Dynamic Video Rate Adaptation based on Pre-Congestion Notification

Steven Latré and Filip De Turck
Ghent University - IBBT - IBCN - Department of Information Technology
Gaston Crommenlaan 8/201, B-9050 Gent, Belgium
Telephone: +3293314940, Fax: +3293314899
e-mail: steven.latre@intec.ugent.be

***Abstract*—Multimedia services such as Video on Demand and network-based personal video recording introduce important new management challenges to network and service providers. Given the high revenue opportunities of these services, it is important to maximize the Quality of Experience (QoE) of multimedia services as much as possible. Traditionally, admission control mechanisms are used to protect the QoE of existing resources and to avoid that the traffic rate on a link exceeds a predefined threshold. Using admission control, flows are blocked when congestion is imminent. For video based services, the traffic rate can also be controlled by switching existing flows to a lower video quality. In this case, the videos can still be viewed but at a reduced QoE, which increases the available resources and thus makes room for new flows. In this paper, we focus on the video rate adaptation process. We propose a distributed video rate adaptation algorithm that allows controlling which qualities are offered to the users and how the videos are adapted as a response to changes in the network load. The video rate adaptation algorithm uses the information available in the Pre-Congestion Notification mechanism, a measurement based admission control mechanism standardized recently by the IETF. The video rate adaptation process is steered by utility functions, which define how the quality of the videos should be adapted as a function of the network load.**

I. Introduction

For non-scalable flows, such as constant data transfers, the most effective way of avoiding congestion is by deploying an admission control mechanism which blocks new flows once congestion is imminent. With the increasing popularity of video services, a new set of mechanisms have been proposed that rely on video rate adaptation to ensure that the traffic rate does not exceed a predefined threshold. Instead of blocking new requests, the quality of existing videos is decreased when the network load is high to make room for new requests.

Transport protocols that allow to adapt the bitrate of videos such as HTTP Live Streaming [1] and the recently standardized MPEG DASH [2] have already been proposed. The problem, however, is that the client and server decide on which rate to send out, as an end-to-end congestion control mechanism. Scalable Video Coding (SVC) allows changing the video rate by dropping video quality layers [3], but does not provide an algorithm to steer the rate adaptation. In practice, a service provider wants to choose the transmitted video rate as a function of the network load. Recently, mechanisms have been proposed which adapt the video rate on nodes inside the network based on in-network feedback [4]. However, they still require admission control mechanisms to block flows when even the lowest quality level cannot be allowed anymore. The challenge is to effectively combine video rate adaptation with traditional admission control mechanisms.

In this article, we propose a dynamic video rate adaptation algorithm for SVC on top of the Pre-Congestion Notification (PCN) mechanism [5], which is a decentralized measurement based admission control mechanism, recently standardized by the IETF. The PCN system can be deployed in a Diffserv domain: each PCN node measures the network load and signals this load information to the PCN endpoints through the marking of packets. Packets are marked in a similar way as performed by an Active Queue Management system such as Random Early Detection and Explicit Congestion Notification. An introduction to PCN, including a survey of algorithmic options for the various PCN functions is discussed in [6] and further evaluated in [7]. We have previously investigated PCN's performance in protecting video services and have proposed several extensions optimized for video services [8] and also proposed a static quality differentiation algorithm that only allows deciding upon the video quality to stream during the admittance phase. Furthermore, we derived guidelines for deploying PCN in a video environment in [9].

The novel dynamic video rate adaptation algorithm presented in this paper uses a PCN system to estimate the network load. Through policies a service provider can state the allowed video qualities as a function of the network load. The algorithm is dynamic as it is also possible to scale down the quality of existing flows.

The remainder of this article is structured as follows. In Section II, we provide an overview of the PCN mechanism. The dynamic video rate adaptation algorithm is presented in Section III. In Section IV, the impact on the admittance process and the perceived video quality is characterized. Finally, Section V, concludes this article.

