

# Protecting Video Service Quality in Multimedia Access Networks through PCN

Steven Latré, Klaas Roobroeck, Tim Wauters, and Filip De Turck \*Ghent University - IBBT - Department of Information Technology, e-mail: steven.latre@intec.ugent.be

**Abstract**—The growing popularity of video-based services, and their corresponding unpredictable bursty behavior, makes the design of an admission control system an important research challenge. The Pre-Congestion Notification (PCN) mechanism is a measurement-based approach, recently standardized by the IETF, and optimized towards the admission of inelastic flows, where the number of flows is sufficiently large so that individual bursts of flows can be compensated by silence periods of others. In this article, we discuss the implications of applying PCN to protect video services, which have a less predictable behavior. Several algorithms for protecting video services in multimedia access networks are described. Through performance evaluation, we show the impact of these algorithms on the network utilization and video quality, and present guidelines on how to configure a PCN system.

## I. INTRODUCTION

Thanks to the growing popularity of Triple Play services such as Video on Demand (VoD), video has been given the lion's share of bandwidth consumption in today's access networks. Network providers have dimensioned their networks to scale with this growing demand for bandwidth, but this itself does not protect access networks against packet loss. Even small amounts of packet loss can completely deteriorate the quality as perceived by the end user, defined as the Quality of Experience (QoE). It is therefore crucial to protect the delivery of video services against network congestion.

Such protection is traditionally done through admission control mechanisms that block new requests that would violate the total supported bandwidth consumption. Centralized admission control mechanisms such as the one in TISPAN [1] receive a request for resources with a Quality of Service (QoS) guarantee which they can grant or deny. However, these centralized solutions have difficulties in keeping the available resource information up to date. To tackle this, decentralized admission control mechanisms such as the one in Intserv [2] have been proposed. The problem is that these solutions require a detailed description of every resource.

Measurement-Based Admission Control (MBAC) mechanisms omit the need for detailed traffic descriptors. Instead, admission control is performed by measuring the total bandwidth consumption. The IETF has started a working group that standardizes a promising distributed MBAC mechanism called Pre-Congestion Notification (PCN). The PCN admission control function performs its decisions based on in-network feedback signalled through packet marking. A survey of the PCN algorithms used for flow admission and flow termination is presented in [3]. While the PCN system seems promising

to protect video services, there are some challenges to be investigated. The original PCN mechanism is intended to protect inelastic flows with a known maximum peak rate, which is not the case for video services, which feature bursty traffic. While PCN's performance evaluation carried out in the PCN Working Group [4], [5] also investigated the protection of broadband connections, the investigated connections had a moderate burstiness, while the bandwidth of videos is known to be very bursty.

In this article, we provide an answer to the following research questions: (1) Can PCN be applied to protect video services? (2) What is the impact of applying the traditional PCN algorithm in terms of QoE and network utilization? and (3) Can PCN be optimized to specifically protect video services?

The remainder of this article is structured as follows. In Section II, we provide an introduction to the original PCN mechanism; in Section III, we discuss the impact of the original PCN mechanism on the network utilization and QoE. The modifications to the original PCN mechanism and their gain are proposed in Section IV. In Section V, we present best practice configuration guidelines.

## II. TRADITIONAL PRE-CONGESTION NOTIFICATION

The PCN mechanism builds further upon the ideas of the Explicit Congestion Notification (ECN) protocol, defined in RFC3168 [6], and the re-ECN mechanism<sup>1</sup>, which both use packet marking to signal congestion. ECN marks packets to signal congestion to its receivers, while re-ECN re-inserts these congestion signals to reveal the information about congestion to other nodes in the network. Similarly, PCN starts marking packets when congestion is imminent, as a trigger for an admission control decision.

The PCN architecture, as specified in RFC 5559, is illustrated in Figure 1. As shown, all traffic enters a PCN domain through a PCN ingress node and leaves the domain through PCN egress nodes. Inside a PCN domain, packets are subject to metering and marking. Based on this metering and marking, a congestion assessment is performed, which allows admitting or blocking new flows or terminating existing ones. A PCN system can choose to support flow admission, flow termination or both.

<sup>1</sup>draft-briscoe-tsvwg-re-ecn-tcp-09

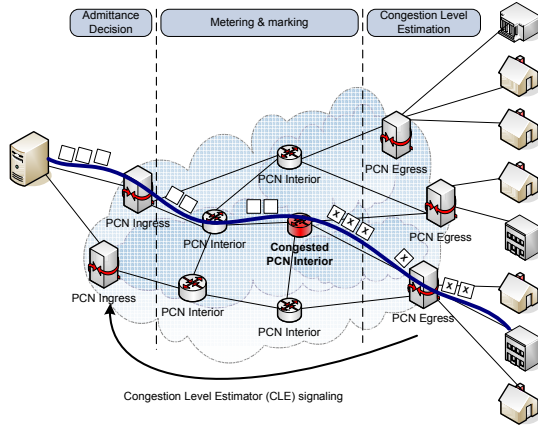


Fig. 1. Overview of the traditional PCN architecture comprising three node types: PCN ingress, PCN interior and PCN egress.

