Video Activity-Based Traffic Policing: A New Paradigm

Evangelos Maratsolas, Polychronis Koutsakis, *Senior Member, IEEE*, and Aggelos Lazaris, *Student Member, IEEE*

***Abstract*—The constant development of new multimedia applications, which are "greedy" in terms of bandwidth and Quality of Service (QoS) requirements, calls for new approaches to the traffic policing problem. This paper proposes, analyzes and presents an extensive performance evaluation of such a new approach, namely the activity-based video traffic policing. Using as a motivation recent work, which has shown that the classic traffic policing mechanisms provide unnecessarily strict policing for conforming but bursty video users, we propose five simple and efficient new mechanisms, which take into consideration and exploit video activity and the Group-of Pictures (GoP) pattern of the video traces. Contrary to the classic approach, our mechanisms do not use a token generator based on a fixed rate, but vary the token generation rate according to specific video activity-based algorithms. The results show significant improvement for conforming users, and reveal that dynamic traffic policing can provide much higher efficiency than the widely used static mechanisms. One of the new mechanisms, the Frame Size Aware Token Bucket, is shown to clearly outperform all other policing approaches for conforming users, and to provide comparable policing results with the classic mechanisms for non-conforming video users.**

***Index Terms*—Group-of-pictures, modeling, traffic policing, video.**

I. Introduction

TRAFFIC from video services is expected to be a substantial portion of the traffic carried by emerging wired and wireless networks [1], [2]. More specifically, according to latest Cisco estimates [24], two-thirds of the global mobile traffic in 2016 will be video traffic. In May 2012, more than 16 billion videos were streamed on Youtube [29]. This explosive growth calls for new sets of traffic control procedures to be implemented in order for the networks to cope with the bursty new applications, which have strict Quality of Service (QoS) requirements. For Variable Bit Rate (VBR) coded video, statistical source models are needed to design networks, which are able to guarantee the strict QoS requirements of the video traffic. Video packet delay requirements are strict, because delays are annoying to a viewer. Whenever the delay experienced by a video packet exceeds the corresponding maximum delay, the packet is dropped, and the video packet dropping requirements are equally strict.

In order to provide the required QoS guarantees, network resources need to be reserved according to both the QoS requirements and the specified traffic parameters of each application. On this subject, one of the fundamental network control issues is the source policing mechanism. The main goal of this control mechanism is to protect the network resources against intentional or unintentional traffic overflow from certain sources. Several policing mechanisms have been proposed in the literature. Traffic that is considered by the mechanisms to be exceeding a user's contract is either dropped immediately or marked, in order to be dropped if needed at any network node, depending on the total traffic load.

Four of the mechanisms which have been most extensively studied (all of them static in nature) are: the Token Bucket and its variations [3]–[11], [25]; the Jumping Window [11]–[13]; the Moving Window (also known as the Sliding Window) [11], [13], [14]; and the Exponentially Weighted Moving Average [11].

In [15], it has been shown that dynamic traffic policing based on accurate H.263 videoconference traffic modeling can clearly outperform the classic static mechanisms, in terms of the percentage of marked packets of conforming users (i.e., users that adhere to their declared traffic parameters). The reason is that the static mechanisms are unable to cope with the burstiness of video traffic, and hence cause the marking of a significant percentage of the transmitted packets. However, accurate prediction is not possible for all types of video sequences, and even when it is, it often involves a higher degree of complexity (e.g., [20]), which would incur additional computational requirements for the system.

Therefore, in the absence of an accurate video traffic model we need to design traffic policing schemes which can improve the performance of the classic mechanisms. In this work, we consider two use cases: a) the case of an edge router (router at the network edge) with a high level of intelligence so that all traffic flows can be efficiently classified and treated according to network policies [30], and 2) the case of a Software Defined Network (SDN) in which a logical controller communicates with the network switches using the Openflow protocol, and installs rules on the switches for flow monitoring in order to perform traffic policing [37]. We propose and evaluate the performance of five new mechanisms, which take into account and exploit the Group-of-Picture (GoP) pattern of H.264 video traffic, as well as the estimates regarding video sources' behavior based on recent video activity. By "video activity" we

Manuscript received January 21, 2013; revised August 06, 2013 and December 23, 2013; accepted February 27, 2014. Date of publication March 11, 2014; date of current version July 15, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Klara Nahrstedt.

E. Maratsolas and P. Koutsakis are with the Department of Electronic and Computer Engineering, Technical University of Crete, Chania 73100, Greece (e-mail: emaratsolas@gmail.com; corresponding author, e-mail: polk@telecom.tuc.gr).

A. Lazaris is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: alazaris@usc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2014.2310592

are referring to the sizes of specific video frames which are crucial in order to derive valid expectations about the upcoming video traffic volume. Our mechanisms are shown to outperform all the classic mechanisms against which they are compared, in terms of providing much better QoS to conforming video users and comparable policing performance with the classic mechanisms regarding non-conforming video users.

The idea of taking advantage of the GoP pattern of video traffic in order to improve the performance of various multimedia systems is not new. Several papers have been proposed in different domains, varying from video traffic modeling to bandwidth allocation. Some representative examples include: [32]–[36]. In [32], a generalization of the Gamma-Beta Auto-Regressive (GBAR) model for MPEG video traffic is proposed that uses the GoP of the video sequence in order to improve the accuracy of the GBAR model. In [34], a Discrete Autoregressive (DAR) model of order 1 is presented that models multiplexed traffic from MPEG-4 video coders by exploiting the GoP pattern of the video traffic in order to model each frame type separately. In [33], a video model has been developed that can capture the network packet loss on the video distortion at the GoP granularity level, and, based on this model, an optimal bandwidth allocation strategy is developed to improve the Quality of Experience (QoE) of the transmitted video in mobile ad-hoc networks. In [35], a rate distortion optimized scheduling algorithm is proposed for video streaming at the GoP granularity level and which generates a number of nested substreams (representing different video qualities) that are dynamically selected to be sent depending on the bandwidth conditions. Finally, in [36], the GoP pattern is taken into account in the calculation of the rate-distortion and the adaptation of QoS in the decision taking framework of MPEG-21 in order to maximize users' experience and the delivered QoS. The proposed work is different from the above papers as none of them exploits the GoP pattern for traffic policing. To the best of our knowledge, this is the first time that activity-based video traffic policing is proposed in the relevant literature.

II. H.264 Video Traces

H.264 has become the most widely accepted video coding standard since the deployment of MPEG-2 at the dawn of digital television. It covers all common video applications ranging from mobile services and videoconferencing to IPTV, HDTV, and HD video storage. Standard H.264 encoders generate three types of video frames: I (intracoded), P (predictive) and B (bidirectionally predictive); i.e., while I frames are intra-coded, the generation of P and B frames involves, in addition to intra-coding, the use of motion estimation and compensation techniques. I frames are, on average, the largest in size, followed by P and then by B frames. The video coding layer of H.264/AVC (Advanced Video Codec) is similar to that of other video coding standards such as MPEG-2 Video. In fact, it uses a fairly traditional approach consisting of a hybrid of block-based temporal and spatial prediction in conjunction with block-based transform coding [17].

In 2007, the Scalable Video Coding (SVC) extension has been added to the H.264/AVC standard. The SVC extension provides temporal scalability, Coarse Grain Scalability (CGS), Medium Grain Scalability (MGS), and SNR scalability in general, spatial scalability, and combined spatio-temporal-SNR scalability [18]. In the rest of this work, we use the term "H.264" to refer to the H.264/AVC video standard.

An important feature of common H.264 encoders is the manner in which frame types are generated. Typical encoders use several Group-of-Pictures (GOP) patterns when compressing video sequences; the GOP pattern specifies the number and temporal order of P and B frames between two successive I frames. A GOP pattern is defined by the distance N between I frames and the distance M between P frames. In our study on video traffic policing, we have used nine different long sequences of H.264 VBR encoded videos in forty-eight formats, from the publicly available Video Trace Library of [18], [19]. The traces used are in Common Intermediate Format (CIF) (i.e. 352×288 pixels) and in High Definition (HD) 720 and 1080 format (i.e. 1280×720p and 1920×1080i, respectively).

In addition, we used several different Quantization Parameters (QP) for the traces under study. The statistics for the nine traces are presented in Table I. The first five traces have a GoP size equal to 16, while the next four have a GoP size equal to 12. The length of all the videos is either 10 or 30 minutes. The data for each trace consists of a sequence of the number of bytes per video frame and the type of video frame, i.e., I, P, or B. The interframe period is 33.3 ms. The B3 and B7 encodings denote the number of B frames between successive I or P frames. Hence, e.g., B7 in a trace with Go $\mathrm{P} = 16$ means that there is one I and one P frame in the GoP, each followed by 7 B frames.

III. Traffic Policing Mechanisms

In this Section, we first describe, briefly, five static traffic policing mechanisms of the literature which we have used in our study. Then, we proceed to present the five new mechanisms we propose.

