

PCN Based Admission Control for Autonomic Video Quality Differentiation: Design and Evaluation

Steven Latré · Bart De Vleeschauwer ·
Wim Van de Meerssche · Koen De
Schepper · Christian Hublet · Werner
Van Leekwijck · Filip De Turck

the date of receipt and acceptance should be inserted later

Abstract The popularity of multimedia services has introduced important new challenges for broadband access network management. As these services are very prone to network anomalies such as packet loss and jitter, accurate admission control mechanisms are needed to avoid congestion. Traditionally, centralized admission control mechanisms often underperform in combination with multimedia services, as they fail to effectively characterize the amount of needed resources. Recently, measurement based admission control mechanisms have been proposed such as the IETF Pre-Congestion Notification (PCN) mechanism, where the network load is measured at each intermediate node and signaled to the edge, where the admittance decision takes place. In this article, we design a PCN based admission control mechanism, optimized for protecting bursty traffic such as video services, which is currently not studied in the PCN working group. We evaluated and identified the effect of PCN's configuration in protecting bursty traffic. The proposed admission control mechanism features three main improvements to the original PCN mechanism: first, it uses a new measurement algorithm, which is easier to configure for bursty traffic. Second, it allows to automatically adapt PCN's configuration based on the traffic characteristics of the current sessions. Third, it introduces the differentiation between video quality levels to achieve an admission decision per video quality level of each request. The mechanism has been extensively evaluated in a packet switched simulation environment, which shows that the novel admission control mechanism is able to protect video traffic while maximizing the link utilization and avoiding packet loss.

Keywords Autonomic · Network management · Multimedia · Adaptive

S. Latré, W. Van de Meerssche, F. De Turck
Ghent University - Department of Information Technology - IBBT,
Gaston Crommenlaan 8/201, B-9050 Gent, Belgium
Tel.: +32-9-331481 - Fax: +32-9-3314899 - E-mail: steven.latre@intec.ugent.be

B. De Vleeschauwer, K. De Schepper, C. Hublet, W. Van Leekwijck
Alcatel-Lucent Bell Labs, Copernicuslaan 50, B-2018 Antwerpen, Belgium

1 Introduction

The advent of multimedia services such as IPTV, Video on Demand and Network-Based Personal Video Recording has put a strain on the management of today's broadband access networks. Although these services provide important new experiences to users, and consequently, considerable more revenue to network operators, their management is far more complex than that of more traditional ones such as browsing the web. On one hand, multimedia services require a much larger amount of bandwidth: these services already make up for the largest share in terms of bandwidth, and their share is continuously growing due to the evolution to higher definition video and constant quality video, obtained through Variable Bit Rate (VBR) encoding. On the other hand, these services have far larger quality requirements: even small amounts of packet loss ratio or jitter can lead to visual and audial artifacts such as blockiness for video.

As fiber or copper based broadband access networks consist of reliable transmission links, the main challenge of network management in access networks is preventing network congestion through admission control mechanisms. As illustrated in Figure 1, operators have tried to tackle this issue by protecting each session one by one and using the peak bit rate of the video as the amount of resources to reserve. While this effectively protects the video from congestion a large amount of bandwidth is wasted as this peak rate represents the worst case scenario, occurring at most a few times during the complete service, since a single VBR session is usually very variable in terms of bandwidth consumption. This results in an underutilized access network. A more efficient way to utilize resources is to protect only the aggregate of sessions as opposed to protecting every individual session independently, one by one. The main advantage lies herein, that the variability of the aggregate will in general be lower than that of individual sessions, thus allowing to carry more video sessions over the same link, enabling better resource usage than in the case of protection on a per-session basis.

Such an aggregate based protection requires a measurement based approach where the network load is dynamically characterized to reach an accurate admission decision. Recently, the IETF suggested a promising measurement based admission control mechanism through the Pre Congestion Notification (PCN) mechanism. In PCN, each node measures the network load independently and signals the information through in-band packet marking to the edge of the network, where the admission decision takes place. Currently, the PCN working group has published the PCN architecture as an RFC [1] and is working on different encoding options for signaling load information. At this stage, the PCN working group has restricted the scope through some assumptions, including the condition that all flows must be constrained to a known maximum rate, which is not straightforward for VBR videos.

In our work, we extend the original PCN mechanism to also enable the protection of bursty video traffic. The contributions of this article are the following. First, we evaluate PCN's performance when applied to bursty traffic

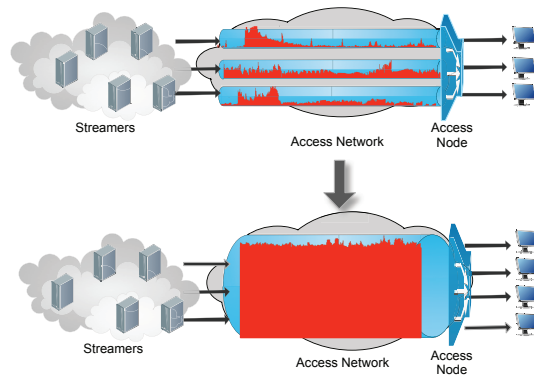


Fig. 1: Possible ways to realize admission control. Top: the centralized approach where resources are reserved using the videos’ peak rate. Bottom: new approach where the aggregate of sessions is protected, leading to a better utilization.

and extract guidelines for configuring PCN. We evaluate both the traditional PCN measurement algorithm and a new one, using bandwidth measurements, which is easier to configure for bursty traffic. Second, based on the obtained guidelines, we propose an adaptive algorithm, which we refer to as aPCN+, and that dynamically changes PCN’s configuration to the behavior of the aggregate. Third, we extend the capabilities of the PCN mechanism with an admission control algorithm that differentiates between different video level qualities and allows to selectively admit sessions based on the desired video quality.

