Optical Switching and Networking 10 (2013) 327–342

Contents lists available at SciVerse ScienceDirect

Optical Switching and Networking

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

Effective edge node configuration for video transport over Optical Burst Switched networks

F. Espina *, D. Morato, M. Izal, E. Magaña

Public University of Navarre, Campus de Arrosadía s/n, E-31006 Pamplona, Spain

ARTICLE INFO

Article history:
Received 2 July 2012
Received in revised form
7 March 2013
Accepted 8 March 2013
Available online 22 March 2013

Keywords:
Optical Burst Switching
Bursty losses
Video quality metrics
Traffic Engineering
Video multiplexation

ABSTRACT

This paper considers digital video transport over Optical Burst Switched networks where burst losses cause data loss from one or more adjacent video frames. Analytical approximations for the frame losses and video playback interruptions are derived and validated using simulations. Both parameters require a very limited and static amount of data about the video on the user side and some quality of service metrics about the network to quantify the quality of the received video. The results take into account the strong dependency in the video traffic structure due to the coding mechanisms. The critical effect of video coding parameters is also revealed. The paper also presents a Traffic Engineering procedure to select the best parameters for the edge node and the video codec to meet a given video quality level on the user side.

1. Introduction

Internet video was 40% of consumer Internet traffic in 2011 and it is expected to rise to more than 60% by the end of 2015 [1]. Therefore, high-speed optical switching backbones will be necessary for video content delivery [2,3].

Video traffic is bursty. It generates large periodic data bursts (inter-frame time T_{if}). An Optical Circuit Switching (OCS) technology is no efficient for the individual transport of video frames due to the cost of creating and destroying the optical path for each one. Therefore, the path would last for the whole video duration, including the periods of time when the source does not generate traffic, resulting in an inefficient use of the optical bandwidth.

The biggest problems of Optical Packet Switching (OPS) are the lack of optical RAM, needed for buffering packets during contention periods, and the need of very high switching rates. These needs are amplified in the case of bursty traffic like video, where the number of short optical packets competing for the same backbone network resources can vary significantly.

Optical Burst Switching (OBS) is an intermediate solution. In OBS, electronic packets are aggregated into large bursts and these bursts are routed and switched through the optical network as independent optical packets. The optical path is established for each burst using one way reservation, lessening the burden of circuit establishment times, and there is no need of very high switching rates as the optical packets are large.

There are various proposals of hybrid switching architectures [2,4,5] that can be used to transport video traffic instead of using 'standard' OBS. Some of these proposals combine OBS with other switching technologies and they can use the results and conclusions from this paper to improve the received video quality. For example, in an OCS+OBS hybrid network like HOS [6], the video usually will be transported by OCS to ensure perfect delivery. When the OCS network does not have a source–destination path available or when the available path does not have enough bandwidth, the whole video flow should be

* Corresponding author. Tel.: +34 948 166033.
E-mail addresses: felix.espina@unavarra.es (F. Espina), daniel.morato@unavarra.es (D. Morato), mikel.izal@unavarra.es (M. Izal), eduardo.magana@unavarra.es (E. Magaña).

1573-4277/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.osn.2013.03.003

discarded or the OBS network should be used for its transport. In this case, the results from this paper can be used to compute the received video quality depending on the OBS network configuration or to choose the OBS parameters to obtain a minimum acceptable video quality.

An Optical Burst Switched (OBS) network [7] is considered in this paper because of the potential it offers for the core of future ISP networks. Edge nodes to an OBS network aggregate input traffic into bursts. The most frequently used OBS burst formation mechanism (a.k.a. *burstifiers*) in the literature is timer-based [8]. A timer of value T_{out} is started on the arrival of a packet to an empty burst formation queue. When the timer expires, the burst is scheduled for transmission on the output port. These bursts use an all-optical data plane from the ingress to egress node where they are disassembled into their contained packets.

For efficient network transport, video flows are compressed, mainly using encoders from the MPEG family [9] or proprietary (but similar in concept) encoders [10]. Given the widespread use of these encoders throughout the industry and reaching millions of user devices, they will most likely remain popular for the foreseeable future.

The coding process in MPEG standards takes advantage of similarities in frames close in time to produce smaller compressed frames. Therefore, the decoding process for most video frames requires previously decoded frames, and the loss of one frame prevents the correct decoding of the following frames [11]. The packet aggregation procedure used on OBS creates bursts that could often contain packets from several contiguous video frames, therefore producing bursty losses of packets and frames in the video flow. The loss of a few consecutive frames could create a large gap in video playback, presenting a serious negative impact on the quality experienced by the user.

A video provider needs to be able to observe and react quickly to these quality problems, optimally before the user perceives them. The concept of *Quality of Experience* (*QoE*) emerged from this identified need. ITU-T defines QoE [12] as "The overall acceptability of an application or service, as perceived subjectively by the end-user". QoE includes all the end-to-end system effects (e.g., client, terminal, network, services infrastructure), and the overall acceptability may be influenced by the user's expectations and context.

Subjective quality metrics are concerned with how video is perceived by a user. Metric tests are expensive in terms of time (preparation and running) and human resources because they must collect user opinions on video quality. Objective quality metrics are mathematical models that approximate the results of subjective quality assessment but are based on metrics that can be measured objectively and automatically evaluated by a computer program.

Objective quality metrics are categorized based on the availability of the original video signal [13]. In full reference (FR) metrics, the entire original video signal is available. In reduced reference (RR) metrics, only some of the information of the original video is available. In no reference (NR) metrics, the original video is completely unavailable. The signal received by the user is assumed to be available for the three metrics.

FR metrics require the original and the received video to be available together at some point. RR metrics require the computation and centralization of some structural parameters from the original and the received video. NR metrics can be computed at the user side based only on the received video. Therefore, NR metrics are the easiest to implement and use for Traffic Engineering in a real video distribution network. This paper analyses two NR metrics mathematically and by simulation.

Popular NR metrics are based on network losses and their effects on the decoding procedure. Due to the bufferless nature of the OBS optical switches, output port contention results in drops of bursts. The proportion of lost packets is a *Quality of Service* (*QoS*) metric about the transport network, but according to the definition given above, it cannot be considered a QoE metric. The proportion of lost frames, named the *Frame Loss Ratio* (*FLR*), is closer to what is expected of a QoE metric but is rather a QoS on the application level metric because it does not include the frames that are not lost but have not been able to be decoded and displayed to the user. The proportion of frames that could not be decoded and displayed can be considered a QoE metric [14]. This proportion is called the *Frame Starvation Ratio* (*FSR*) and its complementary, the *Decodable Frame Rate* (*Q*), was proposed in [15]. It was assumed that direct frame losses are mutually independent, but this assumption cannot be taken in network scenarios with bursty losses, where usually more than one adjacent frame is lost with high probability.

User perception is affected not only by the number of non-displayed frames but also by their grouping in time. This grouping is reflected in the length of the video playback interruptions or cuts. Experimental measurements [16] have shown that video playback interruptions of 200 ms are certainly visible to the user. Even video cuts of only 80 ms may be visible. OBS losses are bursty, so cuts of that length or larger can be frequently generated. Therefore, it is important to analytically describe the relationship between the OBS parameters (e.g., timer, multiplexation level) and the video playback interruption lengths.

The experimental measurements in [16] showed that users seem to be more annoyed by several short video playback interruptions than by one single video playback interruption of the same duration, and they are more annoyed when the video playback interruptions are of different lengths than when they are of the same duration. Therefore, both the number of non-displayed frames and the video playback interruption lengths must be taken into account for video quality evaluation.

In this paper, quality metrics that depend on network parameters are analytically derived for network scenarios with bursty losses. The results allow simple quality prediction for the design and tuning of video distribution networks. Previous works have studied the transport of video over OBS networks [17–19], but none of them have derived analytically the relationship between the network OBS parameters (e.g., timer, loss ratio) and the degradation of video quality.

The rest of the paper is organized as follows. Section 2 introduces the two proposed network scenarios for the

transport of video over an OBS network without and with the multiplexation of videos. Section 3 introduces some important parameters of encoded video. Sections 4 and 5 present the analytical study of the two proposed scenarios. Section 6 presents the validation of these analytical equations. Section 7 shows the contribution of some codec parameters to the quality degradation caused by the network. Section 8 presents a Traffic Engineering procedure for tuning the burstification process to meet a given quality level. Finally, Section 9 concludes this paper.

2. Network scenario

Video servers usually send all packets from a frame back-to-back. These packets will arrive at the edge node burstifier at approximately the same time. In an OBS network with timer-based burstifiers, if a packet from a frame gets into a burst, we assume that the rest of the packets from the frame do so as well. Therefore, by using a timer-based burstifier, bursts will contain one or more whole frames. Inside the OBS network, due to the bufferless nature of the optical switches, output port contention results in the drop of bursts. The loss of one of these bursts (due to output port contention or bit errors) results in a number of lost and non-decodable video frames.

Fig. 1 shows the scenarios under study. In the first scenario, each video flow uses an independent burstifier not shared with other videos. Several frames from the same video could be aggregated into the same burst. In the second scenario, all the video flows from the same ingress node that address the same egress point are aggregated into the same burst formation buffer with timer T_{out}.

The analytical results presented in this paper also apply to any other network technology where video frame losses could take place in bursts. For example, Digital Video Broadcasting-Handheld (DVB-H) [20] technology can present this behaviour. DVB-H is a technical specification for bringing broadcast services to battery-powered handheld devices. In DVB-H, data packets are transmitted as bursts in small time slots at the channel bit rate (greater than the original flow bit rate). A timer-based burstification process similar to the one used on OBS could be employed. Thus, the results for lost frames and video playback interruptions obtained for OBS will be valid for a DVB-H environment with this burstification process.

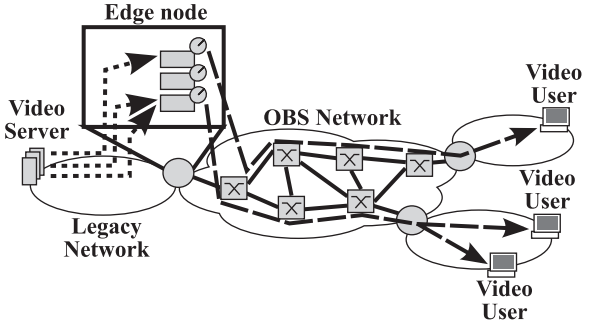

Fig. 1. OBS network scenario under study: timer-based burstifiers with one or more video flows.

3. Video codification

MPEG is standardized in a joint effort by ISO and ITU. MPEG does not standardize the encoder/decoder, but it standardizes the encoded video bit stream that the encoder/decoder exchanges.

The MPEG standards define three types of video frames [21]: intra-coded frames (I-frames), inter-coded or predicted frames (P-frames) and bidirectional coded frames (B-frames). I-frames can be decoded on their own. P- and B-frames hold only the changes in the image from the reference frames, thus improving the video compression rates. P-frames have only one reference frame, the previous I- or P-frame. In MPEG-2 Part 2 (H.262) [22], MPEG-4 Part 2 [23] and MPEG-4 Part 10 (MPEG-4 AVC or H.264) [24], B-frames have two reference frames, the previous I- or P-frame and the following one of either type. Optionally, B-frames from H.264 can have up to 16 reference frames, located before or after the B-frame, and even B-frames can be reference frames for other B-frames.