A. Existing Mechanisms

1) The Token Bucket (TB): The token bucket mechanism has been chosen in the recent past as the traffic descriptor for ATM networks, has been widely studied and is currently used in Cisco equipment [28]. The reason for its popularity is its ability to verify easily whether a source conforms to its declared (at call setup) traffic parameters. The token bucket is the predominant method for network traffic policing/shaping. The leaky bucket mechanism [11] is conceptually close to the token bucket but even stricter; the leaky bucket imposes a hard limit on the source transmission rate, whereas the token bucket allows a certain amount of burstiness (which is necessary for video traffic) while imposing a limit on the average source transmission rate [16]. Therefore, the leaky bucket increases rather than solves the problem of network strictness towards conforming but bursty video traffic. Hence, we do not include the leaky bucket in our study. The basic idea behind the token bucket approach can be described by the following:

a. Tokens are put into the bucket at a certain rate. The bucket has a limited capacity.
b. Each token represents a permission to the source to send a certain number of bytes into the network.

TABLE I
TRACE STATISTICS

Video Name	Codec	Quantization Parameters	Mean (bits/sec)	Peak (bits/sec)
Tokyo Olympics	B3	16	1625669	14992560
		28	305875	6684000
		38	87645	2710080
		48	24260	488160
	B7	16	1715014	15213360
		28	330236	6801120
		38	92371	2820480
		48	23626	539520
Silence of the Lambs	B3	16	707332	12477840
		28	144317	5567520
		38	42705	1989840
		48	14279	448800
	B7	16	744671	12720480
		28	152234	5706000
		38	43737	2078880
		48	13779	473760
Star Wars IV	B3	16	714404	7843440
		28	155670	2520000
		38	46700	1041360
		48	16230	415200
	B7	16	745424	7717680
		28	163573	2561760
		38	48417	1064880
		48	16239	437520
Sony Demo	B3	16	1902281	15898080
		28	384022	6686640
		38	101519	2538960
		48	26609	696960
	B7	16	2004916	15669360
		28	393875	6830400
		38	104736	2627280
		48	26942	764400
NBC News	B3	16	2964255	14244480
		28	438957	5475600
		38	118458	2364720
		48	32523	846480
	B7	16	2975470	14573520
		28	452415	5606160
		38	121284	2429040
		48	31611	893040
Terminator	B2	28	2214602	21722880
		38	701410	7320720
		48	252481	3382800
Kaet's from Mars to China	B2	28	4849710	78457200
Kaet's Horizon	B2	28	1534785	24014640
Sony Demo 2	B2	28	2455620	31852080
		38	675377	11956320
		48	228532	4302240

c. After each transmission from the source, tokens, which correspond to the packets transmitted by the source, are removed from the bucket.
d. Arriving packets of K bytes are conforming and therefore are immediately processed if there are tokens equivalent to K bytes in the bucket. If the current number of accumulated tokens (i.e., its equivalent in bytes) is less than the corresponding number of packets, the exceeding number of packets is nonconforming.
e. Nonconforming packets either wait until the bucket has enough tokens for them to be transmitted or they are discarded or they are marked as nonconforming in order to be discarded in the case of network congestion.
f. If no packets wait to be transmitted, tokens can be accumulated up to the size of the token bucket. If the bucket fills with tokens and the source remains inactive or transmits at a rate lower than the token generation rate, the token buffer overflows and new incoming tokens are discarded, and therefore can not be used by future source packets. In this way the token bucket mechanism imposes an upper limit on the sourceÕs burst length, equal to the token bucket size, i.e., a token bucket permits burstiness, but bounds it. This bound can be described by the following formula:

$$A(s,t) \leq \sigma + \rho \cdot (t-s), \quad s < t, \tag{1}$$

where $A(s,t)$ denotes the amount of traffic leaving the bucket between times s and t, σ is the maximum burst size, and ρ is the token generation rate.

2) The Jumping Window (JW): The Jumping Window mechanism uses windows of a fixed length T side by side through time. A new window starts immediately after the conclusion of the previous one. During a window, only K bytes (or packets) can be submitted by the source to the network. In the case that a source attempts to transmit more than K bytes, the excessive traffic is dropped (or marked as nonconforming, as in the case of the Token Bucket). The mechanism is implemented with the use of a token counter, similar to the one of the Token Bucket, and in each new window the associated packet counter is restarted with an initial value of zero [11].

3) The Moving Window (MW): The Moving Window (Sliding Window) mechanism is similar to the Jumping Window, but more stringent and more complex to implement. This mechanism again ensures that the maximum number of bytes transmitted by a source within any given time interval of duration equal to the fixed window size, T, is upper bounded by K bytes. The difference with the Jumping Window mechanism is that each video frame size is remembered for the width of exactly one window, starting with the specific video frame and ending T frames later. This mechanism can be interpreted as a window which is steadily moving along the time axis, with the requirement that the frame sizes of T frames are stored for the duration of one window [11]. This is the reason that the implementation complexity is considerably higher than for the other two mechanisms (Token Bucket and Jumping Window), as the complexity is directly related to the window size; also, since the content of successive time windows differs by just one frame, it is clear that the mechanism enforces the strictest bandwidth enforcement policy compared to the Token Bucket and the Jumping Window mechanisms.

4) Exponentially Weighted Moving Average (EWMA): The EWMA mechanism uses consecutive-time windows like the JW mechanism. The difference is that the maximum number of accepted packets in the i-th window (N_i) is a function of the allowed mean value of the video trace per interval N and an exponentially weighted sum of the number of accepted packets in the preceding intervals ($X_{i-k}, k \geq 1$) according to the rule:

$$N_i = \frac{N - g \cdot S_{i-1}}{1-g}, \tag{2}$$

where $S_{i-1} = (1-g)X_{i-1} + g \cdot S_{i-2}$.

Eq. (2) can be written as:

$$N_i = \frac{N - (1-g)(g \cdot X_{i-1} + \ldots + g^{i-1} \cdot X_1) - g^{i+1} \cdot S_o}{1-g}, \tag{3}$$

where S_o is the initial value of the EWMA measurement.

The factor g controls the flexibility of the algorithm with respect to the burstiness of the traffic. If $g = 0$, N_i is constant and the algorithm is identical to the JW mechanism. A value of g greater than 0 allows more variable source behavior. Although the computation of N_i can be made efficient for special values of g, the implementation complexity of this mechanism is slightly higher than that of the previous mechanisms [11].

5) The GoP-Based Token Bucket (GBTB): This mechanism was proposed in a previous work of our group, in [25]. It is especially tailored for video traffic, of any GoP pattern. By taking into consideration the GoP pattern, GBTB uses 3 different token buckets, one for each type of video frame (I, P, B). Depending on the type of the video frame, which is expected to arrive at any given instance (this is known from the GoP pattern), the respective token bucket is activated. The mechanism can also be implemented with one token bucket, which uses different token generation rates depending on the expected video frame arrival. GBTB was an attempt to solve the problem of the poor performance of all four mechanisms presented above, in the case of conforming but bursty video users. By exploiting the GoP pattern of each trace, GBTB was shown to largely outperform the standard TB mechanism, despite the fact that the bucket size used in GBTB was much smaller.

B. Activity-Based Video Traffic Policing Mechanisms

The idea proposed in GBTB, i.e., to exploit knowledge about video sources' behavior in order to improve traffic policing for conforming users, is a step towards a new paradigm in video traffic policing. However, GBTB is again static in nature, as it makes no effort to utilize any other knowledge regarding the video source besides the GoP pattern. In this work, we take this idea further and propose new traffic policing mechanisms which exploit knowledge and estimates based on video activity.

1) The Folded Window (FW): The Folded Window mechanism is the closest, conceptually, to GBTB, among the five mechanisms we propose and evaluate. The idea is the following: if the Moving Window is implemented with any window size equal to the GoP or larger than the GoP, then at least one I frame will be included in the "policed" frames. This is not a good choice, because we are forcing the policer to handle together frames of very different sizes (typically, I frames are much larger than P and B frames). What is even worse, in such an implementation, is that during most windows the I frame included in the policing of the P, B frames of the n-th GoP will be the I frame of the $(n+1)$-th GoP, which may significantly differ in size in comparison with the I frame of the n-th GoP (e.g., in the case of a scene change). This can lead to unnecessary and undesirable strictness or leniency of the mechanism. Hence, we propose to implement the Moving Window by "folding" the window in two, and using GoP/2 as the window size. In this way, half of the examined windows contain an I frame and half of them do not. Next, we proceed to generate tokens not by using the mean rate of the source (as in the Moving Window) but by using two different token generation rates, each depending on the number of I,P,B frames in each window. For example, in the case of Go P = 12, B2 encoding, half of the windows will contain 2 P and 4 B frames and will have a token generation rate equal to two times the mean P frame size plus 4 times the mean B frame size; the other half of the policed windows will contain 1 I, 1 P and 4 B frames and will have a different token generation rate, computed again based on the number of each type of frames in the window.

The reason that we do not use an even smaller moving window size than GoP/2 is that such a choice would make the already strict MW-type mechanism even stricter, and would increase the computational complexity through the existence of multiple different token generation rates. The use of GoP/2 window sizes is enough to separate windows including I frames from those that do not.