The remainder of this paper is structured as follows: Section 2 discusses relevant work in the domain of admission control for network management. Section 3 provides an elaborate discussion on the PCN architecture, while Section 4 contains an extensive evaluation of PCN combined with bursty traffic and the construction of guidelines based on the conducted experiments. In Section 5, a novel admission control mechanism is proposed that adaptively changes PCN’s configuration, based on the major shortcoming identified in PCN’s performance evaluation, and makes a fine grained admission decision based on different video quality levels. Finally, the performance of the proposed mechanism is evaluated in Section 6.

2 Related work

Congestion control of multimedia services has traditionally relied on centralized approaches such as those defined in TISPAN [2], ITU-T [3] and Diff-serv [4]. These traditional admission control mechanisms have some important drawbacks for protecting the growing share of multimedia sessions. First, determining how many resources to reserve is crucial and troublesome for today’s multimedia services which often offer constant quality video and are by nature

very bursty. A second limitation, is that the amount of knowledge can be very large and hard to keep up to date, which introduces some scalability issues. Traditionally, operators have tackled these limitations by over-dimensioning the number of requested resources for each session. Although this effectively protects the existing sessions, the network is under-utilized.

Distributed admission control mechanisms such as the standardized one in Intserv [5], successfully tackle the scalability issue but still have difficulties to cope with the dynamically changing resource demands. The RSVP TSPEC, used as traffic specification in Intserv, is too static to cope with the continuous resource evolution of video services such as video streaming. In literature, also other decentralized solutions have been proposed that abandon the flow-state maintenance of Intserv by introducing passive measurements at ingress and egress nodes in an Intserv domain [6]. In [7], the abstraction of a statistical envelope is used as a general framework for characterizing a service. In [8] the focus is on a wireless setting and in [9] the authors discuss an admission control system for large-scale media delivery systems. However, also for these mechanisms, the lack of an accurate measurement of the network load remains a challenge.

It is clear that distributed measurement based mechanisms can solve this issue, as they have a dynamic view on the network status. Recently, the IETF proposed the PCN mechanism as a fully distributed and measurement based admission control mechanism which does not require flow state information in the core. The PCN mechanism measures the network load in the network core and signals this to the egress nodes in a PCN domain who estimate the congestion level. Within the PCN working group, simulations have shown that PCN is able to effectively protect narrowband services such as Voice over IP in a Diffserv domain [10,11]. In our work, we evaluate PCN's performance for protecting services which do not have a known maximum rate (e.g. VBR video services) [12]. These results have led to the concept of adapting PCN's configuration based on earlier measurements. While the algorithm proposed in [13], which in this article is labeled aPCN, resulted in satisfying results for low request rates (i.e. 2 requests per second), in this article, we have further refined the adaptive algorithm, labeled aPCN+, which supports higher request rates (e.g. in the case of flash crowds).

Specifically for video services, another form of admission control mechanisms exist that relies on the use of scalable video coding (SVC) for controlling the network load. In an SVC based admission control mechanism, the option exists to not only block the request but to allow a new request at a reduced video quality level [14,15] when congestion is imminent. This approach can force the video to switch to a lower video quality level but still requires other, more traditional, admission control mechanisms to block sessions when even the lowest quality level cannot be admitted anymore. In this article, we combine a more traditional measurement based admission control mechanism, based on the standardized PCN mechanism, with SVC based admission control.

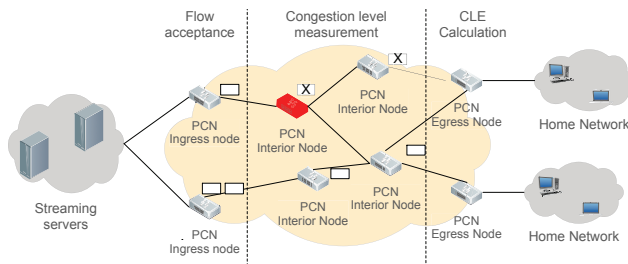


Fig. 2: Overview of the PCN architecture. PCN interior nodes measure the network load and start marking packets if the load becomes too high.

3 Pre-Congestion Notification mechanism

3.1 Architecture Description

An overview of the PCN architecture is illustrated in Figure 2. The PCN architecture consists of three different node types: all traffic enters a PCN domain through a PCN ingress node and leaves the PCN domain through a PCN egress nodes. The PCN interior nodes are PCN-capable nodes inside a PCN domain. These node types all support different types of functionalities. The interior nodes are responsible for measuring and marking: they measure the network load caused by already existing traffic and will start marking once the load exceeds a certain threshold, a sign that the particular interior router is (almost) congested.

The egress nodes collect the marked packets and calculate a Congestion Level Estimation (CLE). A CLE value can be between 0 and 1, where 1 denotes pre-congestion and 0 corresponds to no congestion. The CLE value at time n is calculated using an exponentially weighted moving average: $CLE_n = X * (1 - w) + w * CLE_{n-1}$, $w \in [0, 1]$. Here, w is the CLE weight and X is 1 if the packet is marked and 0 if the packet is not. The higher the CLE weight, the more previous measurements will contribute to the overall CLE value. Once calculated, the CLE information is signaled back to the ingress node which is responsible for admitting or blocking sessions.

3.2 Measurement function: algorithmic options

As discussed in Section 3.1, the interior node provides a measurement function, where the network load is monitored and compared to a threshold. This measurement function is the heart of the PCN architecture: as the measured network load forms the basis for the admittance decision, an accurate and timely detection of a pre-congestion state is needed. In this section, we discuss two possible measurement algorithms: a token bucket approach and a bandwidth metering approach.

3.2.1 Token bucket

A token bucket can be regarded as a simple bit counter with known boundaries. The token bucket measurement algorithm for PCN has three parameters: a token rate R , a depth d and a threshold t : bits are continuously added to the token bucket at a constant rate R until the number of tokens in the bucket has reached the maximum value, denoted by depth d . At the same time, tokens are removed from the bucket when a packet arrives at the interior node. The evolution of the bucket indicates whether or not the bandwidth of the aggregate is lower than the rate R . The number of tokens in the bucket increases when the bandwidth of the aggregate is lower than the given token rate R and decreases when the bandwidth of the aggregate is higher than the token rate R . Hence, a small number of tokens in the bucket is a sign that the network load has increased above the token rate. The third parameter of the token bucket, the threshold t is used to decide to mark a packet or not: if the total number of tokens is below the threshold t , the packet is marked, as a sign of pre-congestion, otherwise the packet is left untouched. In the remainder of this paper, we refer to the token rate as the configured rate.