### A. Metering and marking function

The metering and marking function is deployed on the ingress and interior nodes. For flow admission, an admissible rate  $AR(l)$  on each link  $l$  of the PCN domain is defined. Similarly, for flow termination, a sustainable aggregate rate  $SAR(l)$  is defined. As packets traverse through the metering and marking function, the PCN traffic rate is metered and compared to  $AR$ ,  $SAR$  or both. Packets are correspondingly marked if they exceed the configured rates. 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. The metering and marking behavior is standardized in RFC5670, together with a first encoding approach that allows defining two states in RFC5696. Currently, a draft is under consideration that allows encoding three states <sup>2</sup>.

In the traditional PCN algorithm, the metering function is performed by a token bucket-based algorithm: tokens are added to the bucket at the configured rate. Every time a packet arrives at the PCN interior node, the status of the token bucket is investigated and  $b$  tokens are removed from the bucket, where  $b$  is the size of the packet in bits. The token bucket marks packets if the number of tokens decreases below a threshold. Note that the token bucket algorithm allows to average the measurements through its depth: the larger the depth, the longer it will take for a full token bucket to reach the token bucket threshold. As all parameters are in bits, the timeframe at which is averaged depends on the micro-level information such as the packet inter-arrival time. A large burst of consecutive packets, followed by a silent period, can lead to full bucket depletion: this would not happen if the packets are equally spread over the same timeframe.

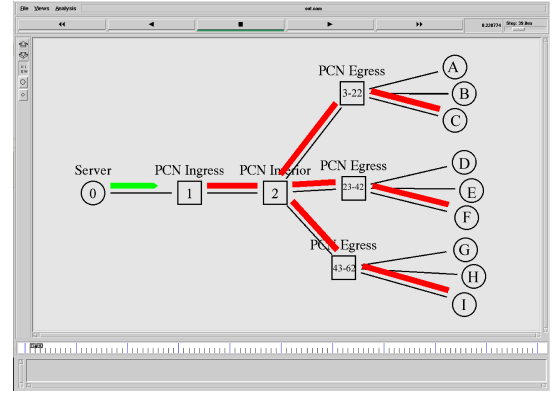


Fig. 2. Screenshot of a run of the modified NS-2 simulator, highlighting the network topology used for the experiments. A red packet represents a marked packet.

### B. Congestion assessment function

At the edge of the PCN domain, the PCN egress nodes investigate the packets that traverse and check whether or not they are marked. Per ingress-egress pair, a Congestion Level Estimator (CLE) is calculated. This CLE represents the ratio of marked to total PCN-traffic. One single CLE is calculated per ingress-egress pair. The behavior at the PCN edge is still under investigation; two modes are currently defined: a single marking mode <sup>3</sup> (using excess-traffic-metering and -marking) and a controlled load mode <sup>4</sup> (using both excess-traffic-metering and threshold-metering and -marking). The CLE is continuously calculated and reported to the decision point. Both modes support CLE report suppression that allows only reporting the CLE after a configurable time-out or if the CLE changes significantly.

### C. Decision point

The decision point receives regular CLE updates from the egress nodes and performs flow admission or flow termination depending on the received CLE. If the decision point is located on the PCN ingress node, a signaling mechanism such as SIP or RSVP is assumed that performs the request for the flow. The decision point includes a timer-based mechanism that detects missing boundary node reports: when the timer is exceeded, an alarm is raised to the management system and the decision point ceases to admit new flows.

## III. IMPLICATIONS FOR VIDEO-BASED SERVICES

### A. Experimental set-up

In order to characterize the impact of applying PCN to videos, we have evaluated PCN's performance through the NS-2 simulator. Figure 2 illustrates the simulated network topology. As shown, the topology models a tree-based access network where a video server is streaming to a set of 1000+ active clients. We introduced a bottleneck on the PCN ingress node, where the link rate dropped from 2Gbps to 1Gbps. No other bottlenecks existed in the network. We modelled a