II. PCN architecture for Video Rate Adaptation

A. IETF's PCN architecture

The goal of the PCN admission control system is to protect the QoS of inelastic flows in a Diffserv domain. In contrast to the traditional centralized Diffserv admission control system, the PCN system features a distributed measurement-based architecture. In the original PCN architecture, traffic enters

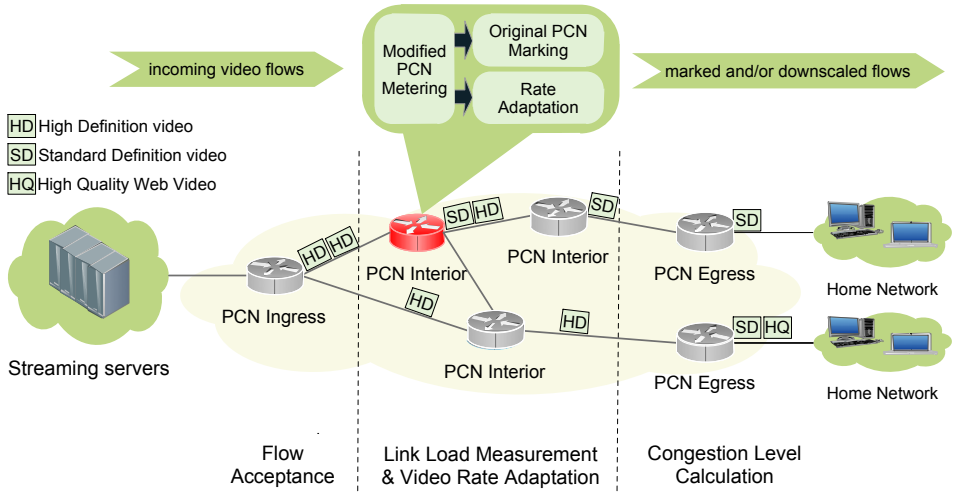

Fig. 1. Modified PCN architecture for supporting video rate adaptation.

the domain through PCN ingress nodes, and leaves through PCN egress nodes. PCN interior nodes meter the traffic rate and mark traversing packets as an in-band congestion signal (through Diffserv code points). The metering and marking functions support two marking behaviors: (i) threshold-metering and -marking, which marks all packets when the bit rate is greater than a reference rate, and (ii) excess-traffic-metering and -marking, where packets are marked at a ratio equal to the difference between the bit rate and the excess threshold rate. At the egress node, a congestion level estimate is calculated, by investigating the fraction of received marked packets, and signaled to the decision point (e.g., a PCN ingress node), where new flows are admitted and/or existing ones are terminated.

B. Extensions to the IETF's PCN architecture

In order to support the rate adaptation of videos, we propose two extensions to the metering and marking functions (e.g., in an interior node) of the IETF's PCN architecture as illustrated in Figure 1. First, the metering function must be able to characterize the traffic rate, which we call the Monitored Rate (MR), instead of only comparing the rate with a threshold. In previous work [8], we proposed such a metering algorithm based on sliding-window based bandwidth measurements. This metering algorithm can cope better with the bursty nature of video sessions and is more robust against changes in the traffic characteristics. The second modification is the inclusion of the video rate adaptation algorithm itself next to the metering and marking function. The video rate adaptation algorithm uses the MR value to locally re-scale the rate of SVC flows; its algorithmic details are discussed in the next section. Note that, as this algorithm only uses the metering function and does not mark packets, it is complementary to PCN's admission control functions. However, it does impact the measured load, which allows more sessions to be admitted at a reduced video quality.

III. Dynamic video rate adaptation

Ideally, an operator should stream the highest video quality possible if there is ample capacity left. Once the network load increases, existing streams can then be downscaled to make room for more flows, and thus more paying customers. The proposed algorithm enables an operator to set a policy on the number of flows that can be admitted for each video level, and dynamically scale the flows if the network load changes. This is done through utility functions $U_i(lo)$: these utility functions define the share of the capacity of the link that can be used for video quality level i as a function of the network load lo.

The algorithm is deployed on a PCN interior node and has two functions: a monitoring function and decision function. The monitoring function uses the local measurements MR of the modified PCN metering function and the admissible threshold rate AR. A normalized network load lo is calculated as follows: $lo = min(1, \frac{MR}{AR})$. A lo value of 1 indicates a pre-congested network, while a lo value of 0 represents a network without any load. Note that we use AR as the denominator of this fraction to eliminate the introduced headroom caused by the burstiness of the aggregate.