3.2.2 Bandwidth metering

A token bucket provides a low complexity approach to measure the network load but can only provide information about its proximity to the token rate R . A more complex algorithm is performing actual bandwidth measurements, which provides a more detailed view on the network status. Different options exist to perform such bandwidth measurements; in our approach, we focus on a sliding window algorithm as this provides the most accurate calculation of bandwidth. In this algorithm, the measurement window (denoted by MI) slides over the arrived packets, and each time a packet arrives the bandwidth is measured based on the packets received during the last MI seconds. Packets sent during bandwidth peaks, introduce more measurements than packets during lower bandwidths. In this approach, packets are marked when the bandwidth measurement at that time is higher than a predefined threshold rate R . In the remainder of this paper, we refer to this threshold rate as the configured rate.

4 Performance characterisation of the original PCN mechanism

4.1 Access network model

All evaluations were carried out on an access network model as depicted in Figure 3. This access network model represents a typical tree based access network where a video head end streams a bouquet of videos to home networks. All results were obtained through a modified NS-2 simulator, developed by the authors [16], capable of transmitting real video sequences over a simulated network topology and calculating the video quality of the received videos.

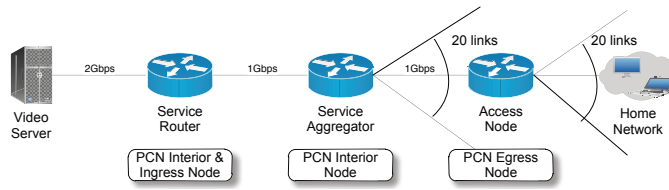


Fig. 3: Investigated network topology modeling a tree based multimedia access and aggregation network.

This set-up contains one congestion point on the service router where the link capacity decreases from 2 Gbps to 1 Gbps. In the evaluated scenario, a set of 400 clients randomly requests videos to the head-end: the PCN mechanism is responsible for blocking or allowing the requested sessions. We modeled the request process through a uniform random distribution with a fixed request rate up to 1000 requests per second. This high request rate allows to evaluate the admission control mechanism's performance on protecting itself against flash crowds. Any higher request rate is assumed to be handled by an external request shaper.

To evaluate PCN's performance, we investigated the transmission of CBR data traffic and VBR video traffic. The CBR data traffic had a bitrate of 2.5 Mbps, while the VBR video traffic was encoded as an H.264 video with a PAL resolution and a framerate of 25 fps. We used a constant quality coding scheme, which resulted in traffic which is bursty of nature but had an average of 2.5 Mbps. Unless otherwise stated the CLE weight was set to 0.9.

Besides network metrics such as the number of admitted sessions, we also investigated the effect on the video quality. The video quality is measured through the Structural Similarity (SSIM) [17]. The SSIM metric produces a score between 0 and 1, where 1 stands for perfect quality. The SSIM scores should be interpreted as follows: a video with a SSIM score above 0.9 is indistinguishable from the original, a SSIM score between 0.8 and 0.9 corresponds with a moderate quality while a SSIM score of 0.7 and lower results in a video which is barely watchable.

4.2 Terminology

In this section, we provide a definition of the most important terms used in describing the behaviour of the PCN algorithm, throughout the rest of this article.

- *Bandwidth aggregate*: the bandwidth aggregate is the bandwidth that can be measured on a specific link and that is inflicted by multiple sessions traversing that link.
- *Goal rate*: An admission control mechanism typically wants to protect its existing resources by avoiding that the bandwidth aggregate exceeds a threshold. We call this threshold the goal rate: this goal rate can be set to the link capacity or lower.

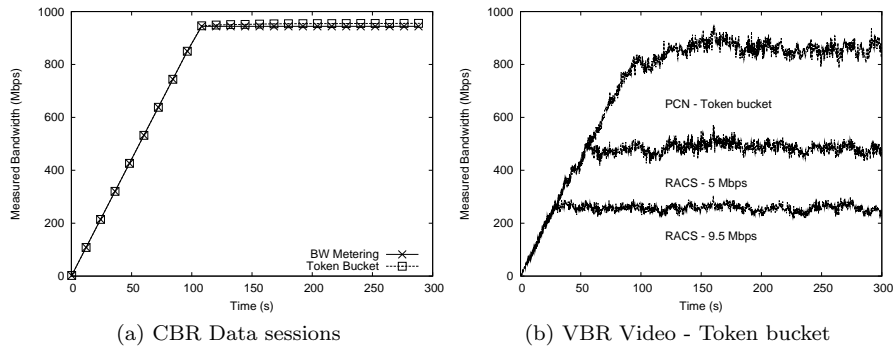


Fig. 4: Evolution of the measured bandwidth over time for a mix of CBR data sessions (a) and VBR video sessions (b). In the VBR case, the PCN mechanism is compared with two configurations of the RACS mechanism.

- *Configured rate*: While the goal rate corresponds with the desired behaviour of the system, the configured rate is the rate used to achieve this goal. The configured rate is a parameter of the PCN measurement algorithm in the interior nodes. For the token bucket algorithm, it corresponds with the token rate R ; for the bandwidth metering algorithm, it corresponds with the threshold rate R .
- *Aggregation interval*: We define an aggregation interval as the time that a PCN measurement algorithm uses to base the measurement on. For each measurement algorithm, this aggregation interval consists of two components. For the token bucket algorithm, the aggregation interval consists of the token bucket depth and the CLE weight. For the bandwidth metering algorithm, the measurement window and CLE weight are the contributing parameters to the aggregation interval.