2) The Variable Exponentially Weighted Moving Average (V-EWMA): The V-EWMA mechanism, which is implemented with three different window sizes (GoP, GoP/2, GoP/4) is a modified version of the EWMA mechanism. The difference between the two mechanisms is that V-EWMA uses variable values of g, depending on the video activity. More specifically, the idea is to adjust the strictness of the mechanism depending on the sizes of the frames of the past few windows; this choice helps us make a prediction on the expected sizes of the frames of the immediate next windows. Hence, by changing the value of g we are able to control the strictness of the mechanism. The V-EWMA policing algorithm is shown below.

Algorithm 1 The V-EWMA policing algorithm

if $N_i, N_{i+1} <$ *mean frame size in a window* **then**
  if $N_i > N_{i+1} > N_{i+2}$ **then**
    $g \leftarrow g + g_{increase}$
  end
else
  if $N_i, N_{i+1} >$ *mean frame size in a window* **then**
    if $N_i < N_{i+1} < N_{i+2}$ **then**
      $g \leftarrow g_{initial}$
    end
  end
end

The logic of the algorithm is that, for a given constant window size and for a user that transmits according to his declared traffic parameters, if three consecutive windows contain frames with sizes smaller than the mean, and the frame sizes in each subsequent window get smaller, then it is highly probable that a window with large frame sizes will soon arrive in one of the next scenes. For this reason, we increase the value of g, in order to make the mechanism more lenient. Even if larger frames do not arrive (i.e., the user continues to transmit below his declared

mean rate) the increase of g does not influence the policing results. However, the value of the $g_{increase}$ parameter is important, as a malevolent user could deliberately send a number of lower-sized frames to acquire the opportunity (via a significantly increased g value) to transmit back-to-back very large chunks of video traffic. Continuing with the logic of the algorithm, if frame sizes in subsequent windows grow larger while they are already above the mean, then for a conforming user we expect soon the arrival of windows with small frame sizes; hence, there is no reason for increased leniency, so we let g return to its initial (default) value. This latter policy is also useful in the case of a non-conforming user, in order to prevent him from continuing to transmit at high rates. We discuss the V-EWMA mechanism further in Section V.

3) The Hybrid V-EWMA Token Bucket (VEWMA-TB): This mechanism, as indicated by its name, is a combination of the V-EWMA and the Token Bucket mechanisms. More specifically, for P and B frames we use the classic TB mechanism, while for I frames we use V-EWMA in order to offer the system the ability to dynamically adjust to the large changes in I frames' sizes, which usually reflect respective changes in the video scene activity. We discuss this mechanism further in Section V, while presenting its results.

4) The Frame Size Aware Token Bucket (FSA-TB): As explained in [23], an I-frame provides direct reference for the B frames and the P frame that immediately follow it in a GoP, and, ultimately, each I frame is the source of all temporal encoding within a GoP. Also, a well-known and extensively studied fact (e.g., [27]) regarding video traffic is that traffic of the same video trace can be divided into high and low activity scenes. Therefore, when an I frame arrives that is much larger/smaller than the mean, it is expected that the P and B frames of the same GoP will exhibit similar behavior, i.e., the I frame is representative of the activity in the whole GoP. Hence, in FSA-TB we use the basic idea of GBTB (three different token buckets, one for each type of video frame), but build upon it by introducing a video activity-based policing component.

More specifically, at the beginning of each GoP we calculate the ratio:

$$R = \frac{\alpha_I}{\mu_I}, \tag{4}$$

where α_I is the allowed size of an I frame, and μ_I is the mean size of the I frames. Then, we use R to adapt, respectively, the token generation rate for the P, B frames of the same GoP as follows:

$$R_P = R \cdot \mu_P, \tag{5}$$

$$R_B = R \cdot \mu_B, \tag{6}$$

where R_P, R_B is the token generation rate of P, and B frames, respectively, and μ_P, μ_B is the mean size of the P, and B frames, respectively. This approach is intuitively very well-tailored to the needs of conforming users, as it is based upon the idea of dynamically adhering to the users' instantaneous needs and not trying to police them statically based on a generic mean rate. This approach also takes into account the case of non-conforming users, which remains the major role of any policing mechanism: the ratio R is based on the allowed size of each I frame, i.e., when a user tries to send I frames which cannot be transmitted in full (insufficient number of tokens), only the portion of the traffic that can be transmitted is entered into the computation of R; if the whole size of the I frame was used in the numerator of R, then the user could potentially be allowed to send very large P and B frames, without any substantial policing.

The efficiency of FSA-TB for both the cases of conforming and non-conforming users will be shown in our results, in Section V.

5) The GoP Modeling Based Jumping Window (GMB-JW): The GMB-JW mechanism is rather different, conceptually, than the other four mechanisms we propose in this work. It shares with them the idea of performing policing based on a variable token generation rate and on information at the GoP scale, but it uses a coarse-grain video traffic model, proposed in [26] for estimating the size of a GoP (in bytes). Of course, many finer-grain models have been proposed in the literature, even within the same work in [26]. However, as we explained in the Introduction, the use of accurate modeling, which was shown in [15] to provide excellent traffic policing results, may not be possible for various reasons. Hence, a coarse-grain estimate of the GoP size needs to be evaluated as an alternative. The simple, first-order Markov chain model proposed in [26] consists of M states, where M is equal to $\frac{G_{max}}{\sigma_G}$, G_{max} denotes the size of the largest GoP and σ_G the standard deviation of the GoP sizes. Thus, the size of the quantization interval is σ_G. The entries of the transition probability matrix $\{P_{i,j}\}$ are estimated by:

$$P_{i,j} = \frac{n_{i,j}}{\sum_{k=1}^{M} n_{ik}}, \tag{7}$$

where $n_{i,j}$ denotes the number of transitions from interval i to interval j.

Our proposed mechanism, GMB-JW, uses a window size equal to GoP and operates similarly to the JW, with the difference that the token generation rate is determined dynamically by the Markov chain model, instead of being constant and equal to the video user's mean declared rate. In Section V, where we evaluate the performance of all the policing mechanisms studied in this work, we will discuss how well the Markov chain model captures the dynamic behavior (i.e., the activity) of the source.

IV. Application Scenarios

Two scenarios where our proposed scheme can be applied are described below. The first example is that of a smart edge router [30] which can enforce traffic policing. The router receives the GoP information of a flow within the header of its first packet (e.g. by sending a GoP identification number that the router can use to lookup for the GoP details in its GoP lookup table), and allows traffic to go through if it is considered conforming. We assume that the core routers have sufficient bandwidth to deal with the bursts due to different frame sizes from multiple edge routers. The second example, which we discuss in more detail, is that of a Software Defined Network (SDN) which provides a significant flexibility compared to traditional networking solutions, since the traffic policing can be done at a logically centralized controller. Software defined networking is a new paradigm

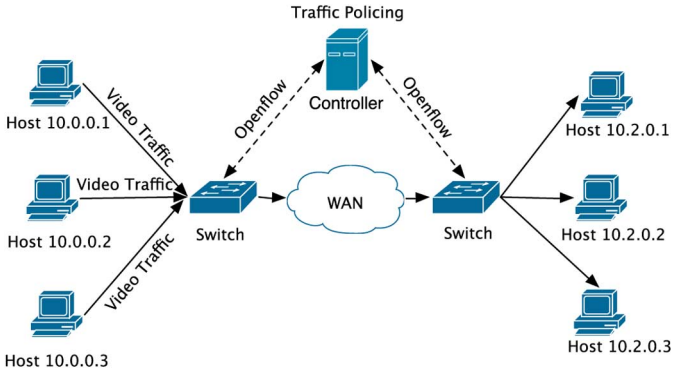


Fig. 1. SDN application scenario where the controller is used to take traffic policing decisions.

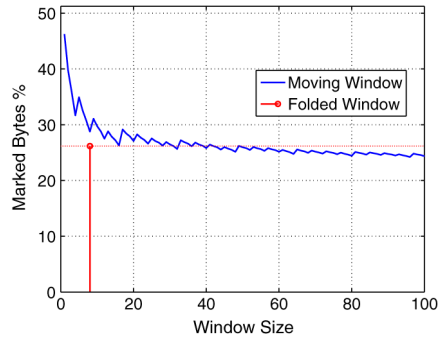


Fig. 2. StarWars B3, QP 16. Marked traffic versus the window size, with the use of the FW and the MW mechanism.

in networking in which the control plane is decoupled from the forwarding plane and built as a distributed system. A logically centralized controller (which runs in one or more servers) has a global view of the network [38] and handles the state distribution of the SDN-enabled switches by collecting data and installing the appropriate rules on them. It also provides a programmatic interface such that developers can build management applications to control the network [37]. In an SDN implementation of our proposed scheme, we assume that the switches will periodically send traffic statistics to the controller using the Openflow protocol ([37]), which in turn will take traffic policing decisions and install rules to the switches in order to enforce the given traffic policy. The traffic policing rules can be installed at the controller either by the network administrators (e.g. different video traffic quality per VLAN, subnet, or flow, etc), or can be pushed to the controller during the call setup. An overview of the SDN application scenario above is shown in Fig. 1 where the video transmitters/receivers are connected to the core through SDN enabled switches. The advantage of the above approach over the traditional networking one is that: a) there is flexibility to implement complex management applications at the controller in a high level language such as Python or C++ in order to implement the traffic policing rules [39], and 2) the controller can take network-wide policy decisions by maintaining status information for the switches in the whole network, thus yielding to more efficient use of the network resources. A complete SDN implementation of our proposed work is outside of the scope of this paper. However, any SDN implementation needs to follow the basic architecture described above.