In this paper, videos that have "classic" B-frames will be studied, i.e., B-frames depending only on the previous and following I- or P-frames of either type.

I-, P- and B-frames are grouped into GoPs (Groups of Pictures). A GoP is a sequence of frames beginning with an I-frame up to the frame before the next I-frame. The GoP structure is the pattern of I-, P- and B-frames used inside every GoP. A regular GoP structure is usually described as *GxBy*, where x is the number of frames in the GoP and y is the number of contiguous B-frames. For example, the GoP structure could be *G12B2* or *IBBPBBPBBPBB*. A GoP has $G_P = \lfloor (x-1)/(y+1) \rfloor$ P-frames and $G_B = x-1-G_P$ B-frames divided into blocks of y frames. Different GoP sizes and structures, and even changes of the GoP structure in a stream, are possible. However, the GoP structure is typically not changed for the duration of a video.

Video frames are usually fragmented into several IP packets due to their sizes. The loss of a portion of a frame due to network packet drops could have a lesser or greater effect on the decoding process. In the OBS scenario analysed in this paper, all the packets from a frame are aggregated into a single burst. Therefore, there are no partial frame losses, i.e., either the whole frame is lost or nothing is lost.

The event of not being able to decode a frame affects other frames that are related by the interdependency of the GoP structure. The adopted criterion in this paper is that a frame cannot be decoded if any of the frames it depends on are non-decodable [14,15]. Because the MPEG standard does not define a specific encoder/decoder behaviour, the adopted criterion presents a worst case scenario. Better decoding results could be obtained in a real implementation, but they would have a strong dependency on the ad hoc techniques used by the specific decoder.

As an example, Fig. 2 (left) shows a *G10B2* GoP structure with a non-decodable P-frame due to network packet losses. In the right side of the figure, all the frames that could not be decoded due to that loss are marked with dashed lines. The loss of a P-frame triggers the impossibility to decode all the following frames in the GoP and the B-frames previous to it and after another I- or P-frame.

The *FLR* is defined as F_l/F, where F_l is the number of lost frames from a video in the network and F is the reference total number of frames (see Table 1 for the notation used in the paper). The *FSR* is defined as F_{nd}/F, where F_{nd} is the number of frames that could not be decoded due to losses and inter-frame dependencies. Clearly, $FSR \geq FLR$ because $F_{nd} \geq F_l$.

C_L is the *average video playback interruption length* or *average cut length*, measured in frames. C_F is the *average number of cuts per frame*. Both C_L and C_F account for non-decodable frames due to losses and inter-frame dependencies. Clearly, $C_L = F_{nd}/(F \cdot C_F)$.

FSR, C_L and C_F are NR quality parameters, as no information from the original video is needed to compute their values.

4. Network scenario with one video per burstifier

The scenario in which each burstifier aggregates traffic from one single video flow was partially studied in [25], but only suggestions about how to obtain values for *FLR*, *FSR* and C_L were provided. In this paper, an algebraic and systematic method to compute *FLR*, *FSR* and C_L is demonstrated.

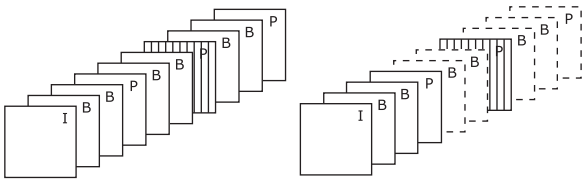

Fig. 2. Example of inter-frame dependence in a GoP.

Optical buffering in the OBS core nodes can be implemented using fibre delay lines, but it is usually non-existent or scarce. Therefore, burst losses in the core network will be the result of output port contention that could not be solved by the core node. In this paper, the network is modelled as a single element including the effect of interfering traffic to the video flow using burst loss ratio p (independent burst losses). This assumption is common in OBS because the losses in the core network are less correlated than in a packet switched network due to its bufferless nature [8].

As the network losses are modelled by an independent burst loss ratio p, the bursts from different burstifiers do not interfere with each other. Therefore, without loss of generality, a network with only one burstifier and one video can be studied. In the analytical derivation, we ignore events of more than one loss per GoP as they are of order p^2, where p will be assumed to be small at the network operating point (less than 10^{-2} for any useful network). In the following sections, the validity of this approximation will be checked.

4.1. Analysis of losses (FLR and FSR derivation)

The burst formation process creates b bursts from a video stream. b_l is the number of lost bursts, $F_b = \lceil T_{out}/T_{if} \rceil$ is the number of frames per burst and $F_{nd}^b[T_{out}]$ is *the*

Table 1
Notation.

Variable	Definition
GxBy	Regular GoP structure (x: number of frames in the GoP; y: number of contiguous B-frames)
G_P	Number of P-frames in a GoP: $\lfloor (x-1)/(y+1) \rfloor$
G_B	Number of B-frames in a GoP: $x-1-G_P$
F	Total number of frames in a video
F_l	Number of lost frames from a video
F_{nd}	Number of non-decoded frames in a video flow
T_{if}	Inter-frame time
T_{out}	Timer value
F_b	Number of frames per burst: $\lceil T_{out}/T_{if} \rceil$
b	Total number of bursts
b_l	Number of lost bursts
p	Burst loss ratio at the network
FLR	Frame Loss Ratio
FSR	Frame Starvation Ratio
C_L	Average cut length, measured in frames
C_F	Average number of cuts per frame
$F_{nd}^b[T]$	Average number of non-decoded frames due to the loss of a burst from a burstifier with timer T
$C_b[T]$	Average number of cuts per lost burst from a burstifier with timer T
N	Average number of multiplexed video flows
$P[F_b]$	Probability that a video contributes to a burst with F_b frames when N videos are multiplexed
$\overline{F_b}$	Average number of frames of a video flow inside a burst when N videos are multiplexed
T_{out}^{min}	Minimum T_{out} that satisfies the C_F constraint for Traffic Engineering
$T_{out}^{min,mul}$	Minimum T_{out} that satisfies the C_F constraint for Traffic Engineering when videos are multiplexed
T_{out}^{max}	Maximum T_{out} that satisfies the C_L constraint for Traffic Engineering (with or without multiplexing)
b_s	Burst size
V_R	Average bitrate of a video
b_s^{max}	Maximum burst size
N_{max}	Maximum number of video flows allowed to be multiplexed into a burstifier

average number of frames that cannot be decoded due to the loss of a burst containing F_b consecutive frames. Remember that T_{out} is the burst formation timer value, T_{if} is the inter-frame time, and *GxBy* is the typical structure of a regular GoP.

It was shown in [25] that the *FLR* is the same as the burst loss ratio p (1). The *FSR* depends on the *FLR*, the inter-frame time T_{if}, the GoP structure, and the burstifier timer T_{out} (2):

$$FLR = \frac{F_l}{F} = \frac{b_l F_b}{b\ F_b} = \frac{b_l}{b} = p \tag{1}$$

$$FSR = \frac{F_{nd}}{F} = \frac{F^b_{nd}[T_{out}]b_l}{bF_b} = \frac{F^b_{nd}[T_{out}]}{F_b}p = \frac{F^b_{nd}[T_{out}]}{F_b}FLR \tag{2}$$

$F^b_{nd}[T_{out}]$ is computed as the average of the $F^b_{nd}[j;T_{out}]$ values:

$$F^b_{nd}[T_{out}] = \frac{1}{x}\sum_{j=1}^{x} F^b_{nd}[j;T_{out}] \tag{3}$$

$F^b_{nd}[j;T_{out}]$ is defined as the number of frames that could not be decoded when a burst of F_b frames is lost. j represents the position in the GoP of the first frame in the dropped burst.

A GoP has only one I-frame, and it is always the first frame ($j=1$), but the position of the P- and B-frames depends on the type of regular GoP, whether open or closed, and the transmission order of the frames. Therefore, the computation of $F^b_{nd}[j;T_{out}]$ will be different for open and closed GoPs.

In an open GoP, the last B-frames depend on the I-frame from the next GoP, as, for example, in *G12B2* or *IBBPBBPBBPBB*. In a closed GoP, there is no dependence with frames out of the GoP. A closed GoP ends with a P-frame, as, for example, in *G9B3* or *IBBBPBBBP*.

A GoP structure such as *IBBPBBPBBPBB* or *IBBBPBBBP* is shown in presentation order, i.e., the order in which the frames will be shown to the user. However, because B-frames require the previous and the following I- or P-frame to be processed, the coding/decoding order will be different. The coding order will be *IbbPBBPBBPBBiBB* for a *G12B2* GoP structure, where the frames in lower-case correspond to frames from the previous or the next GoP, showing that it is an open GoP. The coding order will be *IPBBBPBBB* for a *G9B3* GoP structure.

The transmission order usually corresponds to the coding/decoding order. This ordering decides which frames are lost when a burst is dropped. In this paper, the GoP structure will always be considered in coding/decoding order. Eq. (4) represents the position j of each frame in a GoP in transmission order for both open ($z=1$) and closed ($z=0$) GoPs, where $i=0\ldots G_P$ and $k=0\ldots y$.

$$j = 1+i(y+1)+k-(1-z)\left(y-\left\lfloor 1-\frac{i}{G_P}\right\rfloor (y-k)\right)$$
$$: \begin{cases} \text{I-frame} & \text{if } k=0 \text{ and } i=0 \\ \text{P-frame} & \text{if } k=0 \text{ and } i>0 \\ \text{B-frame} & \text{if } \{z=1,\ k>0 \text{ and } \forall i\} \\ & \quad \text{or } \{z=0,\ k>0 \text{ and } i>0\} \end{cases} \tag{4}$$

The loss of a burst started by an I-frame ($j=1$) will cause the whole GoP to not be decoded, and in the case of open GoPs, even the last B-frame block from the previous GoP will be useless. Eq. (5) provides $F^b_{nd}[j=1;T_{out}]$ for open and closed GoPs.

$$F^b_{nd}[j=1;T_{out}] = x\left\lceil \frac{T_{out}}{xT_{if}}\right\rceil + zy \tag{5}$$

The loss of a P-frame makes decoding all the following frames impossible in the GoP. Thus, the number of frames that could not be decoded when a P-frame started burst is lost will depend on which P-frame in the GoP started the burst.

If the ith P-frame of a GoP is lost, it will always cause $x-i(y+1)+y$ frames to not be decoded. If T_{out} is large enough to reach the I-frame from the next GoP, then all frames from the next GoP will not be decoded.

