.A., de Oliveira, C., 2012. On the impact of adaptive RED in IP networks transporting H.264/MPEG-4 AVC video streams. Computers and Electrical Engineering 38 (1), 128–139.

Van der Auwera, G., David, P., Reisslein, M., 2008a. Traffic characteristics of H2.64/AVC variable bit rate video. IEEE Communications Magazine 46 (11), 164–174.

Van der Auwera, G., David, P., Reisslein, M., 2008b. Traffic and quality characterization of single-layer video streams encoded with the H.264/MPEG-4 advanced video coding standard and scalable video coding extension. IEEE Transactions on Broadcasting 54 (3), 698–718.

Video Trace Library, Arizona State University, January 2012. Online. URL 〈http://trace.eas.asu.edu/〉.

The Wireshark Team, February 2012. Wireshark. Online. URL 〈http://www.wireshark.org/〉.

Xiaogang, Y., Jiqiang, L., Ning, L., 2007. Congestion control based on priority drop for H.264/SVC. In: International Conference on Multimedia and Ubiquitous Engineering, MUE'07, pp. 585–589.

Zell, A., Mamier, G., Vogt, M., Mache, N., Hubner, R., Doring, S., uwe Herrmann, K., Soyez, T., Schmalzl, M., Sommer, T., Hatzigeorgiou, A., Posselt, D., Schreiner, T., Kett, B., Clemente, G., Reczko, M., Riedmiller, M., Seemann, M., Ritt, M., Decoster, J., Biedermann, J., Danz, J., Wehrfritz, C., Werner, O., Berthold, M., 2011. SNNS: Stuttgart Neural Network Simulator – Manual Extensions of Version 4.0. Online. URL 〈http://www.ra.cs.uni-tuebingen.de/SNNS/〉.