V. Results and Discussion

Our simulations were conducted in a Matlab environment, with an Intel Core 2 Duo, 2.4 GHz processor. We implemented all of our proposed mechanisms, as well as the classic mechanisms from the literature. The proposed mechanisms can be applied to both wired and wireless networks, therefore we do not tailor our performance evaluation to the parameters of any specific network. It should be mentioned, however, that in wireless networks any traffic policing mechanism should co-operate with the scheduling module, in order to accommodate "lagging" users, i.e., users who have experienced a noisy channel and therefore have been involuntarily transmitting at a decreased rate.

In our simulations, we assume that each user is transmitting only one type of traffic (i.e., the video chosen for the simulation) which arrives at the edge router and we derive our results by implementing the policing mechanisms for the whole duration of the video trace. The trace durations vary between 10 and 74 minutes. Given that we consider the case of an edge router with a high level of intelligence, we assume that it will be able to maintain the GoP information for all video transmissions. Still, even if that is not possible due to a very large traffic load, a rather easy alternative would be that certain video packets of a source could periodically include in their headers the GoP information. In the case of a router with GoP/video activity policing which is followed by a simple token bucket in another part of the network, the token bucket will eradicate the advantage offered by our policer (i.e., a significantly larger portion of the traffic will be marked, as it will be shown from our results) but other than that, no compatibility problems will arise; our proposed mechanisms do not require/impose any changes in the network.

We first compare the conceptually similar new and classic mechanisms in couples, and then we compare all mechanisms in terms of their policing efficiency both for conforming users (the problem at the core of this work) and non-conforming users. Our results show the percentage of traffic that the router would falsely mark as non-conforming, although the users are conforming. This percentage is the criterion for the evaluation of the algorithms. The respective percentage is also the criterion for the evaluation of the algorithms in the case of non-conforming users, where the higher the percentage of marked traffic, the more efficient the policer is.

A. Results with the Use of FW

We compare the Folded Window with the Moving Window mechanism. Regarding the MW, we started with a very strict policy, controlling the source transmission rate per video frame (allowing the source to transmit no more than the mean rate in every frame), and progressively we increased the window size, up to 100 video frames. The percentage of marked traffic decreases as the window size increases, for all the traces.

All of our results, in Figs. 2–10, confirm the problems that we expected MW to have because of its nature. When the MW is implemented on very short windows (less than GoP/2) it is

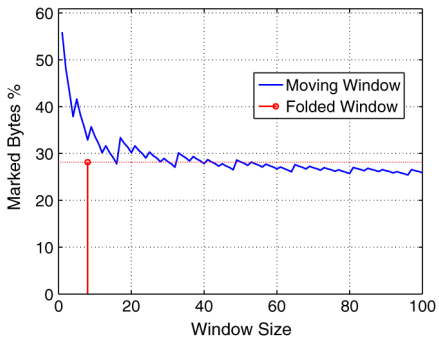


Fig. 3. StarWars B3, QP 28. Marked traffic versus the window size, with the use of the FW and the MW mechanism.

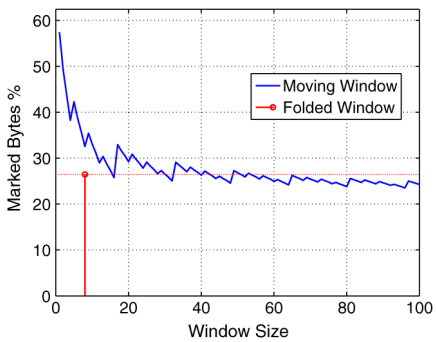


Fig. 4. StarWars B3, QP 38. Marked traffic versus the window size, with the use of the FW and the MW mechanism.

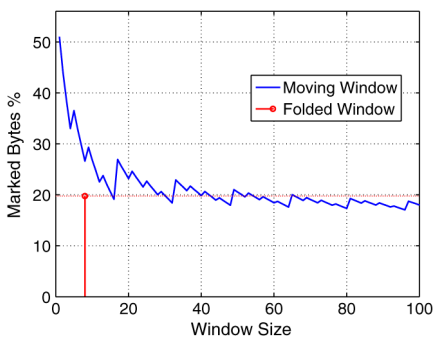


Fig. 5. StarWars B3, QP 48. Marked traffic versus the window size, with the use of the FW and the MW mechanism.

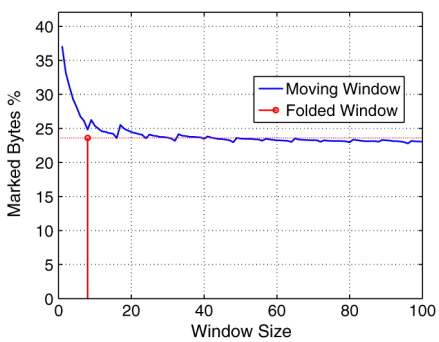


Fig. 6. Sony Demo B7, QP 16. Marked traffic versus the window size, with the use of the FW and the MW mechanism.

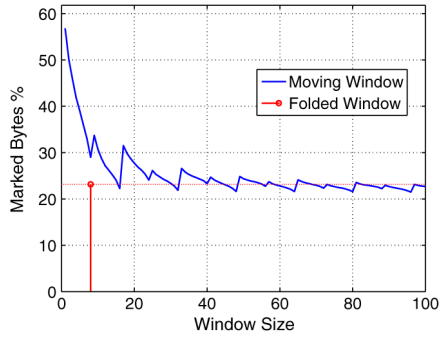


Fig. 7. Sony Demo B7, QP 28. Marked traffic versus the window size, with the use of the FW and the MW mechanism.

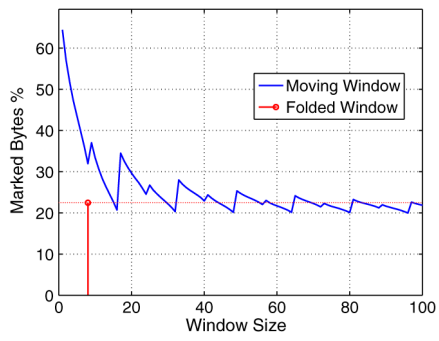


Fig. 8. Sony Demo B7, QP 38. Marked traffic versus the window size, with the use of the FW and the MW mechanism.

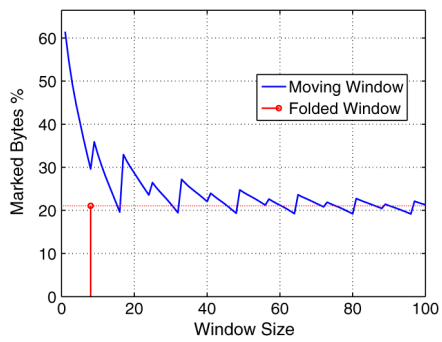


Fig. 9. Sony Demo B7, QP 48. Marked traffic versus the window size, with the use of the FW and the MW mechanism.

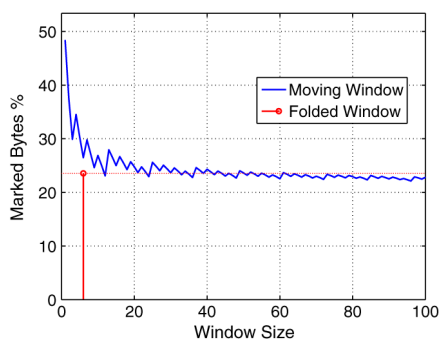


Fig. 10. Kaet's from Mars to China B2, QP 28. Marked traffic versus the window size, with the use of the FW and the MW mechanism.

too strict, not allowing the source any room to "breathe" and, of course, not taking into account the different video frame sizes due to the different frame types. When the window size is equal or reasonably larger than the GoP (even double its size) then, as explained in Section III.B, the MW forces the policer to handle together frames of very different sizes and belonging to different

TABLE II
MEAN MARKED TRAFFIC (%) FOR ALL TRACES, WITH THE USE OF THE FW AND THE MW MECHANISM

Movie	Codec	MW Mean Marked Traffic	FW Mean Marked Traffic	Comparison	Comparison %
Tokyo	B3	27.5	25.6	−1.9	−6.90%
	B7	27.4	26.4	−1.0	−3.60%
Silence of the Lambs	B3	32.5	29.5	−3.0	−9.20%
	B7	31.9	29.9	−2.0	−6.30%
Star Wars	B3	30.2	25.1	−5.1	−16.90%
	B7	29.2	25.6	−3.6	−12.30%
Sony Demo	B3	29.4	22.1	−7.3	−24.80%
	B7	28.8	22.6	−6.2	−21.50%
Nbc News	B3	24.4	19.7	−4.7	−19.30%
	B7	23.7	20.1	−3.6	−15.20%
Terminator	B2	23.3	21.8	−1.5	−6.40%
Kaet's from Mars to China	B2	26.5	23.6	−2.9	−11.10%
Kaet's Horizon	B2	25.5	12.6	−12.9	−50.50%
Sony Demo 2	B2	31.0	22.7	−8.3	−26.80%

GoPs, and possibly to different scenes. Finally, even in the case where the MW is implemented for a window equal to GoP/2, i.e. equal to the FW, the fact that the tokens are generated "blindly" in the MW, without taking into account the video frames' differentiation, leads to steadily worse results than those achieved by the FW.