Eq. (6) provides $F^b_{nd}[j;T_{out}]$ for the case of a loss started by a P-frame for open and closed GoPs.

$$F^b_{nd}[j;T_{out}] = \left\lceil \frac{T_{out}+(j-1)T_{if}}{xT_{if}}\right\rceil x-(j-1)+zy \tag{6}$$

The loss of a B-frame does not affect any other frames. As the timer T_{out} gets larger, more frames will fall inside the lost burst whose first frame is a B-frame. Eventually, an I-frame (if the frames are from the last B-frame block) or a P-frame (otherwise) will be lost too. This scenario will cause other non-dropped frames to fail to be decoded as well. For the kth B-frame from a B-frame block, this scenario happens when $T_{out} > (y-k+1)T_{if}$.

Eq. (7) provides $F^b_{nd}[j;T_{out}]$ for the case of a loss started by a B-frame for open and closed GoPs, where $T_P = T_{out}-(y-k+1)T_{if}$ and $T_I = T_{out}+(j-1)T_{if}-xT_{if}$.

$$F^b_{nd}[j;T_{out}] = \begin{cases} \left\lceil \frac{T_{out}}{T_{if}}\right\rceil & \text{if } \frac{T_{out}}{T_{if}} \le (y-k+1) \\ y-k+1+F^b_{nd}\left[\frac{j+(2-z)y}{y+1};T_P\right] & \text{if } \frac{T_{out}}{T_{if}} > (y-k+1) \\ & \quad \text{and } j < x-y \\ y-k+1+F^b_{nd}[j=1;T_I] & \text{if } \frac{T_{out}}{T_{if}} > (y-k+1) \\ & \quad \text{and } j > x-y \end{cases} \tag{7}$$

Table 2
Average number of frames that cannot be decoded (F^b_{nd}) for different *GxBy* GoP structures (the GoP type is in parentheses).

F_b	*G12B2* (open)	*G12B10* (closed)	*G16B7* (open)	*G16B14* (closed)
1	3.83	2.75	3.25	2.81
2	7.33	4.50	6.37	4.62
3	10.50	6.16	9.37	6.37
4	11.50	7.75	12.25	8.06
5	12.50	9.25	15.00	9.68
6	13.50	10.66	17.62	11.25
7	14.50	12.00	20.12	12.75
8	15.50	13.25	22.50	14.18
9	16.50	14.41	23.50	15.56
10	17.50	15.50	24.50	16.87
11	18.50	16.50	25.50	18.12
12	19.50	17.50	26.50	19.31

Table 2 shows the analytical values of the average number of frames that cannot be decoded (F^b_{nd}) for different *GxBy* GoP structures.

Combining (5)–(7) through (3) and (4), the *FSR* is obtained as

$$FSR = \left(x\left\lceil \frac{T_{out}}{xT_{if}} \right\rceil + zy + \sum_{j=2}^{x} F^b_{nd}[j;T_{out}] \right) \frac{FLR}{xF_b} \tag{8}$$

4.2. Analysis of video playback interruptions (C_L and C_F derivation)

$C_b[T_{out}]$ is defined as the *average number of cuts per lost burst*. Video playback interruptions or cuts must be understood as experienced by the user; therefore, they must be measured from the presentation order of the videos, not the transmission order, where the bursty losses take place. The average number of cuts per frame $C_F = C_b[T_{out}]b_l/F$, where b_l is the number of lost bursts. The average video playback interruption length can be expressed (9) as the ratio of $F^b_{nd}[T_{out}]$ and $C_b[T_{out}]$:

$$C_L = \frac{F_{nd}}{F \cdot C_F} = \frac{F^b_{nd}[T_{out}]b_l}{F \cdot C_b[T_{out}]b_l/F} = \frac{F^b_{nd}[T_{out}]}{C_b[T_{out}]} \tag{9}$$

$C_b[j;T_{out}]$ is defined as the number of cuts produced when a burst of F_b frames is lost, where the first frame in the burst is the jth frame from the GoP. $C_b[T_{out}]$ is computed (10) as the average of $C_b[j;T_{out}]$:

$$C_b[T_{out}] = \frac{1}{x}\sum_{j=1}^{x} C_b[j;T_{out}] \tag{10}$$

The computation of $C_b[j;T_{out}]$ could be different for open and closed GoPs.

The loss of a burst started by an I-frame will cause the whole GoP to not be decoded, creating a single cut. In the case of an open GoP, the last B-frames from the previous GoP will be part of the same single cut when considered in presentation order.

The loss of a burst that is started by a P-frame makes decoding all the following frames impossible in the GoP. Because all these frames are consecutive, both in transmission and presentation order, only one cut is generated. If T_{out} is large enough to reach the I-frame from the next GoP, then all the frames from the next GoP will not be decoded. Again, all these frames are consecutive and result in only one cut.

The loss of a burst started by a B-frame and containing only B-frames creates only one cut. When the burst formation timer is large enough to reach an I- or P-frame, two cuts are generated. This scenario is due to an I- or P-frame, out of the burst, whose position in presentation order falls between lost frames. The threshold for this event is $T_{out} > (y+1-k)T_{if}$, where k is the position of the first frame in the lost burst relative to the B-frame block. For example, suppose a B-frame block has two frames ($k=1\ldots2$). The first frame of the block ($k=1$) needs a timer T_{out} greater than twice the T_{if} to reach the frame outside the B-frame block, i.e., $T_{out} > (2+1-1)T_{if}$. However, the second frame in the block only needs a timer T_{out} greater than T_{if}, $T_{out} > (2+1-2)T_{if}$.

Eq. (11) provides $C_b[j;T_{out}]$ of a B-frame for open and closed GoPs.

$$C_b[j;T_{out}] = \begin{cases} 2 & \text{if } k>0 \text{ and } T_{out} > (y+1-k)T_{if} \text{ and} \\ & \quad \{\text{open GoP or } \{\text{closed GoP and } i>0\}\} \\ 1 & \text{otherwise} \end{cases} \tag{11}$$

Combining (10) and (11), the average number of cuts per lost burst is obtained (12). The expression is valid for both open ($z=1$) and closed GoPs ($z=0$).

$$C_b[T_{out}] = \frac{1+G_P+\frac{G_B}{y}\sum_{k=1}^{y} C_b[j=2-z+k;T_{out}]}{x} = \frac{1+\left\lfloor \frac{x-1}{y+1} \right\rfloor + \frac{x-1-\left\lfloor \frac{x-1}{y+1} \right\rfloor}{y}\sum_{k=1}^{y} C_b[j=2-z+k;T_{out}]}{x} \tag{12}$$

$C_b[T_{out}]$ has a minimum value of 1 when $T_{out} \le T_{if}$, i.e., when only one frame is inside each burst.

$C_b[T_{out}]$ reaches the maximum value of $1+y/x \cdot \lceil (x-1)/(y+1) \rceil$ for all $T_{out} > yT_{if}$, i.e., when there are at least $y+1$ frames in each burst.

Combining (12) and (3), the average cut length C_L (9) is obtained. Table 3 shows this average cut length, measured in frames, for different *GxBy* GoP structures.

The average number of cuts per frame is computed as

$$C_F = \frac{C_b[T_{out}]b_l}{F} = \frac{C_b[T_{out}]bp}{bF_b} = \frac{C_b[T_{out}]}{F_b}p \tag{13}$$

Table 3
Average cut length C_L, measured in frames, for different *GxBy* GoP structures (the GoP type is in parentheses).

F_b	*G*12*B*2 (open)	*G*12*B*10 (closed)	*G*16*B*7 (open)	*G*16*B*14 (closed)
1	3.83	2.75	3.25	2.81
2	5.50	4.15	5.66	4.35
3	6.30	5.28	7.50	5.66
4	6.90	6.20	8.90	6.78
5	7.50	6.93	10.00	7.75
6	8.10	7.52	10.84	8.57
7	8.70	8.00	11.50	9.27
8	9.30	8.36	12.00	9.86
9	9.90	8.65	12.53	10.37
10	10.50	8.85	13.06	10.80
11	11.10	9.00	13.60	11.15
12	11.70	9.54	14.13	11.44

5. Network scenario with a multiplex of videos per burstifier

As in the previous scenario, a burst loss ratio p at the network is assumed. Bursts from different burstifiers do not interfere with each other, so a network with only one burstifier will be studied without loss of generality.

In this scenario, user video requests from a remote network are modelled using a Poisson arrival process. All the flows from the same ingress node that address the same egress node are aggregated into the same burst formation buffer with timer T_{out}. Videos may have

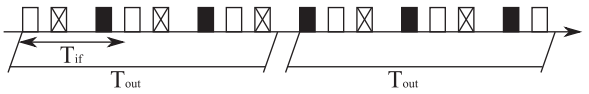

Fig. 3. All videos have F_b or F_b-1 frames inside each burst.

different inter-frame times T_{if} and *GxBy* GoP structures. The number of frames in a burst depends on the relationship between the T_{if} of each video and the timer T_{out}. For simplicity, we assume that all videos aggregated into the same burstifier have the same T_{if}, but the analytical model could be extended to the case of videos with different T_{if}. The average number of multiplexed video flows will be labelled as N.

Each burst could contain a different number of frames from each flow it aggregates. The video flow that initiates the burst formation timer will always aggregate $F_b = \lceil T_{out}/T_{if} \rceil$ frames. The rest of the flows in the burst behave as having a timer for the remaining value when its first frame arrives. The result is that some flows aggregate F_b frames while others aggregate only F_b-1 frames (see Fig. 3 as an example). Those flows that aggregate F_b-1 frames could be considered as having an equivalent timer of $T_{out}-T_{if}$. Because each burst could be initiated by a different video flow, a video flow will not always aggregate the same number of frames.

From Section 4, losses and video playback interruptions can be predicted for a video flow when all the bursts contain the same number of frames. In this new scenario, however, for some bursts the video flow will aggregate F_b of its frames, while for others it will contribute with F_b-1. To apply the previous results, we require the proportion of bursts from each type.

$P[F_b|k]$ is the probability that a video flow contributes to a burst with F_b frames when there are exactly k flows being aggregated in the burstifier. This event takes place when the flow's first frame arrives less than $T_{out}-\lfloor T_{out}/T_{if} \rfloor T_{if}$ time units from the one that starts the timer. Assuming a large number of concurrent flows, the time distance from the timer-starting frame to the arrival of the first frame from each flow can be approximated using a uniform distribution in the range $[0,T_{if}]$. Therefore, the probability of a video flow being below that margin in a burst is just $\alpha = T_{out}/T_{if} - \lfloor T_{out}/T_{if} \rfloor$.

Assuming that all the flows have equal probability of starting the timer, then $P[F_b|k] = 1/k + \alpha(k-1)/k$. This result takes into account the case when the flow starts the timer as well as the case when it does not start the timer but its first frame arrives close enough to the one doing it.