The calculated lo value is used by the decision function to decide how many and which flows need to be rescaled. Each time the monitoring function signals a change in the lo value, the utility functions $U_i(lo)$ are recalculated. For each existing video quality level, the number of flows that need to be (re-)scaled to this level are calculated based on the expected average bitrate of the video quality level i, through the following equation: $nrFlows = \lceil \frac{U_i(lo) \times MR}{getAverageBitrate(i)} \rceil$. This $nrFlows$ value provides the maximum number of flows that can be admitted at a given quality level. To decide which flows are mapped to which quality, the service provider can state its own policy (e.g., prioritizing gold subscriptions) by ordering the flows accordingly.

To ensure a stable output of the utility functions we smooth the output of the utility functions by calculating an exponential weighted moving average on the utility functions. This introduces a hysteresis on the decision function that avoids flows oscillating between video quality levels. The algorithm can be deployed on any PCN node: there is no cooperation needed between the PCN nodes. As each node will independently calculate their utility functions, multiple bottlenecks can cause an additional rescaling of a video further down the path if

the available load is lower there. As such, the videos will be rescaled to the utility functions of the bottleneck with the lowest available load.

Note that this approach assumes the use of homogeneous flows where all flows can be scaled to the same level. If this is not the case the algorithm can be extended by defining separate utility functions per flow type and defining an additional type of functions that estimate how the rescaling of a utility function will affect the total load. Hence, there is one estimation function per utility function. These estimation functions can then be used to compensate the lo value in scenarios where there is an unequal share between different flow types.

IV. Performance evaluation results

A. Experimental setup

The performance of the rate adaptation algorithm was evaluated in an NS-2 based simulator. A tree-based topology was used where a video head end streams SVC videos to 400 clients. This setup contains one bottleneck where the link capacity decreases from 1 Gbps to 500 Mbps. A PCN interior node was deployed on this bottleneck, together with the dynamic rate adaptation extension. The PCN system was configured to use only single marking (i.e., using only excess-traffic-metering and -marking). The AR value was set to 400 Mbps. We considered only flow admission and not termination. In the evaluated scenario, the request process was modeled through a uniform random distribution with a mean of 5 requests per second, a minimum of 2.5 and a maximum of 7.5, to represent a scenario describing realistic average load conditions. The simulation time was 100 seconds, each test was repeated 100 times. The weight of the exponential weighted moving average function was set to 0.95. Four video quality levels, with different average bitrates, were used: Full HD (11Mbps), HD ready (8Mbps), SD (2Mbps) and High Quality (HQ) Web (1.25Mbps). The following utility functions were used:

$$U_{\text{FHD}}(l) = \begin{cases} \min(1, \frac{l-s2}{s1-s2}) & \text{if } l \in [0, s2] \\ 0 & \text{if } l \in]s2, 1] \end{cases} \quad (1)$$

$$U_{\text{HDR}}(l) = \begin{cases} \max(0, \frac{l-s1}{s2-s1}) & \text{if } l \in [0, s2] \\ \max(0, \frac{l-s3}{s2-s3}) & \text{if } l \in]s2, 1] \end{cases} \quad (2)$$

$$U_{\text{SD}}(l) = 1 - U_{\text{FHD}}(l) - U_{HDR}(l) - U_{\text{HQW}}(l) \quad (3)$$

$$U_{\text{HQW}}(l) = \min(0, 1 - \sum_{n=0}^{24} e^{-\frac{x}{2}} \frac{x^n}{2^n \times n!}) \quad (4)$$

where $x = \frac{100 \times (l-s2)}{s1-s2}$ and $(s2, s3) = (s1 + 0.2, s1 + 0.3)$. Here, $s1$ denotes the load threshold at which the Full HD videos are transmitted to HD ready videos and is a parameter that is varied throughout the experiments. U_{HQW} denotes the cumulative Erlang distribution function with $\lambda = \frac{1}{2}$ and $k = 25$. The rationale behind the utility functions is that an increasing network load results in a linear rescaling from Full HD to SD. As we assumed that the lowest quality should be avoided we used an Erlang function between SD and HQ Web flows which has the property that, initially, the decrease of the SD utility function is less than the other utility functions, followed by a steeper increase of HQ Web later. Other functions with this property can be chosen as well.