For these reasons, as shown in the Figures, FW marks a smaller percentage of the traffic transmitted by conforming users in comparison to MW. The MW mechanism needs to be implemented with very lenient parameters (windows larger than 50, 60 or even 80 frames in some cases) in order to be able to achieve the same results that the FW achieves with the GoP/2 window size (equal to 6 or 8 frames, depending on the trace).

The conclusions derived by the Figures are further confirmed by the results presented in Table II, where we make a head-to-head comparison between FW and MW for a window size equal to GoP/2. Table II presents the averages for each trace over all QPs. The reduction in marked Bytes that is achieved with FW is constantly larger for the B3 in comparison to the B7 encoded movies. The reason is that in a B3 encoding with Go P $= 16$ the number of P frames in a GoP is equal to three, whereas in a B7 encoding there is only one P frame in the GoP. Therefore, in a B7 encoding fourteen out of the sixteen frames of the GoP are B frames, creating thus a "uniformity" of frame sizes and decreasing the effect of the FW mechanism (although FW clearly outperforms MW for both B3 and B7 encodings). On the other hand, in a B7 encoding, the better policing of P frames through the "isolation" of I frames in one half of the window gives improved results. The mean reduction in marked bytes, with the use of the FW, is 17%, over all the traces. Still, we need to point out that, despite the improvement achieved by the FW mechanism, the percentage of marked traffic remains very high, given that we are considering conforming users.

B. Results with the Use of V-EWMA

We derived our results with the use of three different initial values for g (0.2, 0.3, 0.5) and for three window sizes: GoP, GoP/2, GoP/4. As explained in Section III.B, the value of the $g_{increase}$ parameter is important in order to control the leniency of the mechanism in the case of non-conforming users: in the case that the algorithm leads to consecutive increases of g, we do not want to allow video users to transmit large bursts for a significant period of time. We have experimented with various different $g_{increase}$ values, to study how the step size in the V-EWMA algorithm influences the results. As expected, especially for the burstier video traces used in our study, a larger $g_{increase}$ value led to smaller percentages of marked traffic for conforming users.

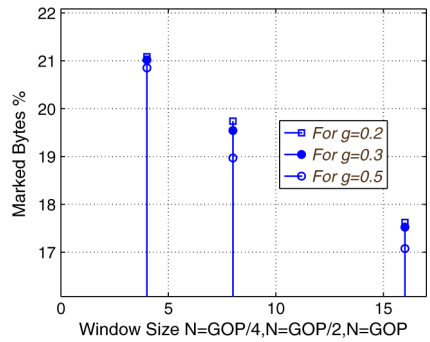


Fig. 11. Tokyo B3, QP 16. Marked traffic versus the window size, with the use of the V-EWMA mechanism for three window sizes (4, 8, 16 frames respectively).

However, we found that this decrease was rather trivial, regardless of the initial value of g: when increasing, for example, the step size by 5 ($g_{increase} = 0.02 \rightarrow g_{increase} = 0.1$) the decrease in marked traffic for a window size equal to GoP fell in the range of 3-4% of the total marked traffic, i.e., it seldom exceeded 1% of the total transmitted traffic and very often was much smaller than 1%. For smaller window sizes (e.g., GoP/4) where a leniency in the value of g can have a larger influence, the decrease in marked traffic again did not exceed 1.5% of the total transmitted traffic. Given the small gains offered by larger $g_{increase}$ values and the possible policing hazards in the case of non-conforming users, we derived the following results with a $g_{increase}$ value equal to 0.05.

As shown in Figs. 11–16, the percentage of marked traffic decreases as the window size and the initial value of g increase. This is expected, as a larger window size increases leniency, and an increase in the initial value of g has the same effect.

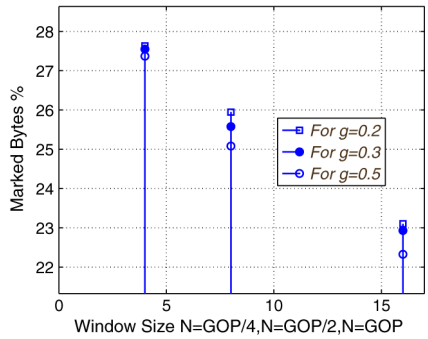


Fig. 12. Tokyo B3, QP 28. Marked traffic versus the window size, with the use of the V-EWMA mechanism.

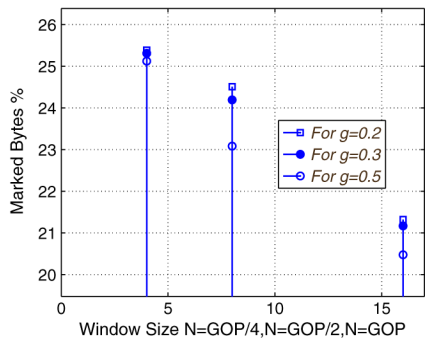


Fig. 13. Tokyo B3, QP 38. Marked traffic versus the window size, with the use of the V-EWMA mechanism.

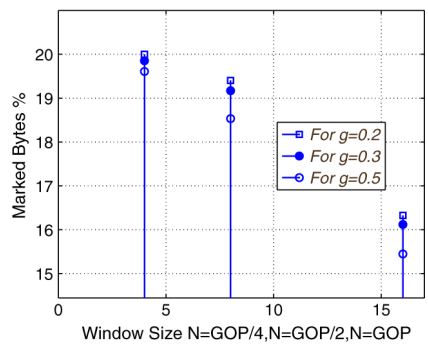


Fig. 14. Tokyo B3, QP 48. Marked traffic versus the window size, with the use of the V-EWMA mechanism.

Table III presents the comparison of the results of EWMA and V-EWMA for conforming users. In these simulations the initial value of g in V-EWMA is set to 0.5, and the constant value of g in EWMA is set to 0.9. The reason for this choice is to make EWMA very lenient, so that we can compare V-EWMA against EWMA while giving EWMA a significant advantage. This is because we are focusing on conforming users; this choice would backfire if EWMA with $g = 0.9$ was implemented for a non-conforming user. For both EWMA and V-EWMA the window size is set equal to GoP. The results clearly show that V-EWMA excels for all video traces in comparison to EWMA when comparing their average marked traffic for each trace, over all QPs. The mean marked traffic is larger for the B7 encoded movies than it is for the B3 encoded ones, and the significant improvement achieved by V-EWMA is comparable, for

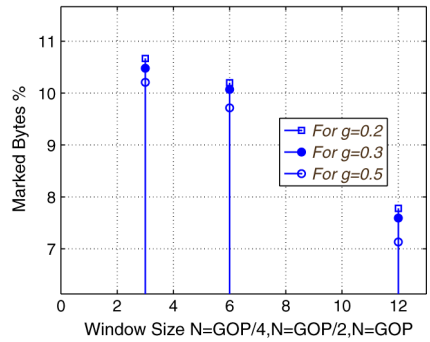


Fig. 15. Kaet's horizon B2, QP 28. Marked traffic versus the window size, with the use of the V-EWMA mechanism.

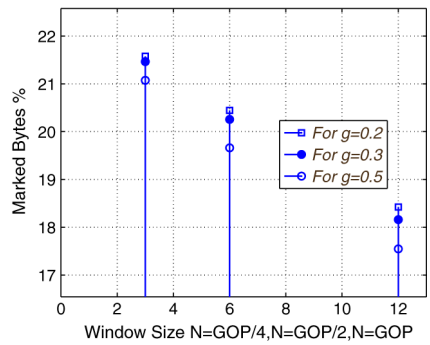


Fig. 16. Kaet's from Mars to China B2, QP 28. Marked traffic versus the window size, with the use of the V-EWMA mechanism.

B3 and B7 encodings. This result is different than the respective result for the FW mechanism (larger reduction for B3 encoded movies). The reason is that the B3 encoding, as already explained, corresponds to the existence of 3 P frames within a GoP, whereas the B7 encoded movies contain only 1 P frame in each GoP; the fact that in B3 encodings the frequency of P frames in-between B frames is larger means that it is more difficult to encounter a situation with three consecutive frames each one larger than the next and all of them smaller than the mean. Therefore, g increases more rarely, so the mechanism is less lenient for B3 encoded movies. Still, when g does increase in the policing of B3 encoded movies, this increase allows the source to "breathe" more than it does for B7 encodings, given the larger number of P frames which need to be policed. Hence, the results for B3 and B7 encoded movies are comparable.

The mean reduction in marked bytes, with the use of V-EWMA, is 15% over all the traces. Our simulations have shown that in order to achieve similar results with V-EWMA, EWMA needs to use $g = 0.9$ and a window size at least equal to 60. This, however, would again be a bad choice in the case of policing a non-conforming user, as it will allow the user to violate his contract much more easily. Finally, we note again, as in Section IV.A for the FW, that the marked traffic of conforming users remains very high, despite the significant improvement achieved by V-EWMA.