Due to the Poisson arrivals, the probability of k videos being multiplexed in a burst aggregator is $P[k] = e^{-N}N^k/k!$, the probability of being in state k of an $M/G/\infty$ system [26] with an average of N videos multiplexed. Because the timer only works when there is at least one video flow, the state probability of interest must be conditioned to the case of at least one flow in the system, as

$$P[k|k>0] = \frac{P[k]}{P[k>0]} = \frac{P[k]}{\sum_{i=1}^{\infty} P[i]} = \frac{\frac{N^k}{k!}e^{-N}}{\sum_{i=1}^{\infty}\frac{N^i}{i!}e^{-N}} = \frac{1}{1-e^{-N}}\frac{N^k}{k!}e^{-N} = \frac{1}{e^N-1}\frac{N^k}{k!} \quad (14)$$

The expression for the probability that a video flow contributes to a burst with F_b frames $P[F_b]$ is computed as

$$\begin{aligned} P[F_b] &= \sum_{i=1}^{\infty}(P[F_b|i]P[i|i>0]) \\ &= \sum_{i=1}^{\infty}\left(\left(\frac{1}{i}+\frac{i-1}{i}\alpha\right)\frac{1}{e^N-1}\frac{N^i}{i!}\right) \\ &= \frac{1}{e^N-1}\sum_{i=1}^{\infty}\left(\left(\alpha+\frac{1-\alpha}{i}\right)\frac{N^i}{i!}\right) \\ &= \frac{\alpha}{e^N-1}\sum_{i=1}^{\infty}\frac{N^i}{i!}+\frac{1-\alpha}{e^N-1}\sum_{i=1}^{\infty}\frac{N^i}{i!}\frac{1}{i} \\ &= \frac{\alpha}{e^N-1}(e^N-1)+\frac{1-\alpha}{e^N-1}(Ei(N)-\gamma-\mathrm{Log}(N)) \\ &= \alpha+\frac{1-\alpha}{e^N-1}(Ei(N)-\gamma-\mathrm{Log}(N)) \end{aligned} \quad (15)$$

In (15), $Ei(N) = -\int_{-N}^{\infty} e^{-t}/t\,dt$ is the Exponential Integral of N and γ is the Euler–Mascheroni constant [27]. $P[F_b]$ tends to α as the average number of simultaneous videos N grows.

Based on this result, the average number of frames from a flow inside the burst, $\overline{F_b}$, is calculated as

$$\overline{F_b} = P[F_b]F_b + (1-P[F_b])(F_b-1) = F_b - 1 + P[F_b] \quad (16)$$

In the following subsections, an algebraic and systematic method to compute *FLR*, *FSR*, C_L and C_F for the scenario with a multiplex of videos per burstifier is demonstrated.

5.1. Analysis of losses (FLR and FSR derivation)

Similar to the one video per burstifier scenario, the *FLR* for the scenario with multiplexation is just p.

To obtain the *FSR*, the number of dropped frames must be translated into the number of frames that could not be decoded. The bursts that contain F_b frames from a certain video will result in $F_{nd}^b[T_{out}]b_l P[F_b]$ non-decodable frames. Those bursts containing F_b-1 frames result in $F_{nd}^b[T_{out}-T_{if}]b_l(1-P[F_b])$ non-decodable frames. The *FSR* for the N average number of multiplexed video flows is finally computed in (17), where $F_{nd}^b[T]$ can be directly obtained from (3):

$$\begin{aligned} FSR &= \frac{F_{nd}}{F} \\ &= \frac{(F_{nd}^b[T_{out}]P[F_b]+F_{nd}^b[T_{out}-T_{if}]P[F_b-1])b_l N}{\overline{F_b}bN} \\ &= \frac{(F_{nd}^b[T_{out}]P[F_b]+F_{nd}^b[T_{out}-T_{if}](1-P[F_b]))b_l N}{\overline{F_b}bN} \\ &= \frac{F_{nd}^b[T_{out}]P[F_b]+F_{nd}^b[T_{out}-T_{if}](1-P[F_b])}{\overline{F_b}}p \\ &= \frac{F_{nd}^b[T_{out}]P[F_b]+F_{nd}^b[T_{out}-T_{if}](1-P[F_b])}{\overline{F_b}}FLR \end{aligned} \quad (17)$$

The *FSR* in this scenario depends on the average number of multiplexed flows through the $P[F_b]$ term. As the multiplexing level N is increased, $P[F_b]$ tends to α and *FSR* becomes stable.

5.2. Analysis of video playback interruptions (C_L and C_F derivation)

Using the same procedure as for (17), the average video playback interruption length can be obtained from F^b_{nd} using (9), where $F^b_{nd}[T]$ and $C_b[T]$ are already solved (i.e., in (3) and (12)):

$$C_L = \frac{F_{nd}}{F \cdot C_F} = \frac{(F^b_{nd}[T_{out}]P[F_b] + F^b_{nd}[T_{out}-T_{if}](1-P[F_b]))b_l N}{(C_b[T_{out}]P[F_b] + C_b[T_{out}-T_{if}](1-P[F_b]))b_l N} = \frac{F^b_{nd}[T_{out}]P[F_b] + F^b_{nd}[T_{out}-T_{if}](1-P[F_b])}{C_b[T_{out}]P[F_b] + C_b[T_{out}-T_{if}](1-P[F_b])} \quad (18)$$

The average number of cuts per frame for videos in presentation order is computed as

$$C_F = \frac{(C_b[T_{out}]P[F_b] + C_b[T_{out}-T_{if}](1-P[F_b]))b_l N}{F} = \frac{(C_b[T_{out}]P[F_b] + C_b[T_{out}-T_{if}](1-P[F_b]))bpN}{\overline{F_b}bN} = \frac{C_b[T_{out}]P[F_b] + C_b[T_{out}-T_{if}](1-P[F_b])}{\overline{F_b}}p \quad (19)$$

The average number of cuts per frame is inversely proportional to the average number of frames per burst $\overline{F_b}$.

6. Validation through simulation

In Sections 4 and 5, we have presented the analytical models for computing *FLR*, *FSR* and C_L in a timer-based OBS network. In this section, we present simulation results that validate the applicability of the analytical models. We developed a modified version of our OBS simulator [28]. Statistical results are obtained at 95% confidence level, but most confidence intervals are too small to be noticed in the figures. Video traces from [29,30] with same and different bit-rate, GoP and inter-frame time have been used: *The Lord of the Rings III* (*LOTRIII*), *The Matrix*, *Tokyo Olympics* and *Star Wars IV*. The results do not exhibit significant differences and therefore this paper mostly shows figures with the trace *LOTRIII* (*G*12*B*2 GoP structure, $T_{if} = 40$ ms, average bit rate of 714 kbps at 192 min) and *Star Wars IV* (*G*16*B*7 GoP structure, $T_{if} = 40$ ms, average bit rate of 3.143 Mbps at 36 min).

The analytical models require that the losses in the OBS network could be modelled by an independent burst loss ratio p. First, we assume an environment where the burst loss ratio is an independent rate p and we validate the analytical models for one video per burstifier (Section 6.1) and for a multiplex of videos per burstifier (Section 6.2). This environment is modelled using a black box network scenario with an i.i.d. burst loss ratio p. In this environment, different network scenarios were evaluated using $p = [10^{-4}, 10^{-1}]$ and a minimum of 2 million bursts in the stationary state. Previous papers have studied the effect of this range of loss ratios in TCP traffic [31].

Afterwards, a full network topology is used (Section 6.3), where the burst loss ratio is the result of output port contention on core nodes. In this environment, burst losses can present correlation and it cannot be asserted that the analytical models are valid. However, as the results will show, the analytical models stay accurate for low to medium load conditions.

6.1. One video per burstifier scenario

Fig. 4 shows the simulation results of the *FLR* and *FSR* versus the timer value T_{out} (lower x-axis) and the number of frames per burst (upper x-axis). The x-axis range is large enough to cover the GoP duration to show the effects of its structure. The longest timer duration considered in the figure is 500 ms, still reasonable for live multicasting and comparable to coding delays. The results show (Fig. 4) that *FLR* is equal to the burst loss ratio p and is independent of the timer value. The *FSR* tends to *FLR* as the timer value grows.

Fig. 5 shows that the estimation of *FSR* from (8) closely follows the simulation results. The theoretical result provides a slight overestimation of the *FSR* because the situations with more than one burst loss containing frames from the same GoP have been ignored. For example, two lost bursts could contain frames from the same GoP. In this situation, there is a high probability that some frames

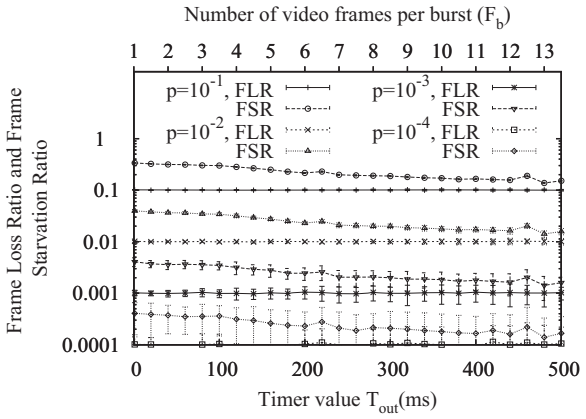

Fig. 4. Losses (*FLR* and *FSR*) for the *LOTRIII* trace (*G*12*B*2 GoP structure).

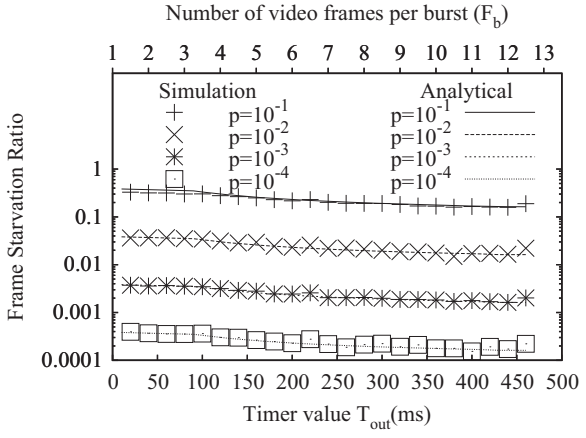

Fig. 5. Analytical model of *FSR* becoming accurate for low burst loss ratios for the *LOTRIII* trace file (*G*12*B*2 GoP structure).

dropped due to the second loss were the same as some not decoded due to those frames dropped in the first one. Therefore, the real *FSR* will be smaller than the computed one. However, as the burst loss ratio p becomes smaller, the probability of more than one loss per GoP becomes negligible and the *FSR* estimation becomes more accurate. Even for high loss rates such as $p = 10^{-1}$, the approximation looks reasonable.

The results were verified with movie files that use different GoPs and frame-rates. Fig. 6 shows the *FSR* for a movie trace file that uses *G12B2* (*LOTRIII*) and for one that uses *G16B7* (*Star Wars IV*). The analytical results match the simulations well in both situations. Fig. 6 shows only the results for several discrete values of T_{out}. Because the determinant factor is the number of frames that fall inside each burst, all the timer values that result in the same F_b obtain the same *FSR*. For a different GoP structure, different *FSR* values are obtained for the same F_b. The influence of the GoP structure on the results will be studied in more detail in Section 7.

Fig. 7 shows the average cut lengths C_L, measured in frames, from the simulations using a trace file with a *G12B2* structure and computed using (9). The approximation matches the simulation results better as the burst loss ratio p becomes smaller. The average duration of the interruption measured in time units is directly obtained by multiplying the value from (9) by the inter-frame time T_{if}.