<sup>2</sup>draft-ietf-pcn-3-in-1-encoding-08

<sup>3</sup>draft-ietf-pcn-sm-edge-behaviour-06

<sup>4</sup>draft-ietf-pcn-cl-edge-behaviour-09

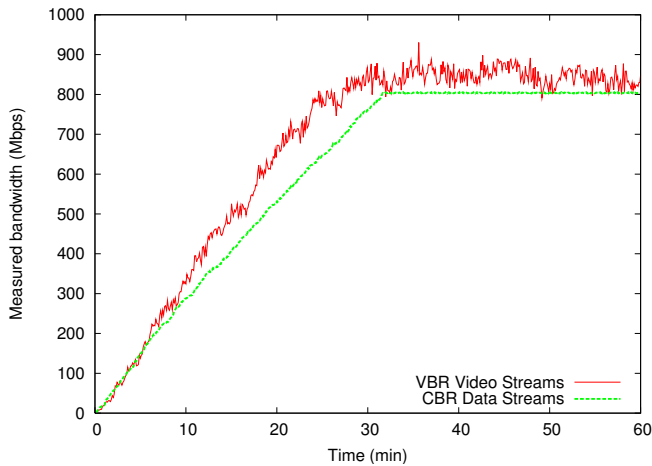


Fig. 3. Evolution of the measured bandwidth over time for a PCN system protecting CBR data streams and VBR video streams, respectively.

scenario where the clients request the transmission of videos from the server. The requests were generated based on the request production trace of the VoD service of a leading European telecom operator. The simulation time was set to 1 hour and during this timeframe, 1171 videos were requested. The highest request rate observed was 5 requests per second for all clients together.

### B. Results description

For this experiment, two stream types were used: a Constant Bit Rate (CBR) data stream with an average bitrate of 2.5Mbps and a Variable Bit Rate (VBR) video stream which bitrate also averaged around 2.5Mbps. In both the CBR and VBR case, PCN's admissible rate parameter of the token bucket at the PCN ingress node was set to 800Mbps. The token bucket was configured differently for both cases: in the CBR case, the token bucket depth was set to 500,000 bits, while in the VBR case the token bucket depth was configured at 16,000,000 bits. Our PCN implementation uses the single marking mode as specified in the latest draft<sup>5</sup>, supporting only flow admission and including the CLE report suppression option with the *T-maxsuppress* timer set to 4.5 seconds and the *CLE-reporting-threshold* set to 0.5.

Figure 3 illustrates the evolution of the measured bandwidth over time for both CBR and VBR. As illustrated, there is a significant difference in measured bandwidth in both cases. The predictable behavior of the CBR data streams results in a constant bandwidth of exactly 800Mbps. The individual VBR video streams on the other hand, are far more bursty and therefore less predictable. Additionally, the measured bandwidth does not stop before the 800Mbps mark of the admissible rate parameter. Instead, the PCN system keeps admitting sessions until *all measurements* in PCN's metering function are above the 800 Mbps threshold. While in the CBR case, the admissible rate acts as an upper bound for the measured bandwidth values when new flows are blocked,

in the VBR case, this rate is more a lower bound. This significantly complicates the configuration.

The above results highlight three major implications of using a PCN system to protect video services. We refer to [7] for an in depth study of the impact of the original PCN parameters on the performance.

- 1) *The burstiness of the network aggregate has a destructive effect on the network utilization:* In the CBR case, we can easily increase the admissible rate parameter to be equal to the link capacity. For VBR, this is not possible because the burstiness of the aggregate requires additional headroom above the admissible rate parameter to avoid congestion. This additional headroom should be equal to the aggregate variability; we define the aggregate variability as the difference between the maximum and minimum traffic rate that can be measured in a marked state. For the CBR case, the aggregate variability is almost zero and thus negligible. For the VBR case, the aggregate variability depends on the encoding settings of the individual videos and is hard to characterise off-line.
- 2) *The VBR case requires an alternative token bucket configuration:* As the variation in bandwidth measurements is higher for VBR, a larger time window is needed to produce stable measurement results. This requires a higher bucket depth to allow for more required packets to drop below the token bucket threshold. Note that, as the token bucket algorithm is a packet-based algorithm, the translation from a good timeframe to a good token bucket configuration is not straightforward. For the protection of video connections, the bucket depth must be considerably larger to ensure that the transmission of one video frame cannot lead to the depletion of the bucket. This means that the token bucket depth must be at least equal to the maximum possible video frame size. In practice, the depth should be a factor higher as the transmission of multiple videos can cause a burst of video frames. In our tests, a depth of 16,000,000 bits showed to be a good value for video.
- 3) *The increased token bucket depth introduces a potential delay issue:* The higher the bucket depth, the longer it takes to reach the bucket's threshold, thus the higher the measurement delay is. While a larger bucket depth can cope with a higher burstiness, it introduces a delay. Increasing the token bucket depth thus comes with the cost of an added delay in marking. This may lead to an overshoot in admitted sessions if the request rate is high. Furthermore, there is a considerable variation in the theoretical measurement delay of a token bucket. As indicated in [7], the theoretical delay of a token bucket system as used in Figure 3 is between 6.6ms and 1,000,000 seconds. Experiments have shown that the mean delay is approximately 180ms.