C. Results with the Use of VEWMA-TB

As explained in Section III.B, the large changes in I-frame sizes motivated us to propose this hybrid mechanism. We

TABLE III
MEAN MARKED TRAFFIC (%) FOR ALL TRACES, WITH THE USE OF THE EWMA AND THE V-EWMA MECHANISM

Movie	Codec	EWMA (g=0.9) (Window size = GOP) Mean Marked Traffic	V-EWMA (g=0.5) (Window Size = GOP) Mean Marked Traffic	Comparison	Comparison %
Tokyo	B3	21.9	18.8	-3.1	-14.20%
	B7	22.6	19.2	-3.4	-15.00%
Silence of the Lambs	B3	25.4	22.9	-2.5	-9.80%
	B7	25.8	23.6	-2.2	-8.50%
Star Wars	B3	19.4	15.5	-3.9	-20.10%
	B7	20.0	15.8	-4.2	-21.00%
Sony Demo	B3	18.8	16.7	-2.1	-11.20%
	B7	19.5	17.5	-2.0	-10.30%
Nbc News	B3	13.9	11.3	-2.6	-18.70%
	B7	14.4	11.6	-2.8	-19.40%
Terminator	B2	17.2	14.7	-2.5	-14.50%
Kaet's from Mars to China	B2	19.9	17.5	-2.4	-12.10%
Kaet's Horizon	B2	9.2	7.1	-2.1	-22.80%
Sony Demo 2	B2	19.2	18.0	-1.2	-6.30%

TABLE IV
MEAN MARKED TRAFFIC (%) FOR ALL TRACES, WITH THE USE OF THE EWMA, V-EWMA AND VEWMA-TB MECHANISMS

Movie	Codec	EWMA (g=0.9) Window Size = GOP Mean Marked Traffic	V-EWMA (g=0.5) Window Size = GOP Mean Marked Traffic	VEWMA-TB (g=0.5) Window Size = GOP Mean Marked Traffic	Comparison (%) V-EWMA vs. VEWMA-TB
Tokyo	B3	21.9	18.8	10.6	-43.60%
	B7	22.6	19.2	16.2	-15.60%
Silence of the Lambs	B3	25.4	22.9	10.3	-55.00%
	B7	25.8	23.6	15.8	-33.10%
Star Wars	B3	19.4	15.5	8.9	-42.60%
	B7	20.0	15.8	13.4	-15.20%
Sony Demo	B3	18.8	16.7	8.2	-50.90%
	B7	19.5	17.5	12.6	-28.00%
Nbc News	B3	13.9	11.3	8.2	-27.40%
	B7	14.4	11.6	10.9	-6.00%
Terminator	B2	17.2	14.7	8.8	-40.10%
Kaet's from Mars to China	B2	19.9	17.5	8.3	-52.60%
Kaet's Horizon	B2	9.2	7.1	2.8	-60.60%
Sony Demo 2	B2	19.2	18.0	6.4	-64.40%

use the V-EWMA mechanism for I frames and the Token Bucket for policing P and B frames. In this way, we try to "capture" significant changes in activity (given that the changes in I frame size reflect activity changes). This approach also implicitly studies how significant the size changes in successive B frames are (since they constitute the vast majority of frames in a GoP). If they are significant, V-EWMA should outperform VEWMA-TB; if not, then the comparison among I frames should show better results for conforming users, hence VEWMA-TB will outperform V-EWMA. By studying the actual sizes of B frames in all of the respective video files, we had the strong indication that VEWMA-TB would provide better results (size changes were small, for B frames); but we had to implement the algorithm to find out if this was true. Hence, we compare VEWMA-TB against V-EWMA and EWMA. For V-EWMA and VEWMA-TB the initial g value is set to 0.5, and for EWMA the g value is set to 0.9, as in Section IV.B. The window size is chosen equal to GoP for all mechanisms. Given that the comparison between V-EWMA and EWMA has already been presented in Section IV.B, we only present here the % comparison between our two proposed mechanisms, V-EWMA and VEWMA-TB (we include EWMA only for completeness of the comparison among the three similar in nature mechanisms).

Our results, presented in Table IV, show that VEWMA-TB greatly outperforms V-EWMA (and EWMA, which achieves even worse results). The reduction in marked traffic is larger for B3 encoded traces than for B7 ones. The reasons for these results are the same with those explained in Section IV.A (more P frames in B3 encoding).

The mean reduction in marked bytes with the use of VEWMA-TB is 38% in comparison to V-EWMA and close to 50% in comparison to EWMA, over all the traces. Our simulations have shown that in order to achieve similar results with VEWMA-TB, EWMA needs to use $\mathrm{g} = 0.9$ and a window size at least equal to 80.

The excellence of VEWMA-TB in comparison to V-EWMA reveals that the algorithm offers much better QoS to conforming users by solely focusing on the changes of I frames' sizes.

D. Results with the Use of FSA-TB

Since GBTB was shown in [25] to outperform all classic mechanisms, including the original Token Bucket, we compare in this section our proposed policing mechanism, FSA-TB,

TABLE V
MEAN MARKED TRAFFIC (%) FOR THE MOVIES, GROUPED BY CODEC, WITH THE USE OF THE GBTB AND THE FSA-TB MECHANISM

Movie	Codec	GBTB Mean Marked Traffic	FSA-TB Mean Marked Traffic	Comparison	Comparison %
Tokyo	B3	8.2	3.1	−5.1	-62.20%
	B7	8.5	3.2	−5.3	-62.40%
Silence of the Lambs	B3	17.1	6.1	−11.0	-64.30%
	B7	17.2	6.2	−11.0	-64.00%
Star Wars	B3	6.5	0.7	−5.8	-89.20%
	B7	6.9	0.7	−6.2	-89.90%
Sony Demo	B3	10.2	5.3	−4.9	-48.00%
	B7	10.3	5.8	−4.5	-43.70%
Nbc News	B3	2.2	1.0	−1.2	-54.50%
	B7	2.3	1.1	−1.2	-52.20%
Terminator	B2	7.2	0.6	−6.6	-91.70%
Kaet's from Mars to China	B2	2.2	1.0	−1.2	-54.50%
Kaet's Horizon	B2	2.6	0.7	−1.9	-73.10%
Sony Demo 2	B2	10.4	6.8	−3.6	-34.60%

TABLE VI
MEAN MARKED TRAFFIC (%) FOR THE MOVIES, GROUPED BY CODEC, WITH THE USE OF THE JW AND THE GMB-JW MECHANISM

Movie	Codec	JW Window Size = GOP Mean Marked Traffic	GMB-JW Window Size = GOP Mean Marked Traffic	Comparison	Comparison %
Tokyo	B3	24.6	21.1	-3.5	-14.20%
	B7	25.5	21.6	-3.9	-15.30%
Silence of the Lambs	B3	28.1	22.6	-5.5	-19.60%
	B7	28.5	24.1	-4.4	-15.40%
Star Wars	B3	23.6	19.8	-3.8	-16.10%
	B7	24.3	19.8	-4.5	-18.50%
Sony Demo	B3	20.2	16.7	-3.5	-17.30%
	B7	20.9	15.9	-5.0	-23.90%
Nbc News	B3	17.9	15.4	-2.5	-14.00%
	B7	18.4	15.9	-2.5	-13.60%
Terminator	B2	21.0	17.6	-3.4	-16.20%
Kaet's from Mars to China	B2	22.6	17.0	-5.6	-24.80%
Kaet's Horizon	B2	11.4	9.0	-2.4	-21.10%
Sony Demo 2	B2	20.4	15.0	-5.4	-26.50%

against GBTB. The results presented in Table V show that FSA-TB (with a bucket size equal to the peak of the source) not only outperforms GBTB, but is also the only mechanism among all that we have studied and proposed which achieves low percentages of marked traffic ($\sim 3\%$ over all the traces). The mean reduction in marked bytes, with the use of the FSA-TB, is 63% over all the traces, in comparison to GBTB.

E. Results with the Use of GMB-JW

The results for GMB-JW have been derived as the averages of 10 independent runs. Table VI shows that even a coarse-grain model, such as the one used in GMB-JW, is enough to provide a substantial decrease in the "unfairly" marked traffic (unfairly in the sense that the users are conforming), in comparison to the classic JW mechanism. However, it is again clear that this decrease is not sufficient - in almost all of the studied cases the percentage of marked traffic was higher than 10%. The mean reduction in marked bytes, with the use of the GMB-TB, is 18% in comparison to JW, over all the traces.

F. Comparison of All Mechanisms

In this section we compare FSA-TB against all other mechanisms used in our study. We first compare the mechanisms in the case of conforming users, which has been the main subject of this work, and then in the case of non-conforming users, which is a top priority for every policing mechanism.