6.2. Multiplex of videos per burstifier scenario

The computation of *FSR* for a video in the multiplexed scenario (17) is the result of combining the effects of fitting F_b or F_b-1 frames in a burst. The probability of a video fitting a certain number of frames in a burst depends on the distance between the frame that starts the aggregation process and the first frame from the measured flow (Fig. 3). It was assumed that this random variable would follow a uniform distribution if the number of concurrent flows was large enough.

Fig. 8 shows the experimental cumulative distribution function (CDF) for this variable in scenarios with a different average number of video flows being multiplexed.

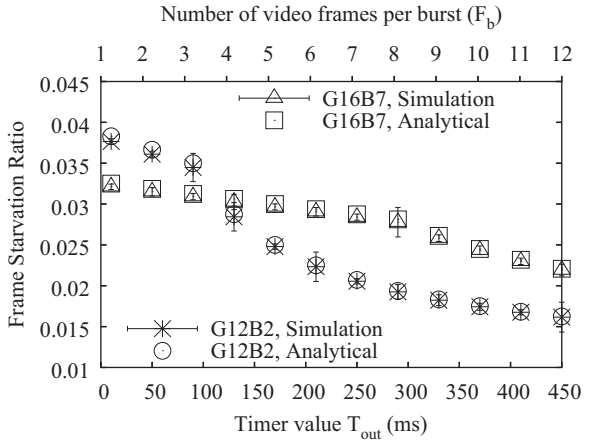

Fig. 6. Analytical model of *FSR* becoming accurate for different *GxBy* GoP structures and $p = 10^{-2}$.

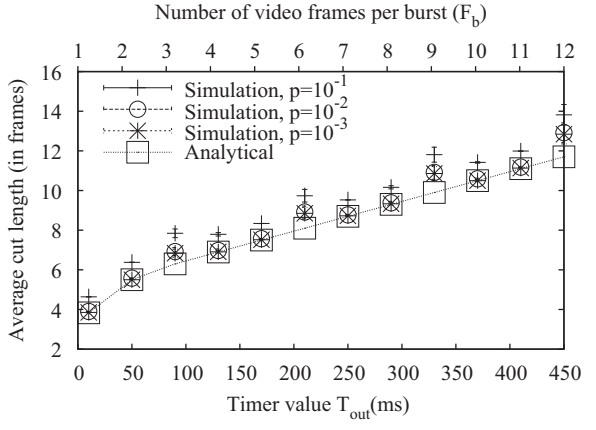

Fig. 7. Analytical model of C_L improved for low burst loss ratios. *LOTRIII* trace file (*G12B2* GoP structure).

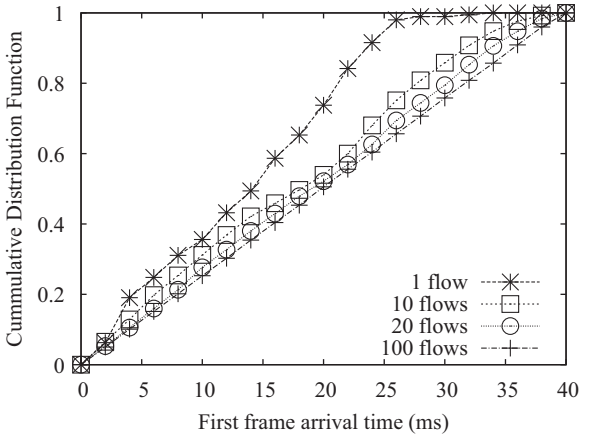

Fig. 8. Convergence to uniform distribution for the first frame arrival time by increasing the multiplexing level for the *LOTRIII* trace file (*G12B2* GoP structure) and $T_{out} = 95$ ms.

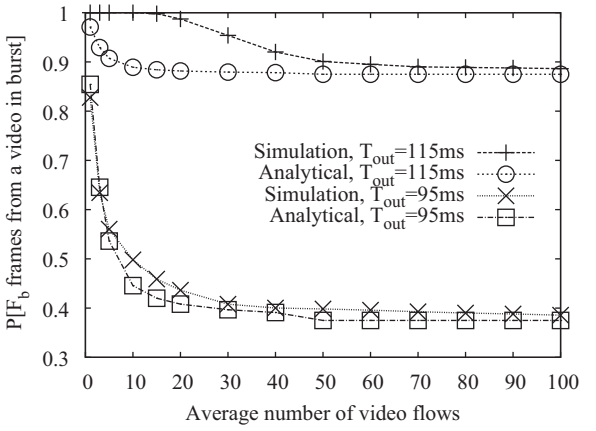

Fig. 9. Convergence of $P[F_b]$ to analytical results by increasing the multiplexing level for the *LOTRIII* trace file (*G12B2* GoP structure).

Because the behaviour is similar for any timer value, only the simulation results for $T_{out} = 95$ ms are presented. As the average number of video flows grows ($N \to \infty$), the CDF approximates a uniform distribution. Therefore, the multiplexing level plays an important role in the validity of the results for *FSR*, and better approximations are expected as the number of concurrent flows grows.

Fig. 9 shows the values of $P[F_b]$ (analytical and simulation) versus the multiplexing level using two different timer values. As expected, the model gets closer to the simulation results as the number of concurrent flows grows. Additionally, as predicted, $P[F_b] \to \alpha$, which only depends on the timer T_{out} and the inter-frame arrival time T_{if}.

For bursts containing frames from only one video, (3) provides the relationship between bursty losses in the network and the proportion of non-decoded frames at the receiver. Eq. (17) extends the previous result to the general scenario with Poisson arrivals of video requests and multiple video flows per burstifier.

Fig. 10 shows that the estimation of *FSR* from (17) closely follows the simulation results when the average number of video flows is high enough. As the burst loss ratio p in the network gets lower, the analytical approximation improves. This effect is reasonable, as (17) depends on (3), and the latter is only valid for low burst loss ratios. Even for high loss rates such as $p = 10^{-1}$, the approximation appears reasonable.

Fig. 11 presents the dependent part of (17), namely *FSR*/*FLR*. The results shown are for timers $T_{out} = 95$ ms and $T_{out} = 115$ ms, and no significant deviation has been observed for other timer values. The solid line represents the analytical result. As the average number of video flows grows, the simulation values stabilize. The lower the loss rate, the closer the analytical result is to the simulated one.

The results were checked with movie files that use different GoPs and frame-rates. Fig. 12 shows the *FSR* for a movie trace file that uses *G*12*B*2 (*LOTRIII*) and for one that uses *G*16*B*7 (*Star Wars IV*). The analytical results match the simulations well in both situations.

Fig. 13 shows the average cut lengths, measured in frames, from the simulations with the *LOTRIII* trace file and

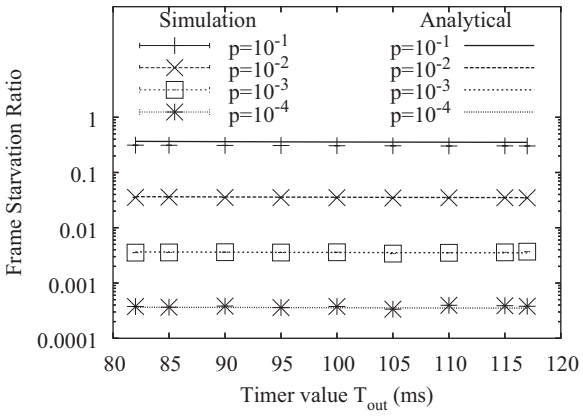

Fig. 10. Analytical model of *FSR* becoming accurate for low burst loss ratios for the *LOTRIII* trace file (*G*12*B*2 GoP structure) and $N = 100$.

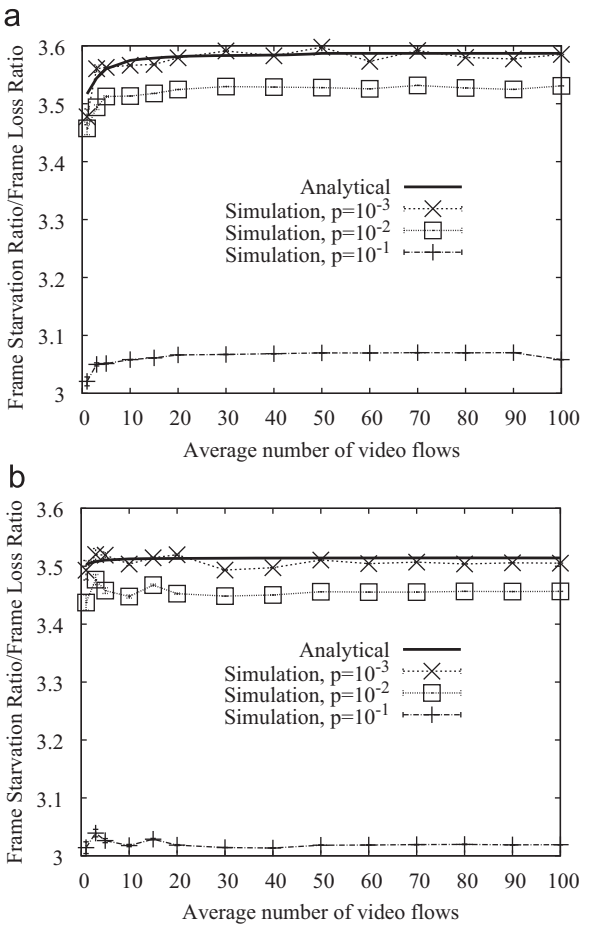

Fig. 11. Analytical model of *FSR* becoming accurate for low burst loss ratios and high N for the *LOTRIII* trace file (*G*12*B*2 GoP structure). (a) $T_{out} = 95$ ms. (b) $T_{out} = 115$ ms.