## IV. PCN ARCHITECTURE FOR PROTECTING VIDEOS

Figure 4 provides an overview of our architecture that tackles the three issues discussed in the previous section.

<sup>5</sup>draft-ietf-pcn-sm-edge-behaviour-06

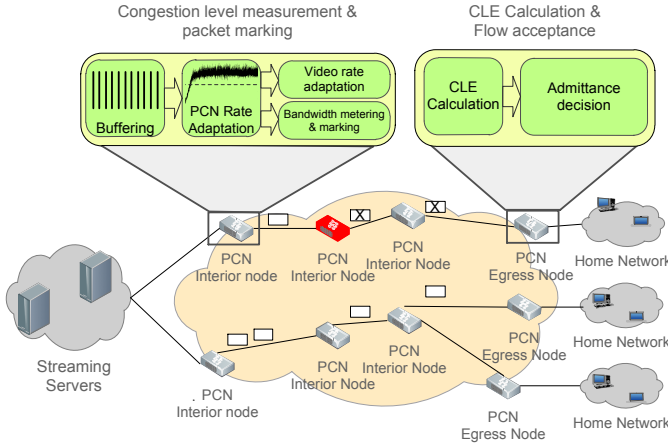


Fig. 4. Overview of the PCN-based admission control system, specifically optimized towards the protection of video services in access networks.

We propose three distinct modifications to the original PCN system: (1) an adaptive algorithm for configuring PCN’s metering rate (2) a buffer mechanism that reduces the aggregate variability and (3) a video rate adaptation algorithm.

#### A. Automatic PCN rate configuration

As discussed in Section III, a suitable configuration for the rate in PCN’s metering algorithm should take into account the necessary headroom. The goal is to find the highest rate value that still avoids any congested related losses. As this depends on the aggregate variability, we propose an adaptive algorithm that continuously monitors this variability and configures the rate accordingly. As metering algorithm, we propose a sliding-window-based bandwidth metering algorithm instead of the traditional token bucket. The sliding-window-based metering algorithm works as follows: it keeps a record of the size of the packets received during the last measurement window  $mw$ . Every time a new packet is received, it is added to the window and packets received earlier than  $mw$  are removed. The bandwidth can then easily be calculated by

$$\frac{\sum_{i \in W} s_i}{mw}$$

where  $s_i$  is the size of the  $i$ th packet in the window  $W$ . Once the bandwidth has been calculated, the metering algorithm compares the calculated bandwidth with the configured rate to decide whether to mark the packet or not.

This alternate, time-based, metering algorithm overcomes the limitations of the packet-based token bucket, and is therefore more suitable for video. The algorithm serves as an alternative to the standard PCN metering algorithm and can thus be deployed on the PCN ingress or interior node. The marking behavior of the original PCN system is not modified: only the way traffic is metered is modified. The increased accuracy of the sliding-window based metering algorithm comes with the cost of an increased memory requirement. The traditional token bucket is in essence a bit counter and only requires the storage of this counter to maintain the state of the monitoring algorithm. In contrast, the sliding-windows-based metering algorithm requires the storage of the size of

the packets received during the last measurement window  $mw$  and their timeframe. As such, the required memory is linearly proportional with  $mw$ . The exact value also depends on the number of packets that can traverse through the monitoring function, and thus the bottleneck link attached to the monitoring function. For example, with a bottleneck link of 1Gbps, a  $mw$  of 160ms and packets of 1500 bytes, the corresponding required memory is 53.2kB, as at most 13,333 packets can traverse during the 160ms timeframe. In terms of computational complexity, both algorithms run in constant time.

The automatic PCN rate algorithm continuously estimates the aggregate variability by measuring the maximum and average bandwidth and calculating the estimated variability value as:  $\text{variability} \equiv 2 \times (\text{MaxBW} - \text{AvgBW})$ . This value is used for setting the rate of the metering algorithm. A network provider typically wants to set a goal rate the link bandwidth may not exceed. This goal rate is then used for setting the rate parameter of the metering algorithm, by subtracting the goal rate with the estimated variability value. As such, the automatic rate configuration algorithm allows a network provider to configure a PCN system again through the goal rate value. For more details about this algorithm, we refer to [8].

Figure 5 shows the combined performance of the modifications to the original PCN mechanism in terms of number of admitted sessions and QoE. As an estimation of the QoE, we used the Structural Similarity Score (SSIM) [9], which is an objective video quality metric, where a SSIM score above 0.9 typically means videos without any visual artefacts and a SSIM score below 0.7 denotes a barely watchable video. All experiments were repeated 100 times. Various experiments were conducted, each with an alternate encoding of the video content ranging from a set of constant bit rate videos to a set of constant quality videos. To show the impact of various encoding settings on PCN’s performance, we present the 25th percentile, mean and 75th percentile of these aggregate variability values. For the traditional PCN algorithm, we used the identical settings as those presented for the VBR case in Section III-A.