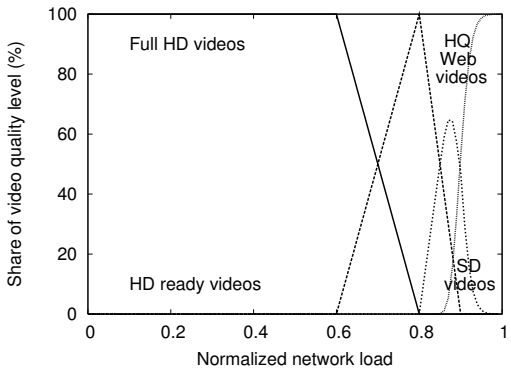

Fig. 2. Utility functions used. The rationale of the utility functions is to gradually lower the video quality as the network load increases to admit more connections at a reduced quality.

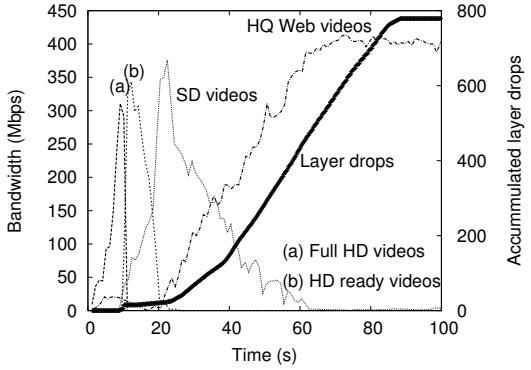

Fig. 3. Impact of the configured utility functions on the admission control in terms of bandwidth per quality level and number of quality switches.

To validate the algorithm we investigate the impact on the network load. In this experiment, $s1$ was set to 0.6 which resulted in the utility functions as shown in Figure 2. The effect these utility functions have on the bandwidth share per quality level is illustrated in Figure 3. Also the number of dropped layers is shown, illustrating the stability of the rate adaptation. The desired effect, defined through the utility functions, is reached, but with an additional burstiness in the network load, caused by the video burstiness. As shown in Figure 3, the rescaling decision is stable thanks to the introduced hysteresis: as the load increases, the number of layer drops gradually increases without any significant fluctuations. The share of Full HD and HD ready videos increases first but quickly drops as the network load increases further and all videos are scaled to HQ Web. At 88 seconds, we reach an equilibrium where new requests are blocked.

B. Obtained gain by the algorithm

We compare the performance of the dynamic rate adaptation algorithm with the original PCN mechanism (that does not feature video quality differentiation). Furthermore, we compare with an earlier proposed, static video quality differentiation mechanism [8] that can only scale a flow during the admittance phase. As performance metrics we use the number of admitted flows and the average video quality. The video quality was measured by using the Structural SIMilarity (SSIM) score [10],

TABLE I
COMPARISON OF THE DYNAMIC RATE ADAPTATION ALGORITHM WITH A STATIC VERSION AND THE ORIGINAL PCN MECHANISM

	Video quality	Admitted flows
original	0.981	46
static	0.905	112
dynamic	0.880	334

which is a video quality metric that produces a value between 0 and 1. The SSIM scores of the original videos ranged from 0.880 (HQ Web) to 0.981 (Full HD). For the dynamic algorithm we used the utility functions illustrated in Figure 2. In the original mechanism all flows were admitted as Full HD videos. The static case was configured to admit Full HD videos up to a load of 60%, HD ready videos up to a load of 80%, SD videos up to a load of 90% and HQ Web videos afterwards.

Table I shows the number of admitted flows and video quality at a network load of 100%. The original mechanism is able to provide the highest video quality as only Full HD videos are admitted, but with only a small number of admitted flows. The static mechanism can already admit more videos but the highest gain is obtained by using the dynamic mechanism, because all videos are eventually scaled to HQ Web videos. In the dynamic mechanism, the obtained video quality will be higher than the static version during lower network loads, as we can admit a flow initially at a high quality and downscale it afterwards.

C. Analysis of the impact on the video quality

The definition of the utility functions has an important impact on the admission control. In this experiment, we investigate the impact of varying the $s1$ parameter and present an analytical estimation for the number of admitted flows and average SSIM score. In the dynamic algorithm, the estimated number of admitted flows of video quality level i for a non-normalized load l is

$$\frac{U_i(\frac{l}{AR}) \times l}{\text{Bitrate}_i} \tag{5}$$

where Bitrate_i is the expected average bitrate of video quality level i (e.g., given as meta-data by the video server). The total number of admitted flows is then given by summing all n quality levels: $\sum_{i=1}^{n} \frac{U_i(\frac{l}{AR}) \times l}{\text{Bitrate}_i}$.