4.3 Original PCN mechanism: results description

Impact of the traffic type Figure 4 illustrates the evolution over time of the bandwidth aggregate for the CBR (Figure 4a) and VBR case with the token bucket as measurement algorithm. (Figure 4b). In Figure 4b, only the behaviour of the token bucket measurement algorithm is illustrated but the bandwidth metering algorithm can be configured to obtain identical results [12]. As illustrated in Figure 4b, we also compared PCN’s performance with a traditional centralized admission control mechanism as defined in the TISPAN RACS [2], where a fixed amount of resources is reserved for each session.

In the CBR case (Figure 4a), the configured rate was set to 990 Mbps, the request rate was set to 4 requests per second and the token bucket depth was set to 500,000 bits in the token bucket algorithm. It is clear that configuring a PCN mechanism for protecting CBR data sessions is very straightforward. Furthermore, there is no difference between the token bucket and bandwidth

metering approach. In both approaches, the measured bandwidth averages around 990 Mbps, while exactly 396 sessions are allowed. The same behavior can be observed for the centralized RACS mechanism: the resource requirement per session is set to 2.5 Mbps for each session, again exactly 396 sessions are allowed.

When the traffic consists of VBR sessions, the optimal configuration changes dramatically. This is illustrated in Figure 4b, which shows PCN's performance compared with two configurations of the RACS mechanism, where the resource requirement per session is fixed to 5 Mbps and 9.5 Mbps, respectively. This latter value corresponds with the peak rate of all transmitted videos. In this experiment, the request rate was set to 4 requests per second, the configured rate was set to 825 Mbps and the token bucket depth was set to 16,000,000 bits. As will be discussed in the remainder of this section, these values were chosen as they result in an accurate PCN mechanism during normal (i.e. non flash crowds) operation. This alternate configuration was needed to cope with the variability of the aggregate, a phenomenon which is of course not present in the CBR case. In the PCN mechanism, the token bucket and bandwidth metering algorithms continue admitting sessions until all measurements (in terms of tokens or actual bandwidth measurements) are above the configured rate. When only some measurements are above the configured rate, the interior nodes will not mark all packets resulting in the sporadic admission of new sessions. Therefore, the configured rate must be a lot lower than in the CBR case to reach the same goal.

If we compare PCN's performance with a centralized RACS mechanism, we see the clear gain of using PCN. Applying PCN results in a large increase in the network utilization for bursty traffic when compared to the RACS mechanism. When the RACS is configured at 9.5 Mbps, the bandwidth aggregate stabilizes around 250Mbps, resulting in a network utilization which is 3 times lower than when PCN is applied. A RACS configuration of 5 Mbps results in a bandwidth aggregate of approximately 500 Mbps, which is still considerable lower than when PCN is applied.

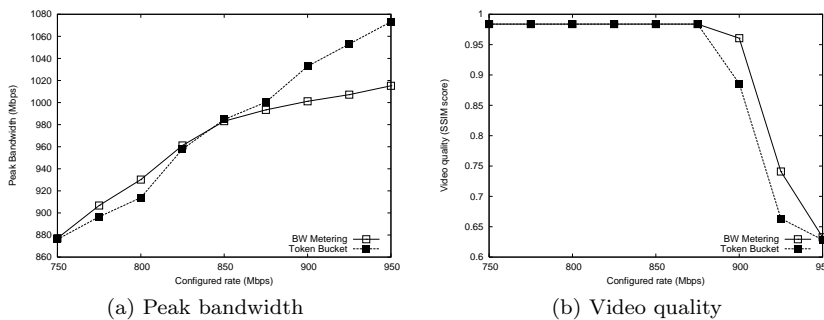


Fig. 5: Impact of the configured rate on the measured bandwidth and average video quality for the bandwidth metering and token bucket approach.

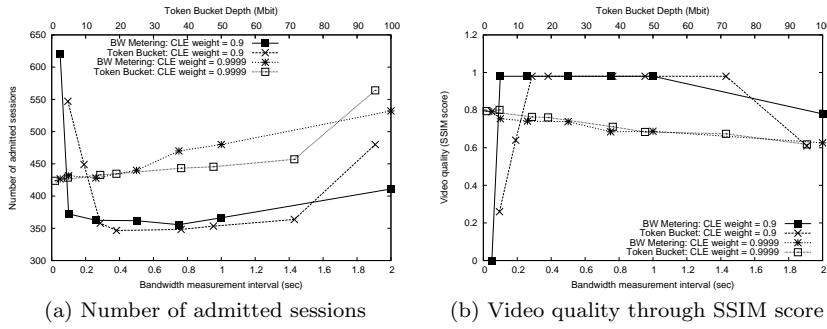


Fig. 6: Influence of the aggregation interval on the admittance (a) and video quality (b) for the bandwidth metering aggregation interval (lower x axis) and the token bucket depth (upper x axis).

Impact of the configured rate The results in the previous paragraph already suggested that the configured rate is an important parameter in the configuration of any PCN mechanism for VBR sessions. This is further investigated in this set of experiments, where we evaluate the impact of the configured rate. Here, the request rate was set to 4 requests per seconds. The measurement algorithms were set as follows: the token bucket depth d was set to 16,000,000 bits, while the the measurement window MI of the bandwidth measurement algorithm was set to 140 ms.

The influence of the configured rate on the peak bandwidth and the SSIM for both measurement algorithms is illustrated in Figure 5. We define the peak bandwidth as the highest observed bandwidth measurement during the complete experiment. The most important observation that can be made here is that the configured rate in both measurement algorithms should be seen as a parameter of the system and not as the goal rate of the admission control mechanism. Increasing the configured rate to 900 Mbps and more will lead to a drop in video quality as packet loss occurs. Although this configured rate is still 100 Mbps lower than the link capacity of 1 Gbps, the variability of the aggregate, which is higher than 100 Mbps, causes packet loss.

Impact of the aggregation interval While the CLE definitely contributes to the aggregation interval, the token bucket depth d and measurement window MI need to be large enough as well: it is not advisable to use a small bucket depth or measurement window and a very large CLE. This is illustrated in Figure 6, which shows the effect on a varying token bucket depth and measurement window in combination with a small and large CLE weight in terms of admitted sessions (Figure 6(a)) and SSIM score (Figure 6(b)). As can be expected the combination of a small depth or measurement window in combination with a small CLE weight results in over-admittance and consequently a drop in video quality. This is because the size of the aggregation interval is too small to cope with the bursts.