In the mean case, no significant changes can be observed between the original PCN mechanism and the one augmented with the adaptive PCN rate algorithm. However, the original PCN mechanism fails in the other two cases. In the 25th percentile, the PCN mechanism admits only 76 sessions while the adaptive PCN rate algorithm admits up to 82 sessions without affecting the QoE. The network is thus underutilized as at least 7.3% more resources could have been admitted. On the other hand, in the 75th percentile, the high variability leads to over admission and thus congestion related packet loss, which in turn leads to severe visual distortions in the QoE. The adaptive PCN rate algorithm is able to cope with these changes in variability, caused by different video encoding settings, by adapting PCN’s AR parameter, and thus allows optimizing the network utilization, while avoiding QoE deterioration.



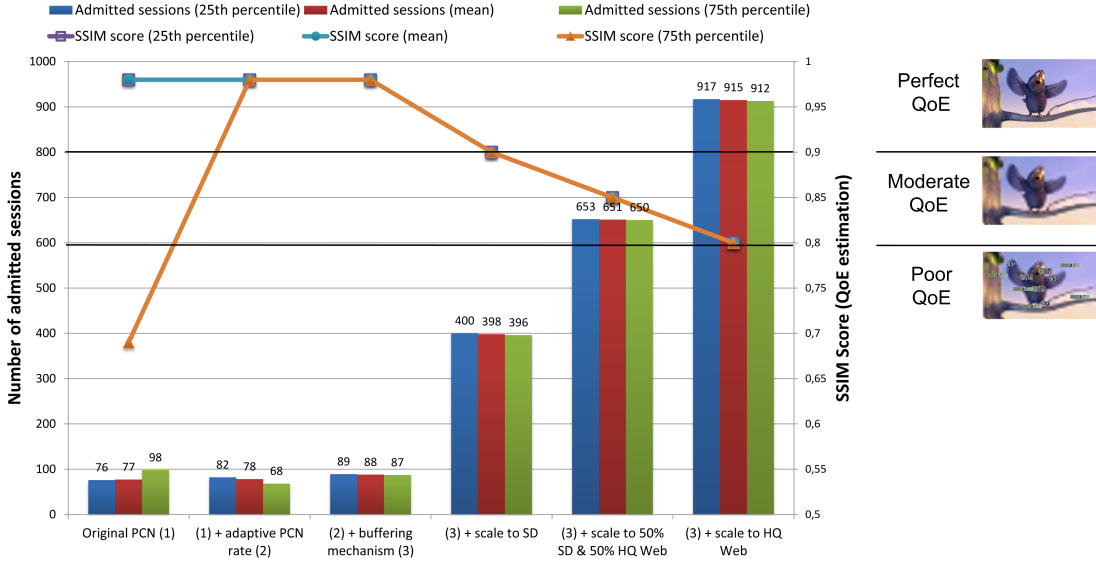


Fig. 5. Impact of the presented PCN-based system on the number of admitted sessions and QoE, estimated through the SSIM score.

### B. Buffering mechanism

While the automatic PCN rate configuration algorithm successfully changes the rate parameter to cope with the necessary headroom, the headroom itself can still be regarded as a waste in network utilization. The higher the aggregate variability is, the lower PCN's rate is configured and thus the fewer sessions are admitted. In our architecture, we apply a buffering step just before PCN's metering function to reduce the required headroom. As such, the buffering mechanism can be deployed both on the PCN ingress and interior nodes; the mechanism does not interfere with PCN's marking function. The buffering mechanism works as follows: it calculates a threshold rate  $T$ , where  $T = \frac{G+AR}{2}$ , with  $G$  and  $AR$  the network providers's goal rate and PCN's configured rate, respectively. This  $T$  is then used to construct a buffer that can support a burst rate of  $T$  during  $mw$ . Packets from this buffer are served to a weighted fair queueing scheduling component with a weight that is equal to  $\frac{T}{G}$ . If packets get dropped from this buffer, they are served to a weighted fair queueing scheduling component with a weight that is equal to  $1 - \frac{T}{G}$ . Buffering the data comes of course with a memory requirement, which is linearly proportional to the size of the buffer. In the buffering mechanism, this buffer size is determined based on the buffer threshold rate  $T$ . As  $T$  can evolve over time, so can the required buffer size. However,  $T$  will never be higher than  $AR$ , which allows defining an upper bound. In our experiments, the maximum required buffer size of the mechanism was therefore 20MB, as the buffer threshold rate supported at most a rate of 1Gbps during the 160ms measurement window. The buffering has an impact on the delay experienced by the customers, which makes it less applicable to interactive services such as videophony. However, as the delay is limited (i.e., less than 2 seconds), increasing the play-out buffer at the customer with a few seconds can overcome this limitation for non-interactive scenarios such as a VoD system where an increase in start-up delay of 2 seconds is tolerable.