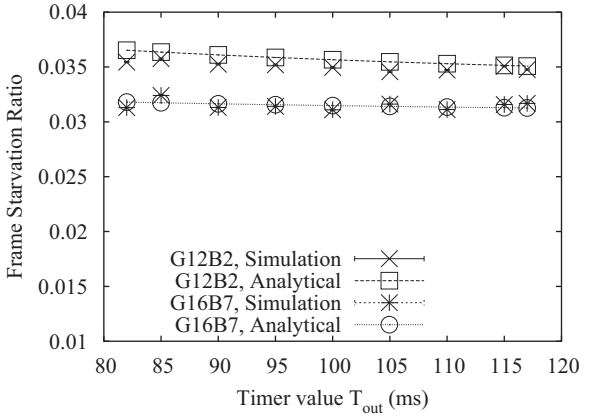

Fig. 12. Analytical model of *FSR* becoming accurate for different *GxBy* GoP structures and $p = 10^{-2}$.

computed using (18). As in the case of one video per burstifier, the approximation matches the simulation results better as the burst loss ratio p gets smaller. For large p, there is a higher probability that two losses

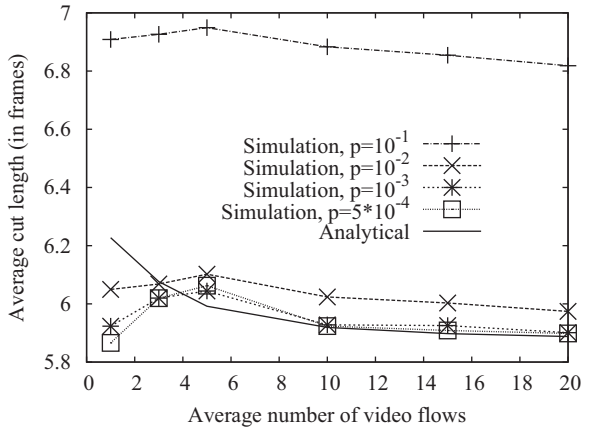

Fig. 13. Analytical model of C_L becoming accurate for low burst loss ratios for the *LOTRIII* trace file (*G*12*B*2 GoP structure) and $T_{out} = 95$ ms.

are in the same GoP or adjacent GoPs with the result of a larger video cut. Only results around $p = 10^{-1}$ are significantly separated from analytical ones. As shown before, the analytical result becomes closer to the simulation values as the number of multiplexed flows grows due to the assumptions taken for the analytical derivation. Finally, the timer value does not significantly influence the quality of the approximation, as was verified for the scenario of one video per burstifier.

6.3. NSFNet network scenario

In this subsection, we employ the NSFNet topology with 14 core nodes (see Fig. 14) to validate the analytical models over a full network topology. In this network, the burst loss ratio is the result of output port contention on core nodes. Therefore, burst losses can present correlation which invalidates one of the model hypotheses. We intend to check whether the results still hold.

All the data links between nodes are bidirectional with 12 data wavelengths at 1 Gbps. Each core node has one edge node that introduces traffic to the network. The edge nodes generate two types of traffic, namely video traffic and Internet data traffic.

Each edge node sends a video flow to the other 13 edge nodes. In an interactive scenario, the users can select a movie and start a new video flow at any time, although the movie could be the same, the video flows will not be synchronized. This is modelled selecting a different random starting time between $[0,T_{if}]$ and a different random starting frame for each video.

The Internet data traffic is modelled [32] as a burst process with an exponential interarrival time distribution of mean T_{out}/m and a Gaussian size distribution of mean $T_{out}\mu$ and variance $T_{out}^{2H}\sigma^2$. The traffic from each edge node is evenly distributed to the other 13 edge nodes. As suggested in [32], the traffic parameters inferred from the *Bellcore traces* are used (coefficient of variation $c_v^2 = \sigma^2/\mu^2 = 0.1$ and Hurst parameter $H = 0.78$) and m (number of FECs on the edge node) is fixed to 10. The key parameter of the average burst size, μ, is set to different values in order to obtain the different load

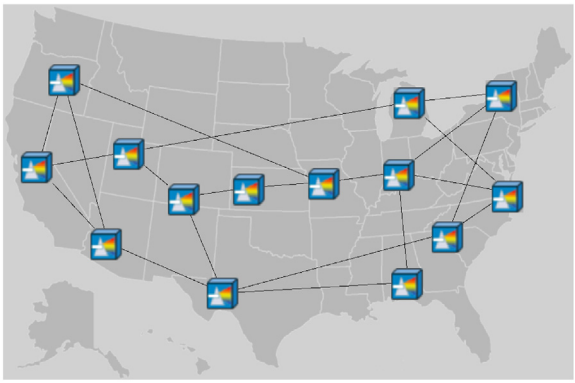

Fig. 14. NSFNet core network with video and interference traffic from all to all.

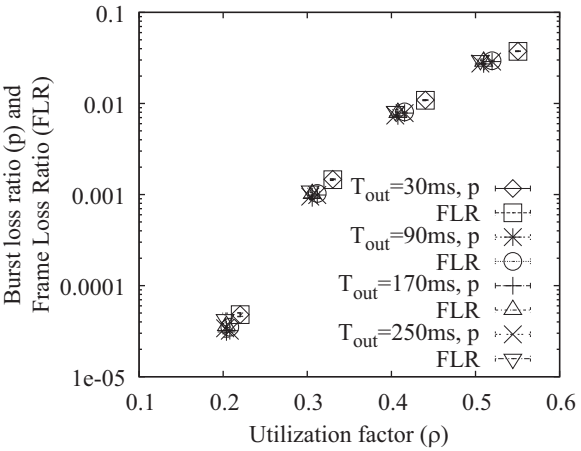

Fig. 15. Validation of analytical model of *FLR* for the *Star Wars IV* trace file (*G*16*B*7 GoP structure) in the NSFNet scenario.

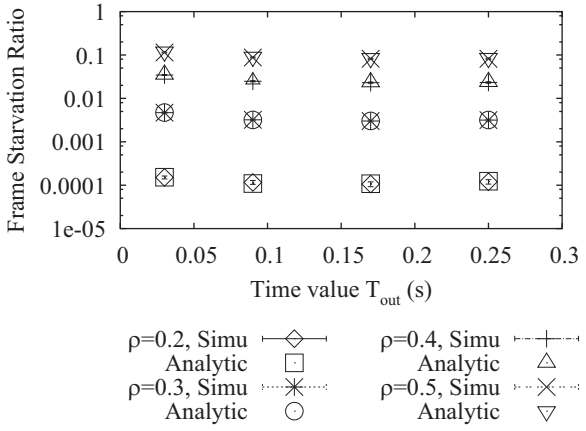

Fig. 16. Validation of analytical model of *FSR* for the *Star Wars IV* trace file (*G*16*B*7 GoP structure) in the NSFNet scenario.

conditions $\rho = [0.2,0.5]$ at the output port on each edge node, i.e., each edge node sends on average an Internet data traffic of $\rho * 12/13 * 1$ Gbps to the other 13 edge nodes.

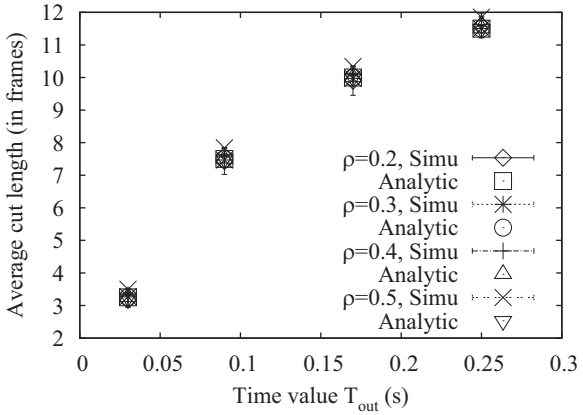

Fig. 17. Validation of analytical model of C_L for the *Star Wars IV* trace file (*G*16*B*7 GoP structure) in the NSFNet scenario.

Fig. 15 shows the simulation results for the burst loss ratio p and *FLR* in the network versus different load conditions. The results show that *FLR* is equal to the burst loss ratio p and that it is independent of the timer value.

Fig. 16 shows that the estimation of *FSR* from the analytical model (8) closely follows the simulation results for the trace *Star Wars IV* with *G*16*B*7 structure.

Fig. 17 shows the average cut lengths C_L, measured in frames, from the simulations using the trace *Star Wars IV* with *G*16*B*7 structure and those computed using (13).

With these simulation results, we can conclude that the analytical model for one video per burstifier can be applied over a full network topology.

In the case of several multiplexed videos per burstifier, the results in a whole network scenario are similar. The only difference in the model is based on the uniform distribution of the arrivals inside a burst, but this phenomenon takes place on the edge of the network and therefore its validity will not change on a network scale scenario.

7. Influence of GoP structure on results

Section 6 showed that the analytical equations presented in Sections 4 and 5 follow the simulation results well, obtaining better approximations as the burst loss ratio p becomes smaller. All these equations depend on the *GxBy* GoP structure, as observed in Fig. 6. Thus, the influence of the *GxBy* GoP structure on the *FSR* and C_L must be studied in greater detail.

7.1. One video per burstifier scenario

Fig. 18 compares the *FSR* as obtained from (8) for different GoP structures. For every GoP structure, as the timer value T_{out} grows, more frames fall into each burst; hence, when a burst is lost, fewer frames outside the GoP are affected. Therefore, the larger the T_{out}, the better. However, this timer increase will result in larger bursts and larger video playback interruptions.

Fig. 18 shows that when using the same timer value T_{out}, better results (lower losses) can be obtained by coding

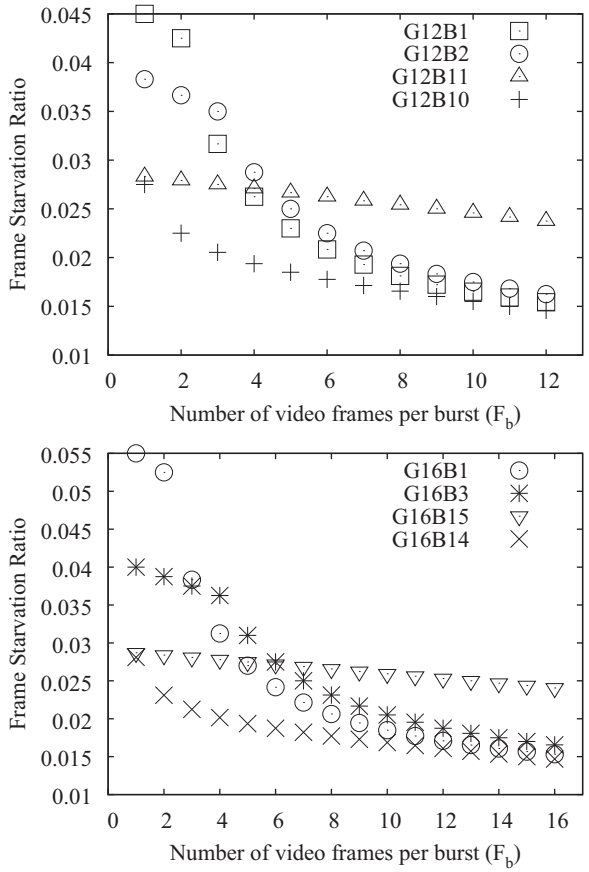

Fig. 18. Analytical model of *FSR* for different *GxBy* GoP structures and $p = 10^{-2}$.

the movie with the proper GoP structure. For example, a target *FSR* quality value of 0.025 (2.5% of frames not decoded) in a network that drops 1% of the bursts cannot be achieved with a timer value T_{out} of 250 ms and a GoP structure *G*16*B*7, but it is obtainable with a *G*12*B*2 structure. It must be noted that the GoP structure can be changed and the same flow rate can be kept by adjusting the coding quality.

As shown in Fig. 18, the GoP structure has a non-linear effect on the *FSR*. A similar behaviour is observed for C_L. Fig. 19 compares C_L as obtained from (9) for different GoP structures. For any GoP structure, as the timer value T_{out} grows, more frames fall into each burst; hence, when a burst is lost, the video playback interruption lasts longer. For larger GoP structures, video cuts become longer, even reaching or surpassing the GoP length. Therefore, the shorter the GoP, the better. However, this GoP length reduction will result in larger bit rates because the number of I-frames will increase.