TABLE VII
COMPARISON OF FSA-TB WITH ALL THE OTHER MECHANISMS USED IN OUR STUDY, FOR CONFORMING USERS

		Average Improvement (Decrease) in Marked Traffic of Conforming Users
FSA-TB	MW Window Size = GoP/2	90%
	FW Window Size = GoP/2	87%
	JW Window Size = GoP	87%
	EWMA (g=0.9) Window Size = GoP	85%
	GMB-JW Window Size = GoP	83%
	V-EWMA (g=0.5) Window Size = GoP	83%
	TB Bucket Size = 5×Peak	83%
	VEWMA-TB (g=0.5) Window Size = GoP	68%
	GBTB	63%

1) Conforming Users: In this comparison, we have chosen to implement all other mechanisms with parameters which increase their leniency. Of course, this is a bad choice in the case

TABLE VIII
COMPARISON OF FSA-TB WITH GBTB REGARDING THE REQUIRED TOKEN GENERATION RATE TO ACHIEVE 1% AND 0.01% MARKED TRAFFIC FOR CONFORMING USERS

Movie	Codec	QP	Marked Traffic (%) with CGTB	GBTB X times the mean (for 1%)	GBTB X times the mean (for 0.01%)	Marked Traffic (%) with FSA-TB	FSA-TB X times the mean (for 1%)	FSA-TB X times the mean (for 0.01%)
Tokyo	B3	16	6.9	1.19	1.24	1.2	1.04	1.28
		28	10.2	1.33	1.4	3.4	1.29	1.48
		38	8.8	1.32	1.42	4.3	1.33	1.48
		48	6.7	1.23	1.31	3.6	1.24	1.36
	B7	16	7.5	1.19	1.26	1.1	1.03	1.28
		28	10.3	1.30	1.4	3.2	1.28	1.44
		38	8.2	1.32	1.42	4.3	1.33	1.48
		48	7.8	1.25	1.33	4.0	1.26	1.40
Silence of the Lambs	B3	16	19.7	2.96	3.26	3.6	1.76	2.44
		28	20.5	2.96	3.27	7.0	2.50	3.24
		38	17.3	2.67	3.33	7.9	2.68	3.40
		48	10.9	1.79	3.52	5.7	1.80	2.56
	B7	16	20.1	2.80	3.01	3.5	1.74	2.40
		28	20.6	2.86	3.16	6.9	2.48	3.20
		38	17.3	2.72	3.38	8.1	2.73	3.44
		48	10.9	1.90	2.60	6.1	1.91	2.64
Star Wars	B3	16	6.2	1.48	1.68	0.1	1.00	1.08
		28	6.7	1.44	1.65	0.3	1.00	1.24
		38	7.3	1.48	1.58	1.2	1.03	1.52
		48	5.9	1.31	1.56	1.3	1.04	1.60
	B7	16	6.5	1.39	1.55	0.1	1.00	1.08
		28	6.9	1.42	1.55	0.2	1.00	1.20
		38	7.6	1.42	1.53	1.1	1.03	1.52
		48	6.4	1.31	1.59	1.4	1.05	1.64
Sony Demo	B3	16	8.6	1.10	1.13	0.7	1.00	1.16
		28	11.1	1.17	1.23	6.0	1.19	1.28
		38	11.7	1.18	1.22	7.9	1.18	1.28
		48	9.4	1.13	1.16	6.4	1.14	1.20
	B7	16	9.2	1.10	1.18	2.5	1.10	1.24
		28	10.8	1.18	1.23	6.0	1.19	1.28
		38	11.6	1.17	1.21	8.0	1.18	1.28
		48	9.6	1.13	1.16	6.7	1.15	1.20
Nbc News	B3	16	1.5	1.02	1.05	0.3	1.00	1.12
		28	2.2	1.02	1.77	1.1	1.02	1.16
		38	2.8	1.03	2.98	1.4	1.03	1.20
		48	2.1	1.02	3.97	1.2	1.02	1.24
	B7	16	1.5	1.01	1.06	0.3	1.00	1.12
		28	2.2	1.02	1.50	1.1	1.02	1.16
		38	3.0	1.03	2.50	1.5	1.03	1.20
		48	2.3	1.02	3.59	1.3	1.02	1.24
Terminator	B2	28	9.1	1.28	1.47	0.4	1.00	1.20
		38	7.2	1.30	1.71	0.6	1.00	1.16
		48	5.4	1.29	1.67	0.9	1.00	1.12
From Mars to China	B2	28	2.2	1.02	1.04	1.0	1.00	1.08
Horizon	B2	28	2.6	1.91	3.76	0.7	1.00	1.32
Sony Demo 2	B2	28	10.2	1.22	1.28	5.8	1.23	1.32
		38	10.8	1.19	1.24	7.7	1.21	1.28
		48	10.1	1.14	1.18	6.9	1.15	1.24

of non-conforming users; it is used here in order to provide a "fairer" comparison for all mechanisms against FSA-TB.

FSA-TB is shown to outperform all mechanisms for every video trace used in our study, and to significantly decrease the percentage of marked traffic on average, as shown in Table VII. This result is important for both conforming video users and for the network service providers, in order to ensure a fairer treatment of conforming users? traffic.

Given that GBTB was shown, from all of our results, to be the second best policing mechanism for conforming users, we proceeded to make one more comparison between FSA-TB and GBTB. We have found, via simulation, the required token generation rate for which each mechanism achieves 1% ([21]) and 0.01% ([22]) of marked traffic, i.e., very good and excellent video QoS, respectively. These rates are presented in Table VIII for each trace used in our study. On average, FSA-TB needs a 15% smaller token generation rate than GBTB to achieve 1% marked traffic, and a 21% smaller token generation rate than GBTB to achieve 0.01% marked traffic over all the traces studied.

An interesting observation is that GBTB is shown in these results to partially "close the gap" between its own performance and that of FSA-TB, in the sense that although FSA-TB is again clearly better, GBTB?s performance is not as far behind as it was in all the previous results, and for some traces GBTB is even marginally better (i.e., it needs a very marginally smaller token generation rate to achieve the same low percentage of marked traffic as FSA-TB). The reason for these results can be traced

TABLE IX
MARKED TRAFFIC (%) FOR NON-CONFORMING USERS

Movie 1 Transmitted		Movie 2 Declared	Movie 3 Declared	Movie 4 Declared	Movie 5 Declared
Sony Demo B7-QP16		Silence of the Lambs B3-QP16	Star Wars B3-QP16	Tokyo Olympics B3-QP28	Nbc News B7-QP48
Mechanism	**Marked Traffic (%)**	**Marked Traffic (%)**	**Marked Traffic (%)**	**Marked Traffic (%)**	**Marked Traffic (%)**
FW	23	73	72	88	98
FSA-TB	2.5	60	58	90	99
VEWMA-TB	18	46	48	55	60
V-EWMA	19	62	60	76	95
GMB-JW	20	33	35	60	97
EWMA	21	68	67	86	98
GBTB	9	74	73	88	98
JW	23	68	67	85	98
TB	21	67	67	85	98
MW	25	68	68	86	98

to the nature of both mechanisms. FSA-TB uses the I frames? sizes to dynamically adjust, on the fly, the token generation rate for P and B frames; i.e., FSA-TB acquires the needed token generation rate for P and B frames on its own, and can benefit only from a larger token generation rate for I frames. On the other hand, GBTB uses a static token generation rate for each frame type, therefore the increase in token generation rate has a significant positive influence (larger leniency) for all types of frames; hence, a uniform increase of the token generation rate for all frames can significantly improve its performance.

2) Non-Conforming Users: Table IX presents the results of all mechanisms used in our study for 4 cases of non-conforming users. Movie 1 is the video that is actually transmitted by malevolent users, who falsely declare the traffic parameters of one of the four other movies shown in the Table. We also present the percentage of marked traffic for a conforming user transmitting Movie 1.

The results show, as intuitively expected, that among our mechanisms the ones that exhibited greater strictness for conforming users (e.g., FW and V-EWMA) succeed in marking a higher percentage of traffic transmitted by non-conforming users than the more lenient ones (e.g., VEWMA-TB, which however is fairer for conforming users).

Most importantly, these results show that the dynamic nature of FSA-TB and the other mechanisms we propose in this work, which exploit knowledge and estimates based on video activity, are sufficient in order to steadily provide comparable (and for FSA-TB in some cases stricter) policing results for non-conforming users when compared against the classic mechanisms.

VI. CONCLUSION

In this work we have proposed, for the first time in the relevant literature to the best of our knowledge, a new paradigm for policing bursty video traffic.

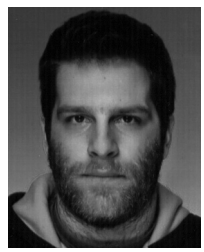
Classic mechanisms such as the Token Bucket, the Moving and the Jumping Window and the Exponentially Weighted Moving Average perform static policing, i.e., policing based on a video source's declared traffic parameters. The five new mechanisms proposed in our work, on the contrary, vary the token generation rate by exploiting knowledge regarding the GoP pattern and by making estimates regarding the expected traffic; these estimates are based on video activity, i.e., on the already transmitted video content (sizes of specific frames and GoPs).

Our mechanisms are shown to provide very significant improvements, in terms of the offered QoS to conforming users, in comparison to the static mechanisms with which they are conceptually similar. This is an important result because the static nature of the classic mechanisms is contradictory with the bursty, dynamic nature of video traffic and this results in very high percentages of unfairly marked traffic for conforming users. Additionally, as shown in [31], TCP performance can degrade significantly when policed with classic static mechanisms and the solution of using a large token bucket size creates other problems related with packet loss and latency; the solution of using a dynamic bucket size, in [31], was shown to be TCP friendly. Similarly, our mechanisms, which are dynamically tailored to each transmitted video, can be equally beneficial to TCP. Our FSA-TB mechanism, in particular, is shown to not only outperform all other mechanisms (classic and new) in providing the best QoS to conforming video users, but also to provide at least comparable and often better policing results in the case of non-conforming users. Based on all of our results, we believe that this new paradigm we introduce, of activity-based video traffic policing, is very promising for next generation networks in order to handle bursty multimedia traffic.