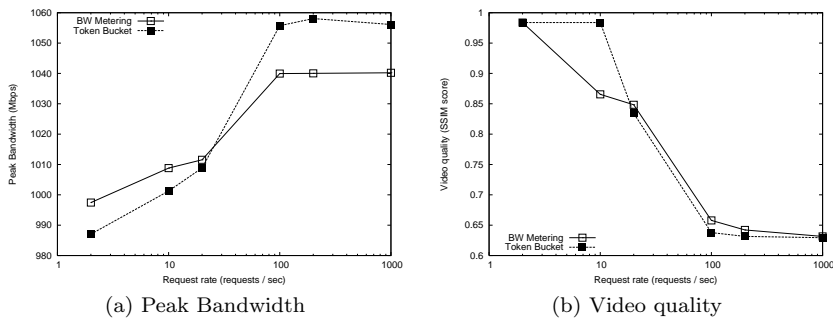


Fig. 7: Influence of the request rate on the measured bandwidth and average video quality for the bandwidth metering and token bucket approach.

What is less intuitive is the fact that combining a large CLE weight with a small bucket depth or measurement window also results in a poor visual quality, as can be seen from the SSIM score of 0.8 when using a CLE weight of 0.9999. This is due to the small token bucket depth or measurement window on one hand and the way information is signaled in the PCN mechanism on the other hand. The small bucket depth or measurement window causes an oscillating effect on the marking of packets. One would expect that a CLE weight of 0.9999 would smooth out these oscillations. However, the way information is signaled in the PCN architecture results in loss of information. As the marking of packets is the only information the CLE calculation can rely on, it needs a continuous stream of marked packets to recalculate the value. When a silence (e.g. after a burst) occurs, the CLE value is not adjusted, as no new packets arrive, which results in inaccurate measurements.

A last observation we can make is that increasing the aggregation interval can have a deteriorating effect on the video quality. This effect occurs because the delay introduced by the measurement algorithm becomes too large and the PCN mechanisms starts reacting too slow. While, for this experiment, we used a request rate of 4 requests per second, the importance of this delay increases for higher request rates. This is discussed in the next section.

Impact of the request rate The previous section indicated that the aggregation interval needs to be sufficiently large to cope with the burstiness of the traffic. This large aggregation interval has an effect on the delay introduced by the measurement algorithm: the larger the time needed to smooth out the bursts, the longer it will take for the measurement algorithm to detect a pre-congestion state. As such, the introduced delay can become too high when a flash crowd occurs. This is illustrated in Figure 7, which shows the influence of the request rate on the peak bandwidth and video quality. Here, the token bucket depth was set to 16,000,000 bits and the measurement window was set to 140 ms. In both measurement algorithms, we used a rate of 875 Mbps. The figure shows that this configured rate results in a perfect video quality for low request rates

but continues to drop when the number of requests per second increases. This is because the PCN mechanism will admit too much sessions as it detected the pre-congestion state too late.

4.4 Comparison between measurement algorithms

Although both approaches can be used to obtain information about the network load, it is important to stress that these are two different mechanisms with different complexities and ease of configuration.

4.4.1 Complexity

The token bucket algorithm has the lowest complexity for measuring network load: it is in essence just a bit counter with known boundaries and therefore easy to implement in hardware. Additionally, token buckets are often already present in today's routers. The discussed bandwidth metering algorithm has a greater complexity both in memory usage and computational power: it requires memory equal to the maximum number of packets that can be received during the time frame MI . As calculating the bandwidth requires partially sliding over the window of packets it also requires more computations.

4.4.2 Aggregation interval

The added complexity of the bandwidth metering approach also introduces added benefits. Both approaches rely on an aggregation interval to smooth out bursty measurements. As discussed in the previous section, a good aggregation interval is key to a well performing measurement algorithm.

The bandwidth metering works on a time basis as it aggregates over the packets received during the last measurement window. In this case, the aggregation interval is a direct parameter of the measurement algorithm as it is equal to the measurement window MI . For the token bucket approach, it is not possible to directly translate the token bucket parameters to the aggregation interval. The token bucket approach is a packet based algorithm, where a burst of packets will cause the token bucket to decrease sooner than when packets arrive at a constant bit rate.

This has important consequences for the ease of configuration of these approaches. In situations where bursts of data are likely to occur, finding an appropriate aggregation interval is crucial. In this case, the configuration of a bandwidth metering algorithm is much easier than the token bucket algorithm. For the latter, information about the size of bursts is needed to find a suitable configuration. Although it is possible to reach the same performance with a token bucket mechanism (e.g. by performing a sweep of all token bucket parameters), we believe that the bandwidth metering approach is best suited for protecting bursty traffic due to this ease of configuration.

4.5 Construction of configuration guidelines

Based upon the obtained results, discussed in Section 4.3, we can construct a number of guidelines for configuring the bandwidth metering measurement algorithm in the case of bursty video traffic. These guidelines should strictly be interpreted as rules of thumb. For the *configuration of the aggregation interval*, it is important to choose a token bucket depth or measurement window that is large enough to cope with the variability of the aggregate. A small CLE weight (e.g. 0.9) should be chosen as the PCN mechanism cannot completely smooth out all fluctuations in marked packets due to loss of information. In the bandwidth metering case, a good rule of thumb is to choose the measurement window MI by taking the lowest measurement window with a stable $\text{maxVariability}(MI)$, where $\text{maxVariability}(MI) = \text{maxBW}(MI) - \text{minBW}(MI)$ and $\text{maxBW}(MI)$ and $\text{minBW}(MI)$ are the maximum and minimum bandwidth measured using a measurement window MI when congestion is imminent. The first stable maxVariability measurement in this context is defined as the smallest measurement interval where one of the following two conditions holds:

- The decrease in measured variability, to be expected from further decrease of the measurement interval, goes below a pre-assumed threshold i.e.

$$\frac{\text{maxVariability}(MI + Y)}{\text{maxBW}(MI + Y)} - \frac{\text{maxVariability}(MI)}{\text{maxBW}(MI)} < X \quad (1)$$

where X is small value such as 0.02, and Y is a short time, such as 0.01s.