The impact of this mechanism is illustrated in Figure 5. The buffering mechanism periodically monitors PCN's configured rate parameter, and adjusts  $T$  accordingly. The buffering mechanism has the following effect on the admission process: the outgoing traffic is smoothed as the peaks in bandwidth are avoided by sending the bursts of packets out at a slower rate than the link capacity. This decreases the variability of the traffic, which in turns triggers the adaptive PCN rate algorithm discussed in Section IV.A to increase the rate of the metering algorithm. As a result, more sessions will be admitted. Because the network is better utilized, as silence periods are filled with peak periods, more sessions can be admitted without deteriorating the QoE. Of course, the buffer cannot completely smooth out the traffic, which leads both algorithms to converge to a similar rate. In our experiments, the aggregate variability was decreased to approximately 30Mbps, and PCN's configured rate converged to 965Mbps: this resulted in an increase in network utilization from 78 to 88 (11%) admitted sessions and no QoE drop.

### C. Dynamic video rate adaptation

For videos, a viable alternative to blocking connections is adjusting the video rate. Scalable Video Coding (SVC) is a video coding standard that encodes video in multiple quality layers. By simply dropping a layer, the video quality, and consequently the video rate, can be decreased, making room for additional flows at a reduced QoE.

We designed a video rate adaptation algorithm that uses the PCN architecture to avoid congestion by dynamically down-scaling existing SVC videos. While several rate adaptation algorithms exist (e.g., as part of HTTP adaptive streaming), the decisions in these algorithms are triggered by clients. Therefore, they are more suited for an Over-The-Top video scenario. The novelty of our approach is that we allow a network provider to manage the QoE in its network directly through policies for use in a managed network.

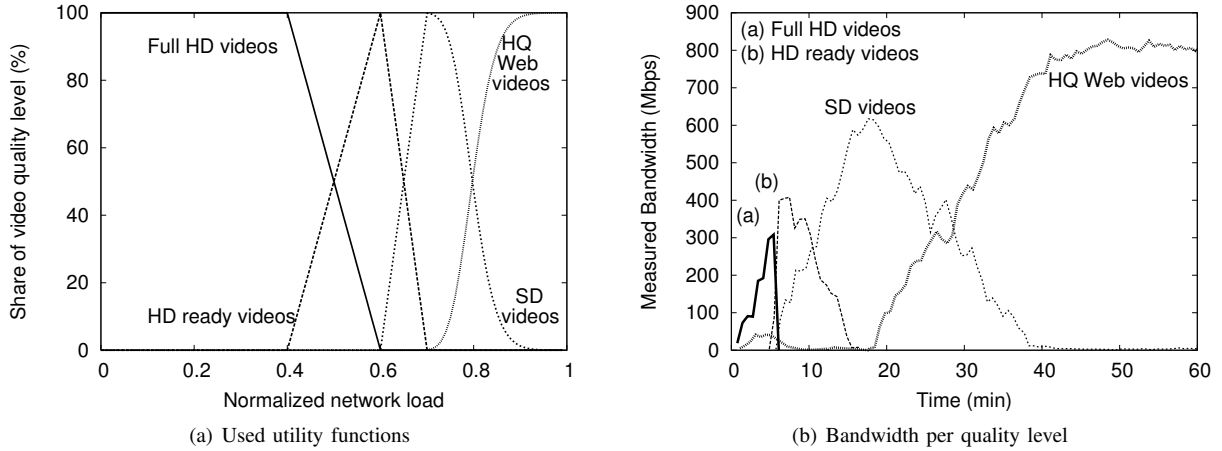


Fig. 6. An example of possible utility functions (a) in the video rate adaptation algorithm and their impact on the admission control (b)

The goal of the algorithm is to decide if and how a particular video needs to be downscaled. This downscaling occurs on each PCN ingress or interior node locally. The video rate adaptation algorithm uses the measurement information from PCN’s bandwidth metering to assess the current network load. Based on this load assessment and a policy of the network provider, the algorithm decides upon the allowed share of each video quality level and scales the videos accordingly. These policies are configured through utility functions, which define the share of each video quality level as a function of the network load. Hence, there is one utility function per quality level per PCN node. By simply calculating the value of the utility functions for a given network load, measured through PCN’s bandwidth metering, we can obtain the required share of each video quality level. Note that the current objective of the proposed utility functions is to control the QoE level that is provided under varying load conditions. Other objectives such as enforcing fairness between heterogeneous devices can be investigated as well: this is part of future work.