7.2. Multiplex of videos per burstifier scenario

Fig. 20 presents the analytical quotient of *FSR* over *FLR* for different GoP structures. To facilitate visual inspection, the values are computed as an increment of the value for

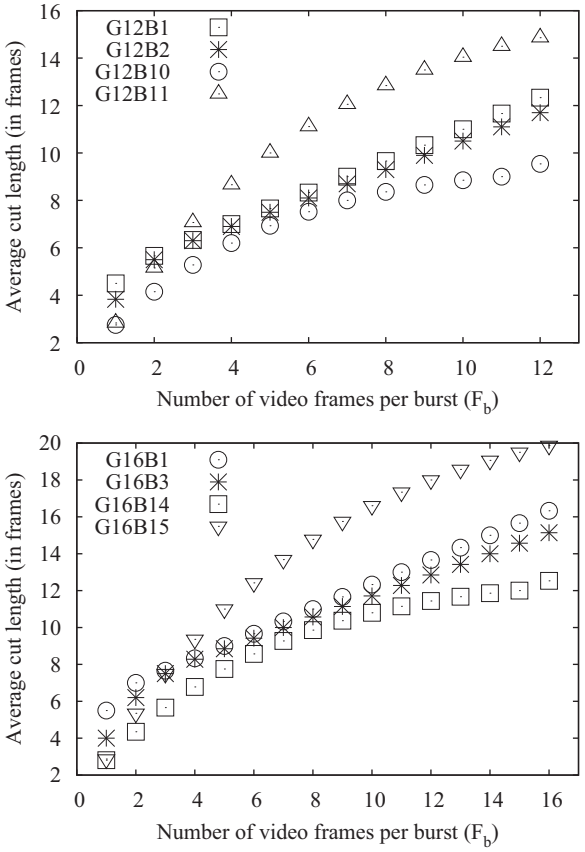

Fig. 19. Analytical model of C_L for different *GxBy* GoP structures.

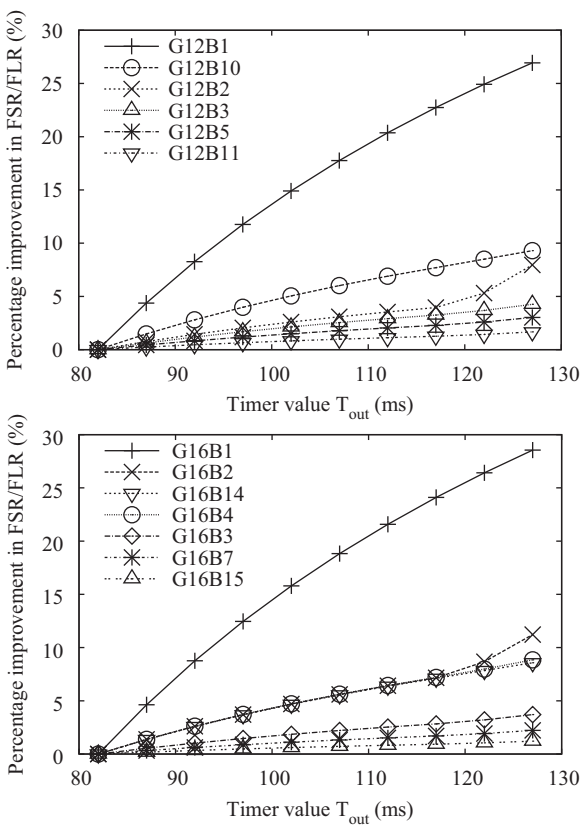

Fig. 20. Loss reduction with increasing timer value T_{out} depends on the *GxBy* GoP structure.

timer $T_{out} = 82$ ms. The figure shows that as the timer value increases (*x*-axis), there is a reduction of the *FSR* of as much as a 30% with respect to the value at $T_{out} = 82$ ms. However, this improvement is obtained only for some *GxBy* GoP structures, e.g., *G*12*B*1 or *G*16*B*1. For other GoP structures, e.g., *G*12*B*5, this improvement is insignificant. Therefore, selection of the GoP structure has a strong impact on performance, measured as the number of frames that could not be decoded.

Fig. 21 presents, as in Fig. 20, the analytical quotient of *FSR* over *FLR* for different GoP structures, but now, the absolute values are plotted. Obviously, the figure shows that as the timer value increases, there is a reduction of the *FSR* for some GoP structures. However, the figure also shows that the *FSR*/*FLR* absolute values of each GoP structure start from different levels. Thus, some GoP structures may continue to have a worse *FSR* as the timer value increases, although they greatly improve it, e.g., *G*16*B*1. Other GoP structures may improve just enough as the timer value increases to outperform other GoP structures. For example, *G*12*B*1 outperforms *G*12*B*2 for timers greater than 105 ms, and *G*16*B*4 outperforms *G*16*B*15 for timers greater than 117 ms.

Therefore, the selection of the *GxBy* GoP structure not only has a strong impact on performance, but it also has a strong impact on the selection of the timer value T_{out} for the aggregation process.

8. Traffic Engineering for the burstification process

From a video provider or a network manager's perspective, it is necessary to maintain a minimum QoE as the number of transported flows varies. The aggregation timer T_{out} is a network parameter that critically influences the QoE in an OBS network scenario. The network manager must decide the timer value for existing and newly arriving flows to maintain a given QoE. The GoP structure is an external parameter that also influences the performance on losses and cut lengths. We assume that the GoP structure is imposed due to a video compression quality objective. However, as stated in Section 7, the GoP structure could have a serious impact on network performance results and, if possible, it should also be tuned.

The QoE metric presented in this paper is the set formed by the number of non-decoded frames (F_{nd}), the average cut length (C_L) and the total number of cuts. These three parameters are related, and by establishing two of them, the value of the third is also set. The best QoE system parameters for Traffic Engineering will be ones that do not depend on the video length (measured in time or in number of frames), so the number of non-decoded frames and the total number of cuts are not appropriate. The average number of cuts per frame C_F (or per time C_F/T_{if}) does not present such a problem, and it is directly related to the total number of cuts via the total number of frames

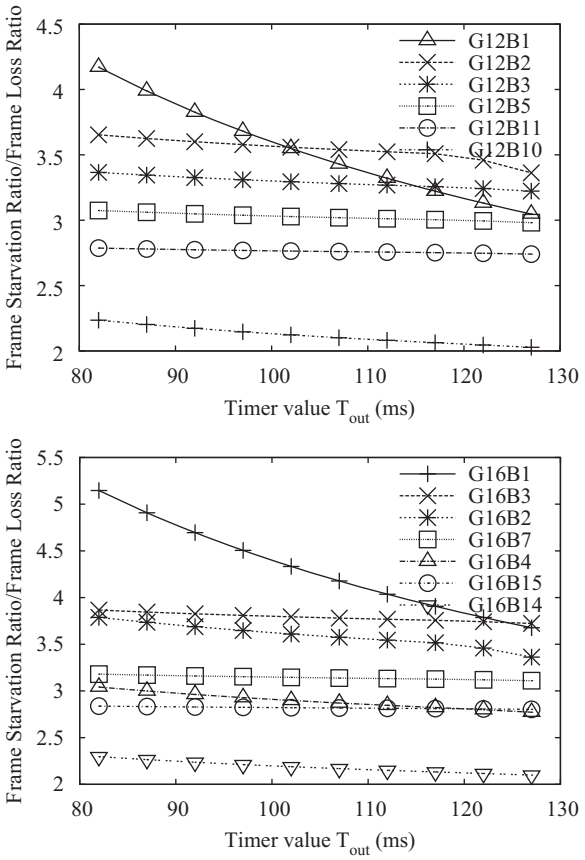

Fig. 21. Loss reduction with increasing timer value T_{out} depends on the *GxBy* GoP structure.

in the video. The second QoE system parameter will be the average cut length C_L.

The Traffic Engineering objective will be to determine a suitable range of timer T_{out}, for a given *GxBy* GoP structure and burst loss ratio p in the network, so that the average cut length C_L is less than or equal to the maximum value δ, and the average number of cuts per frame C_F is less than or equal to the maximum value β.

Looking at the equations for the scenario of one video per burstifier, (9) and (13), it can be observed that they do not depend on the exact value of T_{out} but on the number of times that T_{out} is greater than T_{if}. Therefore, for any value of T_{out} inside each range $(jT_{if},(j+1)T_{if}]$, the equations have the same result, with j being an integer value equal to or greater than 0. Thus, the problem is reduced to finding the ranges $(jT_{if},(j+1)T_{if}]$ where each constraint fails.

Timer values greater than the GoP duration are not reasonable for live multicasting, so j will be bounded from 0 to $x-1$. Because any value within each range $(jT_{if},(j+1)T_{if}]$ is equally valid, the Traffic Engineering procedure will use the highest value in the range $(j+1)T_{if}$.

The proposed procedure to find the range of timer T_{out} for the desired QoE is:

Step 1. Try to find the maximum timer value, T_{out}^{max}, that satisfies the average cut length constraint.
for $j=x-1$ to 0 **do**
Compute $C_L[(j+1)T_{if}]$ using (9)
if $C_L[(j+1)T_{if}]\leq\delta$ **then**
return $T_{out}^{max}=(j+1)T_{if}$
end if
end for
print "QoE requirements cannot be met"
return false

Step 2. Try to find the minimum timer value, T_{out}^{min}, that satisfies the average number of cuts per frame constraint and is smaller than T_{out}^{max}.
for $j=0$ to $x-1$ **do**
Compute $C_F[(j+1)T_{if}]$ using (13)
if $C_F[(j+1)T_{if}]\leq\beta$ **then**
if $jT_{if}\geq T_{out}^{max}$ **then**
print "QoE requirements cannot be met"
return false
else
return $T_{out}^{min}=jT_{if}$
end if
end if
end for
print "T_{out}^{min} has to be larger than a GoP"
print "QoE requirements cannot be met"
return false

If T_{out}^{min} and T_{out}^{max} can be computed, then the valid range of timer T_{out} for the desired QoE is $(T_{out}^{min},T_{out}^{max}]$.

In an OBS network, the video flows can be multiplexed in the same burstifier if they share the same destination edge node. A newly arriving flow at an OBS ingress node can be multiplexed in a burstifier if the destination edge node matches and if the timer being used falls inside the range $(T_{out}^{min},T_{out}^{max}]$. The timer must be chosen carefully in the first instance so that the QoE requirements of the videos in the burstifier are met as the number of multiplexed videos grows.