- The variability is small, i.e.

$$\frac{\text{maxVariability}(MI)}{\text{maxBW}(MI)} < Z \quad (2)$$

where Z is a small value such as 0.05.

Unlike the configured rate, small changes in the value of the aggregation interval do not have an immediate impact on potential errors in the admission control mechanism. As illustrated in Section 4.3, there are a lot of values that contribute to the same performance. Therefore, for the measurement window, a common discriminator can be found using the above rules of thumb and configured statically. As discussed in Section 4.4, it is not possible to construct a similar rule of thumb for the token bucket algorithm without having detailed knowledge of the size of the bursts to determine the impact on the token bucket.

When *configuring the rate* it is important to take the variability of the aggregate into account. The configured rate should always be lower than the goal rate, minus this variability. If the configured rate is higher, occasional packet loss will occur due to over-admission. It is important to further decrease this configured rate to take *flash crowds (i.e. high request rates)* into account. Therefore, we propose the following value for the configured rate:

$$\text{configuredRate} \equiv \max(0, \text{goalRate} - \text{maxVariability} - \text{overshoot}) \quad (3)$$

where overshoot represents the additional sessions that are admitted due to the introduced delay. This is characterized by $\text{overshoot} \equiv MI \times \text{MaxRequestRate} \times \text{avgSessionBW}$. MaxRequestRate is the maximum request rate which should be supported and avgSessionBW is an estimation of the average bandwidth per session. In case of a mixed bouquet of video resolutions (e.g. a mix of HD and SD videos) the avgSessionBW will be a value between those different video resolutions.

5 Measurement based admission control: video quality differentiation

A traditional admission control mechanism can only decide whether or not to admit a new session. However, the inherent structure of today's video services, allows for a more fine grained admission control decision. If not enough resources are available, an admission control mechanism may opt to still allow the requested session but at a reduced video quality, which requires less resources. This paradigm is being frequently used in commercial products such as the BBC iPlayer [18]. While the user will not get the best video quality as originally requested, he will still be able to consume the video, but at a lower video quality. In this section, we propose a mechanism based on the original PCN admission control architecture, which is specifically optimized towards the protection of video traffic. This approach combines the SVC paradigm with the measurement based PCN mechanism. However, to be fully able to protect bursty video traffic, we propose a mechanism to dynamically adapt the configured rate based on the guidelines constructed in Section 4.3.

5.1 Automatic rate adaptation

5.1.1 Real-time estimation of the variability

The constructed rule of thumb for setting the configured rate, as defined in Equation (3), provides a guideline for setting the configured rate. Note that the maxBW and minBW values in the equation of maxVariability are calculated by taking respectively the maximum and minimum bandwidth measurements, e.g. obtained from the bandwidth metering algorithm, during a certain time window. As the equation of maxVariability does not dictate what the time window should be, in an on-line algorithm we will need to define suitable values for these time windows. Two other modifications are needed to Equation (3) to make it suitable for setting the rate dynamically. First, the maxVariability value denotes the difference between the maximum (maxBW) and minimum (minBW) bandwidth measurement when the link is in pre-congestion state. However, in an adaptive scenario we will also need to make an estimation of the configured rate even if only a few sessions are present. Second, the minBW value requires the use of the lowest bandwidth measurement. When,

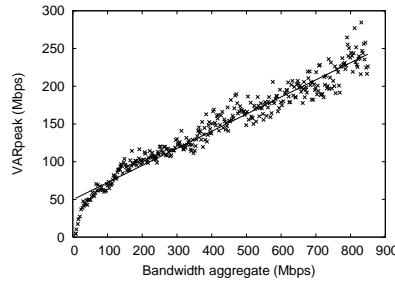


Fig. 8: Influence of an increasing bandwidth aggregate on the VAR_{peak} value. There is a clear linear dependency between these two values.

during a certain time window, new sessions are admitted the $minBW$ value cannot accurately characterise the maxVariability as the minimum bandwidth measurement will correspond with the bandwidth measurements that were obtained before the new sessions were admitted.

Based upon the discussed limitations, we propose to use a new estimator for the variability that (1) only takes into account the maximum bandwidth measurement and (2) estimates the variability in pre-congestion state. If we assume that, for any link in pre-congestion state, the difference between the maximum bandwidth and average bandwidth is about the same as the difference between the average bandwidth and the minimum bandwidth then $VAR_{peak} = 2 \times (\maxBandwidth - \text{averageBandwidth})$ is a metric that discards the minimum bandwidth measurements. In order to provide an accurate estimation of VAR_{peak} even if a few sessions are admitted, we need to investigate how this value evolves as a function of the bandwidth of the aggregate. This is illustrated in Figure 8, which shows a clear linear dependency between the bandwidth aggregate and the VAR_{peak} value. As such we can use this linear dependency to interpolate the VAR_{peak} measurement, taken when only a few sessions are admitted. As will be explained in the next section, this linear dependency will be used in the aPCN+ algorithm to make a better estimation of the variability of the bandwidth aggregate under a high network load, even under low network load values. As the dependency is linear, the calculation is straightforward.

5.1.2 aPCN+ : Algorithm details

The adaptive-PCN+ (aPCN+) algorithm is described in Algorithm 1: here, the rate will be continuously recalculated each time a packet arrives. The algorithm sets the rate based on the calculation of the VAR_{peak} value. However, in the calculation of maxBW and avgBW different time frames are used. To calculate maxBW, the maximum bandwidth measured during a time frame $t_{monpeak}$ is used, a value which will be considerable larger than t_{monavg} , the time frame used to calculate the average bandwidth. In practice, $t_{monpeak}$ will be in the order of 10 seconds or more, while t_{monavg} will only be a few seconds.