As the adaptation occurs locally, there is no additional signalling needed between PCN nodes except the traditional marking of packets for flow admission. Therefore, the rate adaptation algorithm does not modify the marking behavior of the original PCN architecture. Besides storing the static utility functions, the rate adaptation algorithm does not have a significant memory requirement. The main computational requirement is due to the dropping of quality layers of the SVC video. However, in the most common SVC implementations, each SVC layer is encapsulated in its own RTP stream. Dropping a layer can be achieved easily by dropping the corresponding RTP stream [10].

To evaluate this algorithm, we allowed to downscale the videos to multiple quality levels depending on the network load, each with a different video rate: Full HD (11Mbps), HD ready (8Mbps), SD (2.5Mbps) and High Quality (HQ) Web (1.1Mbps). Figure 6(a) illustrates these utility functions as a function of the time. As can be seen, the rationale is that first Full HD videos are admitted but that they are gradually scaled to lower video quality levels until all videos are HQ Web. Figure 6(b), shows the share of each video quality over

time. As more requests arrive and the network load obviously increases, the desired behavior, defined through the utility functions is reached.

Figure 5 shows that, if a network provider is willing to also decrease the video’s QoE, a bigger improvement in network utilization can be achieved. If we scale the existing videos only to SD videos, we are able to admit 4 times more sessions, while the SSIM score drops from 0.98 to 0.90. Scaling to HQ Web videos results in 915 admitted sessions and a SSIM score of 0.80. These SSIM scores correspond to videos that are perceived as moderate to very good, having no visual distortions. As illustrated in Figure 5 and 6(b), the utility functions can be used to properly tune the QoE score end-users receive. The difference with the observed decrease in QoE in the performance of the original PCN algorithm is two-fold. First, the drop in QoE is much lower than that of the original PCN mechanism, since we avoid visual distortions. Second, we are able to estimate and control this QoE drop, by selecting the appropriate utility functions. The operator should first define the minimum QoE level he wants to support and define its utility functions accordingly. In the next section, we present a QoE estimation function that can be used to estimate the effect of the utility functions on the QoE.

## V. PCN CONFIGURATION GUIDELINES

In this section, we provide guidelines to configure an operational PCN enabled system that can protect video services.

- 1) *Select the metering algorithm:* The packet-based token bucket algorithm has some important implications when used for video, which makes the configuration dependant on the flow’s micro-level behavior. Therefore, for video services, it is better to use the sliding-window-based algorithm, which is less sensitive to individual bursts and easier to configure.
- 2) *Configure the metering algorithm:* With the introduction of the automatic PCN rate algorithm, only the  $mw$  parameter needs to be configured. As  $mw$  is linearly proportional with the detection delay of the PCN system, it is crucial to find a balance between a measurement window which provides stable results and minimizes the

delay. A good choice for  $mw$  is the lowest value that still provides stable results for the measurement of the difference between the maximum and minimum bandwidth measurements. For example, a  $mw$  of 50ms might theoretically lead to a fast detection, but in practice the measurement output will continuously oscillate, as the individual bursts of packets can be longer. On the other hand, a  $mw$  of 1sec will result in a stable measurement output but cannot protect the network against a flash crowd. For video, experiments showed that a good value is between 150ms and 250ms.

- 3) *Set the goal rate:* The provider's goal rate depends on the delay of the metering algorithm. The delay of the metering algorithm is equal to the size of  $mw$ . To identify the maximum potential overshoot it should be multiplied by the maximum possible request rate and the expected average bitrate of a flow. The provider should limit the maximum number of requests that can arrive in a PCN system and decrease the goal rate to cope with this overshoot. For example, for a maximum request rate of 100 requests per second, the admission of 2Mbps video sessions and a  $mw$  of 250ms, the overshoot is  $2 \times 0.250 \times 100 \equiv 50$ . This means that a goal rate of 1Gbps should be lowered to 950Mbps.
- 4) *Define the appropriate utility functions:* To construct the appropriate utility functions, a network provider should know the minimum level of QoE that is offered to the customers, averaged between all clients. This can easily be calculated through the utility functions by multiplying the value of each utility function, at a load of 1, with an expected SSIM score, i.e. for  $n$  video quality levels and  $U_i$  utility functions:  $\sum_{i=0}^n U_i(1.0) \times SSIM(i)$ . For example, to find the QoE level offered when flows are downscaled to 50% SD videos and 50% HQ Web videos, the SSIM score is  $0.5 \times 0.9 + 0.5 \times 0.8 \equiv 0.85$ . As the impact of the utility functions on the SSIM score can be calculated, the network providers can use the desired SSIM score to select the appropriate utility functions.