To calculate the average video quality, it is sufficient to know the share of each video quality level. The average video quality at load l is given by $\sum_{i=1}^{n} U_i(l) \times \text{SSIM}(i)$, where $\text{SSIM}(i)$ is the average SSIM score of video quality level i. To obtain the average SSIM score over a timeframe $[t_1, t_2]$, the equation can be integrated:

$$\frac{\int_{t_1}^{t_2} \sum_{i=1}^{n} U_i(L(t)) \times SSIM(i) \times tdt}{\int_{t_1}^{t_2} \sum_{i=1}^{n} U_i(L(t)) \times tdt} \tag{6}$$

where $L(t)$ is an estimator for the expected load over time t.

Figure 4 shows the average SSIM score as a function of the network load for different $s1$ values. The figure shows that an increasing network load has a decreasing effect on the SSIM score as more videos are being rescaled to HQ Web videos. This decreasing trend starts sooner for lower $s1$ values. The results also show that the analytical model (ana) is a very good estimator for the results obtained through simulation (sim).

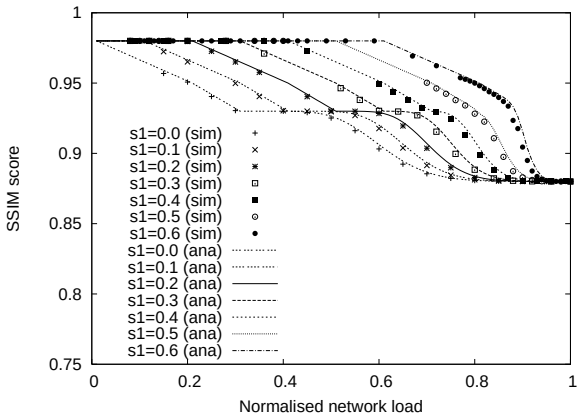

Fig. 4. SSIM score as a function of the network load for different $s1$ values.

As the obtained average video quality can be characterized analytically, this can be used to determine an adequate policy by setting appropriate values for the utility functions. An operator should set the values that represent the lowest possible video quality scores he is willing to tolerate for different network loads. Based on these values he can then determine which $s1$ parameter value, or even other types of utility functions, are most suited to obtain the expected behaviour.

V. CONCLUSIONS

We presented a video rate adaptation algorithm that augments a PCN system with the option to dynamically modify the video quality of existing flows. The algorithm uses utility functions that define the share of each video quality level as a function of the measured load. Performance evaluation results show that the algorithm indeed succeeds in scaling the videos according to the utility functions and characterizes the gain compared to other approaches. Furthermore, we proposed an analytical model that allows finding suitable utility functions.

REFERENCES

[1] R. Pantos and W. May, "HTTP Live Streaming," 2010. [Online]. Available: http://tools.ietf.org/html/draft-pantos-http-live-streaming-04
[2] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE Multimedia*, vol. 18, pp. 62–67, 2011.
[3] B. Libaek and O. Kure, "Congestion control for scalable VBR video with packet pair assistance," in *Proceedings of 17th International Conference on Computer Communications and Networks, 2008. ICCCN '08.*, 2008.
[4] C. Liu *et al.*, "Advanced rate adaption for unicast streaming of scalable video," in *Communications (ICC), 2010 IEEE International Conference on*, 23-27 2010, pp. 1 –5.
[5] P. Eardley, "Pre-Congestion Notification (PCN) Architecture," RFC 5559 (Informational), Jun. 2009.
[6] M. Menth *et al.*, "A survey of pcn-based admission control and flow termination," *Communications Surveys Tutorials, IEEE*, vol. 12, no. 3, pp. 357 –375, 2010.
[7] M. Menth and F. Lehrieder, "Pcn-based measured rate termination," *Computer Networks*, vol. 54, no. 13, pp. 2099 – 2116, 2010.
[8] S. Latré *et al.*, "PCN based admission control for autonomic video quality differentiation: Design and evaluation," *Journal of Network and Systems Management*, 2010, accepted for publication.
[9] S. Latré and, K. Roobroeck, T. Wauters, and F. Turck, "Protecting video service quality in multimedia access networks through pcn," *Communications Magazine, IEEE*, vol. 49, no. 12, pp. 94 –101, december 2011.
[10] Z. Wang *et al.*, "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, February 2004.