Algorithm 1 The algorithm for automatically adapting the configured rate based on previous bandwidth measurements.

```

let avgBW = updateAverageBW(p,tmonavg)
  where  $t_{\text{monpeak}} > t_{\text{monavg}}$ 
let currentVARpeak =  $2 \times (\text{maxBW} - \text{avgBW})$ 
let estimatedVARpeak =  $(\alpha \times \text{currentVARpeak} + \beta) - \text{overshoot}$ 
let rate = link.capacity - estimatedVARpeak
setRate(rate)

```

All values, except the t_{monpeak} and t_{monavg} values can be obtained by using the guidelines constructed in Section 4.5. These values are used to make sure that we can avoid congestion but still react quickly to changes in the network load. The rationale behind using two different values for the t_{monpeak} and t_{monavg} is the following: the large t_{monpeak} value will take a conservative approach in calculating the peak bandwidth: even peaks in the bandwidth that happened a long time ago will be taken into account. This guarantees that the variability is not estimated too low because of a temporarily absence in bursts. At the same time, the much smaller t_{monavg} value allows the timely detection of increases in the overall bandwidth, e.g. because new sessions were admitted. The received value is used as the basis for a linear interpolation that estimates the variability in a pre-congestion state. In our algorithm, the α and β are determined by periodically sampling the currentVARpeak values and adjusting the linear interpolation factors accordingly. However, as this linear dependency does not change often, this can be performed at a much slower pace than the rate adaptation itself (e.g. only once every hour). The overshoot value is used to compensate for a potential overshoot due to a flash crowd and was characterized in Section 4.5.

5.2 Video Quality Differentiation

The concept of video quality differentiation within the PCN mechanism is straightforward: instead of returning a binary decision (admit or block), the PCN mechanism returns more information about which video qualities can be admitted based on the current number of resources still available. The network operator should set which video quality levels (e.g. Full HD, SD and a YouTube like video) can be supported and is responsible for setting the policies regarding to which video qualities are allowed under which network load. These policies are set by specifying a percentage of the configured rate that is reserved for a specific video quality. If the bandwidth measurements are above this value, that particular video quality level cannot be admitted anymore and a lower video quality needs to be chosen. For example, suppose the configured rate in a static PCN mechanism is set to 800Mbps. The network operator may choose to reserve 40% of its traffic for HD videos and the remaining 60% for SD videos. In this case, only SD videos will be admitted to the network once the bandwidth measurements are higher than 320 Mbps.

5.2.1 PCN interior functions

The PCN interior nodes need to distinguish between the different video quality levels. By using the policies, stated by the operator, the interior nodes need to calculate the maximum allowed video quality level, where a high video quality level represents a high video resolution. Instead of simply marking a packet with one bit, the video quality level is signaled to the PCN egress nodes. In case there are several interior nodes on a path, the video quality level encoded in the packet is only replaced if the calculated maximum video quality level is lower than the original one.

This function can easily be integrated in both the token bucket and bandwidth metering approach. In the token bucket algorithm, several token buckets need to be maintained: one for each video quality level. These token buckets will all have a different token rate. The highest possible video quality level of which the number of tokens in the corresponding token bucket is higher than its token bucket threshold t is used as video quality level to signal. In the bandwidth metering algorithm, we can simply compare our bandwidth measurement with different rates, each corresponding with a video quality level. The highest rate, that is still lower than the bandwidth measurement, is chosen and the corresponding video quality level is chosen as the highest video quality level.

5.2.2 Marking and encoding options

The signaling can be done in a number of ways. One possibility is to use multiple bits to mark the packets as opposed to one single bit. This is feasible as there will only be a limited amount of video qualities. With 2 bits, already 3 different video qualities and a blocking state can be supported, which is probably sufficient for most cases. This option has the downside of reserving additional bits in the packet header. Another option is to encode the video quality level in a series of packets: in this case a series of packets represents a bit stream that needs to be decoded. Note that this approach only works if there is no re-ordering of the packets: otherwise, the implicitly defined order in the encoding process would be violated. In our implementation, we used the first approach where we reserved 3 bits for signaling 6 different admission levels, consisting of 5 video quality levels and a blocking state.

5.2.3 PCN egress and ingress functions

Instead of a binary decision, the PCN egress needs to calculate the overall maximum allowed video quality level. Similar to the original PCN mechanism, this can be performed by calculating a CLE value through an exponentially weighted moving average. In this case, the resulting CLE is not a value between 0 and 1, as was originally the case, but represents an average value of the maximum allowed video quality level.

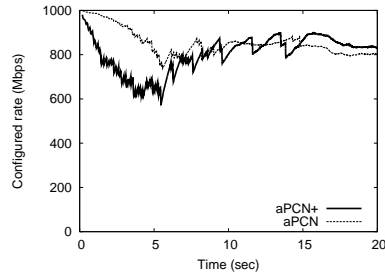


Fig. 9: Evolution of the configured rate over time for aPCN+ and the earlier proposed aPCN.

6 Performance evaluation results

In this section, we evaluate the performance of the video quality differentiation algorithm. First, we focus on the performance of the adaptive rate adaptation algorithm aPCN+ by comparing it with static configurations of the original PCN mechanism and investigating the influence of its parameters. Second, we investigate the gain in terms of video quality for different configurations of the video quality differentiation mechanism. Conceptually, the dynamic algorithm can be applied to both the bandwidth metering and the token bucket approach. Here, we focus on the bandwidth metering approach: during all experiments the measurement window MI was set to 140 ms and the CLE weight was set to 0.9: two values that showed the best performance in the evaluation results discussed in Section 4.