## VI. CONCLUSIONS

In this article, we proposed several modifications to the IETF's original PCN algorithm. The modifications are not intended to replace the original PCN algorithm but can be used to significantly improve the network utilization for video services. They provide mechanisms to control the burstiness of the network aggregate and automate the configuration of the most critical PCN parameters. Moreover, the combination of PCN with a video rate adaptation algorithm was presented, which allows defining how videos are scaled based on PCN measurements. The performance evaluation investigated the impact of each proposed modification on the network utilization and showed that, without performing any downscaling of the video, an increase of 17% in network utilization can be obtained. If this is combined with the downscaling of videos, the increase in network utilization is a multitude of this value. By providing configuration guidelines, we detailed how to configure an operational PCN system, for protecting video services.

## ACKNOWLEDGMENT

Steven Latré and Tim Wauters are funded by the Fund for Scientific Research Flanders (FWO-Vlaanderen). The research was performed partially within the framework of the EUREKA CELTIC RUBENS project and FP7 STREP project OCEAN (under grant agreement no 248775).

## REFERENCES

- [1] ETSI TS 182 019, "Resource and Admission Control Sub-system (RACS); Function Architecture," 2009.
- [2] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633 (Informational), Internet Engineering Task Force, Jun. 1994. [Online]. Available: <http://www.ietf.org/rfc/rfc1633.txt>
- [3] M. Menth, F. Lehrieder, B. Briscoe, P. Eardley, T. Moncaster, J. Babiarz, A. Charny, X. Zhang, T. Taylor, K.-H. Chan, D. Satoh, R. Geib, and G. Karagiannis, "A survey of PCN-based admission control and flow termination," *Communications Surveys Tutorials, IEEE*, vol. 12, no. 3, pp. 357–375, quarter 2010.
- [4] M. Menth and M. Hartmann, "Threshold configuration and routing optimization for PCN-based resilient admission control," *Computer Networks*, vol. 53, pp. 1771–1783, 2009.
- [5] M. Menth and F. Lehrieder, "PCN-Based Measured Rate Termination," *Computer Networks*, vol. 54, 2010.
- [6] P. Eardley, "Pre-Congestion Notification (PCN) Architecture," RFC 5559 (Informational), Jun. 2009. [Online]. Available: <http://www.ietf.org/rfc/rfc5559.txt>
- [7] S. Latré, B. De Vleeschauwer, W. Van de Meerssche, F. De Turck, P. Demeester, K. De Schepper, C. Hublet, W. Rogiest, S. Custers, and W. Van Leekwijck, "Design and Configuration of PCN Based Admission Control in Multimedia Aggregation Networks," in *IEEE Globecom*, 2009, pp. 1–8.
- [8] S. Latré, B. De Vleeschauwer, W. Van de Meerssche, K. De Schepper, C. Hublet, W. Van Leekwijck, and F. De Turck, "PCN based admission control for autonomic video quality differentiation: Design and evaluation," *Journal of Network and Systems Management*, pp. 1–26, 2010, 10.1007/s10922-010-9183-8.
- [9] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, February 2004.
- [10] S. Wenger, Y.-K. Wang, and T. Schierl, "Transport and signaling of SVC in IP networks," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1164–1173, sept. 2007.

## BIOGRAPHIES

### A. Steven Latré

Steven Latré obtained a masters degree in computer science from Ghent University, Belgium, in June 2006. In August 2006, he joined the Department of Information Technology at Ghent University, where he received a Ph.D. in Computer Science Engineering in June 2011. Since then, he is active as a post-doctoral fellow at the same university. His main research interest is the use of autonomic network management approaches with a special focus on Quality of Experience optimization and the management of federations.

### B. Klaas Roobroeck

Klaas Roobroeck graduated as an industrial engineer in computer science from the University College Ghent in 2009. Since then, he works at the Department of Information Technology at Ghent University as a research assistant. His main research topics are the management of multimedia networks and the design of recommendation systems.

### *C. Tim Wauters*

Tim Wauters received his M.Sc. degree in electro-technical engineering in June 2001 from Ghent University, Belgium. In January 2007, he obtained the Ph.D. degree in electro-technical engineering at the same university. Since September 2001, he has been working in the Department of Information Technology at Ghent University, and is now active as a post-doctoral fellow. His main research interests focus on network and service architectures and management solutions for scalable multimedia delivery services.

### *D. Filip De Turck*

Filip De Turck received his M.Sc. degree in Electronic Engineering from the Ghent University, Belgium, in June 1997. In May 2002, he obtained the Ph.D. degree in Electronic Engineering from the same university. At the moment, he is a full-time professor affiliated with the Department of Information Technology of Ghent University. His main research interests include scalable software architectures for telecommunication network and service management, performance evaluation and design of new telecommunication and eHealth services.