Looking at the equations for the scenario with a multiplex of videos per burstifier, (18) and (19), it can be observed that as N grows, the average cut length C_L decreases and the number of cuts per frame C_F grows. Thus, for a given timer T_{out} that satisfies the QoE requirements, as the number of videos in the burstifier grows, the average cut length continues to comply with the constraint, but the average number of cuts per frame constraint fails to be met. This constraint imposes the minimum timer value of the range T_{out}^{min}. Therefore, the minimum value of T_{out} for the multiplexation of video flows, $T_{out}^{min,mul}$, has to be computed for the worst case of N, i.e., for $N\rightarrow\infty$, if we intend to allow further multiplexation into this burstifier. The procedure is

$$F_b=\lceil T_{out}^{min}/T_{if}\rceil+1$$

$$P[F_b]=\frac{C_b[T_{out}^{min}]p-(F_b-1)\beta}{(C_b[T_{out}^{min}]-C_b[T_{out}^{min}+T_{if}])p+\beta}\quad \textbf{from } (19)$$

$$\alpha\approx P[F_b]\quad \textbf{from } (15) \textbf{ and } N\rightarrow\infty$$

$$T_{out}^{min,mul}=\alpha T_{if}+(\lfloor T_{out}^{min}/T_{if}\rfloor+1)T_{if}$$

Thus, the valid range of timer T_{out} for the desired QoE for any level of multiplexation is $[T_{out}^{min,mul},T_{out}^{max}]$. Selecting any timer value inside this interval, we ensure that the

QoE constraints are met even if new flows are multiplexed into this buffer.

As more video flows are aggregated in the same burstifier, the average size of the bursts, $b_s = N\overline{F_b}T_{if}V_R$, grows, where V_R is the average bitrate from the videos. As the burst size grows, more high speed electronic memory is needed, increasing the hardware costs. We assume that the amount of electronic memory available is bounded to limit these hardware costs, so a maximum burst size, b_s^{max}, has to be imposed to more efficiently exploit this memory. Then, the number of multiplexed video flows has an upper limit N_{max}:

$$N_{max} = \left\lfloor \frac{b_s^{max}}{\overline{F_b}T_{if}V_R} \right\rfloor = \left\lfloor \frac{b_s^{max}}{(F_b - 1 + P[F_b])T_{if}V_R} \right\rfloor \quad (20)$$

The maximum number of video flows allowed to be multiplexed on a burstifier is inversely proportional to the selected timer T_{out}. If a T_{out} near the $T_{out}^{min,mul}$ is selected, more videos will be multiplexed than if a T_{out} near the T_{out}^{max} is selected. Therefore, more videos will be affected by the loss of a burst generated with a T_{out} near the $T_{out}^{min,mul}$.

When a new video arrives, it is possible that more than one burstifier will be valid for it, i.e., more than one burstifier can have a T_{out} inside the $[T_{out}^{min,mul}, T_{out}^{max}]$ range for the new video and can have less than N_{max} videos multiplexed. The new video should be multiplexed in the burstifier with less videos, so the burst sizes and the number of videos affected by a burst loss remain balanced between burstifiers.

In summary, when a new video arrives, the complete Traffic Engineering procedure is:

Step (a) Compute the valid timer range $[T_{out}^{min,mul}, T_{out}^{max}]$ for the desired QoE level for any multiplexation level.

Step (b) Look for and select the burstifier with:
- the same destination;
- timer T_{out} inside that range;
- less than N_{max} number of videos in it.

Step (c) If there is more than one burstifier, select the one with less videos multiplexed.

Step (d) If there is none, make a new burstifier with a timer T_{out} inside the range $[T_{out}^{min,mul}, T_{out}^{max}]$. If large multiplexation is wanted, select a timer T_{out} near $T_{out}^{min,mul}$; if not, select one near T_{out}^{max}.

9. Conclusions

This paper has shown how to analytically compute the losses (Frame Loss Ratio and Frame Starvation Ratio) and video playback interruptions (average cut length and average number of cuts per frame) for the case of a network in which losses take place in bursts, as in the case of an OBS network. The results are sufficiently general to apply to any other technology with bursty losses (e.g., some DVB-H networks) or any technology that aggregates traffic into larger packets (e.g., some Optical Packet Switching architectures).

The analytical results were verified against simulations for different GoP structures, timers and number of video flows multiplexed in the same burstifier. A strong dependency between the video GoP structure and the timer value in the traffic aggregation process was revealed.

A Traffic Engineering methodology for the burstification process was presented. This methodology allows the network designer to select the timer value to maintain a given QoE level when a video flow arrives. The process takes into account whether the video can be multiplexed on a burstifier with other videos or if a new burstifier is needed.

Acknowledgements

This work was supported by the Spanish Ministry of Science and Innovation through the research project INSTINCT (TEC-2010-21178-C02-01). The authors would also like to thank the Spanish thematic network FIERRO (TEC2010-12250-E).

References

[1] Cisco Visual Networking Index, Forecast and Methodology, 2010–2015, Technical Report, Cisco Systems Inc., June 2011. URL ⟨http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf⟩.

[2] J. Hernandez, P. Reviriego, J. Garcia-Dorado, V. Lopez, D. Larrabeiti, J. Aracil, Performance evaluation and design of polymorphous OBS networks with guaranteed TDM services, Journal of Lightwave Technology 27 (13) (2009) 2495–2505.

[3] J. Hernandez, J. Aracil, V. Lopez, J. Garcia-Dorado, L. de Pedro, Performance analysis of asynchronous best-effort traffic coexisting with TDM reservations in polymorphous OBS networks, Photonic Network Communications 17 (2009) 93–103.

[4] H.L. Vu, A. Zalesky, E. Wong, Z. Rosberg, S. Bilgrami, M. Zukerman, R. Tucker, Scalable performance evaluation of a hybrid optical switch, Journal of Lightwave Technology 23 (10) (2005) 2961–2973.

[5] K. Vlachos, K. Ramantas, A non-competing hybrid optical burst switch architecture for QoS differentiation, Optical Switching and Networking 5 (4) (2008) 177–187.

[6] C. Xin, C. Qiao, Y. Ye, S. Dixit, A hybrid optical switching approach, in: Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE, vol. 7, 2003, pp. 3808–3812.

[7] C. Qiao, M. Yoo, Optical burst switching (OBS)—a new paradigm for an optical Internet, Journal of High-Speed Networks 8 (1) (1999) 69–84.

[8] X. Yu, J. Li, X. Cao, Y. Chen, C. Qiao, Traffic statistics and performance evaluation in optical burst switched networks, Journal of Lightwave Technology 22 (12) (2004) 2722–2738.

[9] Moving Picture Experts Group, The Moving Picture Experts Group (MPEG) Home Page, May 2012. URL ⟨http://mpeg.chiariglione.org/⟩.

[10] P. Seeling, M. Reisslein, Evaluating multimedia networking mechanisms using video traces, IEEE Potentials 24 (4) (2005) 21–25.

[11] O.A. Lotfallah, M. Reisslein, S. Panchanathan, A framework for advanced video traces: evaluating visual quality for video transmission over lossy networks, EURASIP Journal on Applied Signal Processing, 1 (2006) p. 263. http://dx.doi.org/10.1155/ASP/2006/42083.

[12] ITU-T Study Group 12, Recommendation P.10/G.100 (2006) Amendment 2: Vocabulary for Performance and Quality of Service—New definitions for Inclusion in Recommendation ITU-T P.10/G.100, ITU-T, 2008.

[13] M. Fiedler, T. Hossfeld, P. Tran-Gia, A generic quantitative relationship between quality of experience and quality of service, IEEE Network 24 (2) (2010) 36–41.

[14] J. She, F. Hou, B. Shihada, P.-H. Ho, MAC-layer active dropping for real-time video streaming in 4G access networks, IEEE Systems Journal 4 (4) (2010) 561–572. ⟨http://dx.doi.org/10.1109/JSYST.2010.2086870⟩.

[15] A. Ziviani, B.E. Wolfinger, J.F. Rezende, O.C. Duarte, S. Fdida, Joint adoption of QoS schemes for MPEG streams, Multimedia Tools and Applications 26 (2005) 59–80.

[16] R.R. Pastrana-Vidal, J.C. Gicquel, C. Colomes, H. Cherifi, Sporadic frame dropping impact on quality perception, in: Society of

Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 5292, SPIE, 2004, pp. 182–193.
[17] T. Vargas, J. Guerri, S. Sales, Effect and optimization of burst assembly algorithms for video traffic transmissions over OBS networks, in: 5th International Conference on Broadband Communications, Networks and Systems, 2008. BROADNETS 2008, 2008, pp. 105–112.
[18] K. Ramantas, T. Vargas, J. Guerri, K. Vlachos, A preemptive scheduling scheme for flexible QoS provisioning in OBS networks, in: Sixth International Conference on Broadband Communications, Networks, and Systems, 2009. BROADNETS 2009, 2009.
[19] S. Askar, G. Zervas, D. Hunter, D. Simeonidou, A novel ingress node design for video streaming over optical burst switching networks, in: 37th European Conference and Exhibition on Optical Communication (ECOC), 2011.
[20] European Telecommunications Standards Institute (ETSI), Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H), Standard EN 302 304 ver. 1.1.1., ETSI, 2004.
[21] D. Heyman, T. Lakshman, Source models for VBR broadcast-video traffic, IEEE/ACM Transactions on Networking 4 (1) (1996) 40–48.
[22] ISO JTC 1/SC 29, ISO/IEC 13818-2:2000: Information technology—generic coding of moving pictures and associated audio information: Video, ISO, 2000.
[23] ISO JTC 1/SC 29, ISO/IEC 14496-2:2004: Information technology—coding of audio-visual objects—Part 2: Visual, ISO, 2009.
[24] ISO JTC 1/SC 29, ISO/IEC 14496-10:2010: Information technology – Coding of audio-visual objects—Part 10: Advanced Video Coding, ISO, 2012.
[25] F. Espina, D. Morato, M. Izal, E. Magana, The effect of burst formation timers on video streaming over optical burst switched networks, in: Fifth International Conference on Broadband Communications, Networks and Systems, 2008. BROADNETS 2008, 2008, pp. 282–289.
[26] L. Kleinrock, Queueing Systems, vol. 1, John Wiley and Sons, 1975.
[27] C. Mortici, Improved convergence towards generalized Euler–Mascheroni constant, Applied Mathematics and Computation 215 (9) (2010) 3443–3448.
[28] F. Espina, J. Armendariz, N. Garca, D. Morat, M. Izal, E. Magaa, OBS network model for OMNeT++: a performance evaluation, in: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, SIMUTools '10, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, 2010.
[29] G. Van der Auwera, P. David, M. Reisslein, 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) (2008) 698–718.
[30] P. Seeling, M. Reisslein, B. Kulapala, Network performance evaluation using frame size and quality traces of single-layer and two-layer video: a tutorial, IEEE Communications Surveys Tutorials 6 (3) (2004) 58–78.
[31] W. Zhang, J. Wu, J. Lin, W. Minxue, S. Jindan, TCP performance experiment on LOBS network testbed, in: I. Tomkos, F. Neri, J. Sol Pareta, X. Masip Bruin, S. Snchez Lopez (Eds.), Optical Network Design and Modeling, Lecture Notes in Computer Science, vol. 4534, Springer, Berlin, Heidelberg, 2007, pp. 186–193.
[32] M. Izal, J. Aracil, On the influence of self-similarity on optical burst switching traffic, in: Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE, vol. 3, 2002, pp. 2308–2312.