REFERENCES

[1] M. Etoh and T. Yoshimura, "Advances in wireless video delivery," *Proc. IEEE*, vol. 93, no. 1, pp. 111–122, 2005.
[2] S. M. Cherry, "Fiber to the home," *IEEE Spectr.*, vol. 41, no. 1, pp. 42–43, 2004.
[3] V. Raghunathan, S. Ganeriwal, M. Srivastava, and C. Schurgers, "Energy efficient wireless packet scheduling and queuing," *ACM Trans. Embed. Comput. Syst.*, vol. 3, no. 1, pp. 3–23, 2004.
[4] G. Procissi, A. Garg, M. Gerla, and M. Y. Sanadidi, "Token bucket characterization of long-range dependent traffic," *Comput. Commun.*, vol. 25, no. 11/12, pp. 1009–1017, 2002.
[5] M. Fiddler and V. Sander, "A parameter based admission control for differentiated services networks," *Comput. Netw.*, vol. 44, no. 4, pp. 463–479, 2004.
[6] J.-Y. Le Boudec, "Some properties of variable length packet shapers," *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 329–337, 2002.
[7] J. Sairamesh and N. Shroff, "Limitations and pitfalls of leaky bucket-a study with video traffic," in *Proc. 3rd IEEE Int. Conf. Computer Communications Networks (ICCCN '94)*.
[8] N. L. S. Fonseca, G. S. Mayor, and C. A. V. Neto, "On the equivalent bandwidth of self-similar sources," *ACM Trans. Model. Comput. Simulat.*, vol. 10, no. 2, pp. 104–124, 2000.
[9] T. Ors and S. P. W. Jones, "Performance optimizations of ATM input control using an adaptive leaky-bucket mechanism," in *Proc. 3rd IFIP Workshop Performance Modeling Evaluations of ATM Networks*, 1995.
[10] E. W. Knightly, "Enforceable quality of service guarantees for bursty traffic streams," in *Proc. IEEE INFOCOM*, 1998, pp. 635–642.

[11] E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 3, pp. 325–334, 1991.
[12] P.-Y. Kong, K.-C. Chua, and B. Bensau, "A novel scheduling scheme to share dropping ratio while guaranteeing a delay bound in a multi-Code-CDMA network," *IEEE/ACM Trans. Netw.*, vol. 11, no. 6, pp. 994–1006, 2003.
[13] C. V. N. Albuquerque, M. Faerman, and O. C. M. B. Duarte, "Implementations of traffic control mechanisms for high speed networks," in *Proc. IEEE Int. Telecommunications Symp.*, 1998, pp. 177–182.
[14] A. R. Reibman and A. W. Berger, "Traffic descriptors for VBR video teleconferencing over ATM networks," *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 329–339, 1995.
[15] P. Koutsakis, "Dynamic vs. static traffic policing: A new approach for videoconference traffic over wireless cellular networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 9, pp. 1153–1166, 2009.
[16] G. R. Ash, *Traffic Engineering and QoS Optimization of Integrated Voice & Data Networks*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann, 2006.
[17] D. Marpe, T. Wiegand, and G. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," *IEEE Commun. Mag.*, vol. 44, no. 8, pp. 134–143, 2006.
[18] G. Van der Auwera, P. David, and 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 Trans. Broadcast.*, vol. 54, no. 3, pp. 698–718, 2008.
[19] [Online]. Available: http://trace.eas.asu.edu/h264/index.html
[20] M. Dai, Y. Zhang, and D. Loguinov, "A unified traffic model for MPEG-4 and H.264 video traces," *IEEE Trans. Multimedia*, vol. 11, no. 5, pp. 1010–1023, 2009.
[21] C.-F. Tsai, C.-J. Tsang, F.-C. Ren, and C.-M. Yen, "Adaptive radio resource allocation for downlink OFDMA/SDMA systems," in *Proc. IEEE ICC*, 2007.
[22] D. A. Dyson and Z. J. Haas, "A dynamic packet reservation multiple access scheme for wireless ATM," *Mobile Netw. Applicat. (MONET) J.*, vol. 4, no. 2, pp. 87–99, 1999.
[23] J. Greengrass, J. Evans, and A. C. Begen, "Not all packets are equal, part I: Streaming video coding and SLA requirements," *IEEE Internet Comput.*, vol. 13, no. 1, pp. 70–75, 2009.
[24] [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ ns537/ns705/ns827/white_paper_c11-520862.html
[25] J. Sotiropoulos, A. Delimargas, and P. Koutsakis, "Exploiting GoP patterns for video traffic control over wireless networks," in *Proc. IEEE WCNC*, 2012.
[26] O. Rose, "Simple and efficient models for variable bit rate MPEG video traffic," *Perform. Eval. J.*, vol. 30, no. 1-2, pp. 69–85, 1997.
[27] G. Chiruvolu, T. K. Das, R. Sankar, and N. Ranganathan, "A scene-based generalized Markov chain model for VBR video traffic," in *Proc. IEEE ICC*, 1998.
[28] [Online]. Available: http://www.cisco.com/en/US/tech/tk543/tk545/technologies_tech_note09186a00800a3a25.shtml
[29] [Online]. Available: http://blog.nielsen.com/nielsenwire/online_mobile/may-2012-top-u-s-online-video-sites/
[30] "Carrier-class edge routing," Motorola White Paper [Online]. Available: http://www.motorolasolutions.com/web/Business/Products/Cable%20 Broadband/EDGE%20Routers/BSR%2064000/_Documents/ Static%20 Files/Carrier-Class_Edge_Routing_New.pdf
[31] R. van Haalen and R. Malhotra, "Improving TCP performance with bufferless token bucket policing: A TCP friendly policer," in *Proc. 15th IEEE Workshop Local and Metropolitan Area Networks (LANMAN)*, 2007.
[32] M. Frey and S. Ngyuyen-Quang, "A gamma-based framework for modeling variable-rate video sources: The GOP GBAR model," *IEEE/ACM Trans. Netw.*, vol. 8, no. 6, 2000.
[33] P. Goudarzi, M. Hosseinpour, and F. Ayatollahi, "GoP-level distortion optimization for video quality enhancement in MANETs," in *Proc. 3rd Int. Congr. Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2011.
[34] A. Lazaris, P. Koutsakis, and M. Paterakis, "On modeling video traffic originating from multiplexed MPEG-4 videoconference streams," *Perform. Eval. J., Elsevier Publ.*, vol. 65, no. 1, pp. 51–70, 2008.
[35] S. H. Kang and A. Zakhor, "Effective bandwidth based scheduling for streaming media," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1139–1148, 2005.
[36] D. Mukherjee, E. Delfosse, J. G. Kim, and Y. Wang, "Optimal adaptation decision-taking for terminal and network quality-of-service," *IEEE Trans. Multimedia*, vol. 7, no. 3, pp. 454–462, 2005.
[37] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, Apr. 2008.
[38] T. Koponen, M. Casado, N. , J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, "Onix: A distributed control platform for large-scale production networks," in *Proc. 9th USENIX Conf. Operating Systems Design and Implementation (OSDI'10), USENIX Assoc.*, 2010.
[39] POX and NOX Control Platforms for Software Defined Networks [Online]. Available: http://www.noxrepo.org



Evangelos Maratsolas received his 5-year Diploma in Electronic and Computer Engineering from the Technical University of Crete, Greece in 2012. He is currently working in the telecommunications industry as a production manager. His research interests span the area of computer networks with emphasis on traffic engineering.

Polychronis Koutsakis (M'09–SM'13) received the 5-year Diploma in Electrical Engineering from the University of Patras, Greece, and the Ph.D. degree in Electronic and Computer Engineering from the Technical University of Crete, Greece. From July 2006 to December 2008, he was an Assistant Professor with the Electrical and Computer Engineering Department, McMaster University, Hamilton, ON, Canada. In January 2009, he joined the Department of Electronic and Computer Engineering, Technical University of Crete, as an Assistant Professor. Dr. Koutsakis has authored more than 90 peer-reviewed papers on the design, modeling, and performance evaluation of computer communication networks. He co-holds 1 US patent. He was honored twice, as 2012 and 2013 Exemplary Editor of the IEEE Communications Society for his work as Editor for the IEEE Communication Surveys and Tutorials Journal. He also serves as Area Editor for the Elsevier Computer Communications Journal.

Aggelos Lazaris received his 5-year Diploma and his M.Sc. degree in Electronic and Computer Engineering from the Technical University of Crete, Greece in 2006 and 2008, respectively. He is currently with the Department of Electrical Engineering at the University of Southern California where he is working towards his PhD degree. His research interests span the area of computer communication networks with emphasis on traffic engineering, modeling, and analysis. He is a USC Annenberg Graduate fellow, and a student member of IEEE.