6.1 Dynamic rate adaptation

6.1.1 Evolution of the configured rate

The evolution of the configured rate over time is illustrated in Figure 9 for a request rate of 1000 requests per second. Here, we compared the evolution of the configured rate with that of the rate when applying an earlier version of the algorithm, which we refer to as aPCN, proposed in [13]. In aPCN, the variability was calculated by subtracting the maximum and minimum bandwidth: no linear interpolation was performed to compensate inaccurate measurements when the network is far from congested. Although we showed that aPCN was able to maximize the link utilization and avoid packet loss for low request rates (i.e. 2 requests per second), it runs into difficulties when applying it to larger request rates such as flash crowds of 1000 requests per second. aPCN makes an important over estimation of the needed configured rate during the initial seconds, where only a few sessions are allowed. On the other hand, aPCN+, takes a more conservative approach due to the linear interpolation when only few sessions are allowed by setting the configured rate lower than it should be. However, this conservative approach allows to also protect sessions during

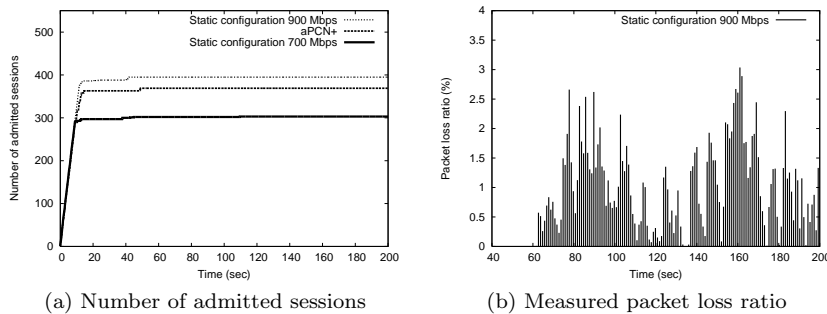


Fig. 10: Evolution of the number of admitted sessions and packet loss ratio for aPCN+ and configurations of the static bandwidth metering algorithm where the rate is set to 900 Mbps and 700 Mbps. Using a static rate of 900 Mbps results in packet loss.

flash crowds, something which is not possible in aPCN. During this experiment, aPCN failed in blocking the arriving requests in time which led to an average packet loss ratio of 1.2 %, a value which already deteriorates video quality.

In Figure 10, the number of admitted sessions and measured packet loss ratio is depicted over time for a request rate of 1000 requests per second. When comparing aPCN+ with static configurations of the PCN algorithm we see that aPCN+ is able to achieve both a maximization of the link utilization and avoidance of any packet loss. As illustrated in Figure 10, aPCN+ admits a total of 369 sessions, which is a bit lower than the 395 sessions, admitted by a static version of the PCN mechanism where the rate was configured at 900 Mbps. In this case, admitting 395 sessions results in packet loss because the rate was set too high. This is illustrated in Figure 10, which shows the average packet loss ratio as a function of the time. Setting the configured rate to 900 Mbps results in an average packet loss ratio of up to 3%.

6.1.2 Impact of the variability

As aPCN+ changes PCN's configuration based on measurements of the variability of the aggregate, this variability will have an impact on the overall performance. This is illustrated in Figure 11, where the evolution of the number of admitted sessions (Figure 11a) and packet loss ratio (Figure 11b) is plotted for an increasing variability of the aggregate. The request rate was set to 100 requests per second. As illustrated, the increasing variability has a decreasing effect on the number of admitted sessions in all configuration flavors. However, for aPCN+, the decrease is much more steep. This steep decrease is observed because aPCN+ detects the increasing variability and adjusts its configured rate accordingly. While, for a variability of 100 Mbps, its behavior is close to that of the static 900 Mbps configuration, it lowers its configured rate as the variability increases until it drops below the static 800 Mbps mark

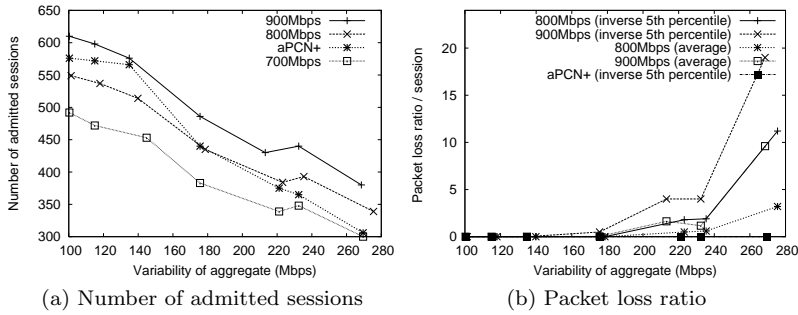


Fig. 11: Impact of the variability of the aggregate on the number of admitted sessions (a) and packet loss ratio (b). As the variability increases, aPCN+ has a steeper decrease in the number of admitted sessions to avoid packet loss.

on a variability of 190 Mbps and higher. When we take a closer look at Figure 11b, we notice that this behavior is indeed needed: at a variability of 190 Mbps and higher the 800 Mbps static configuration starts to experience packet loss, which increases when the variability increases. Hence, aPCN+ decreases its configured rate successfully to avoid packet loss when variability increases.

6.1.3 Impact of the t_{mon} parameters

The aPCN+ algorithm has two parameters, $t_{monpeak}$ and t_{monavg} , that have not been determined through the guidelines described in Section 4.5. The $t_{monpeak}$ parameter represents the aggregation interval used to calculate the maximum bandwidth. The higher $t_{monpeak}$, the more history is taken into account. As only the maximum value is taken into account, a large $t_{monpeak}$ value will not result in a larger detection delay. As such, the exact value of the $t_{monpeak}$ parameter is less critical. In our experiments, we varied the $t_{monpeak}$ value from 8 to 30 seconds and found no significant difference in performance.

The value of the t_{monavg} parameter, the aggregation interval used to measure the average bandwidth, is of more importance: this is illustrated in Figure 12, which shows the impact of an increasing t_{monavg} value on the number of admitted sessions and SSIM score. The figure shows that there is a trade-off in setting this parameter. If the t_{monavg} value is lower than 1.5 seconds, the aggregation interval is too low to smooth out the variable bandwidth measurements. This results in a configured rate, which is variable in nature and ultimately results in over admission and a video quality drop. By increasing the $t_{monpeak}$ value, the SSIM value per sessions remains the same but the number of admitted sessions decreases. This is because a large peak is taken longer into account in the average bandwidth calculation. However, this only affects a small number of sessions.

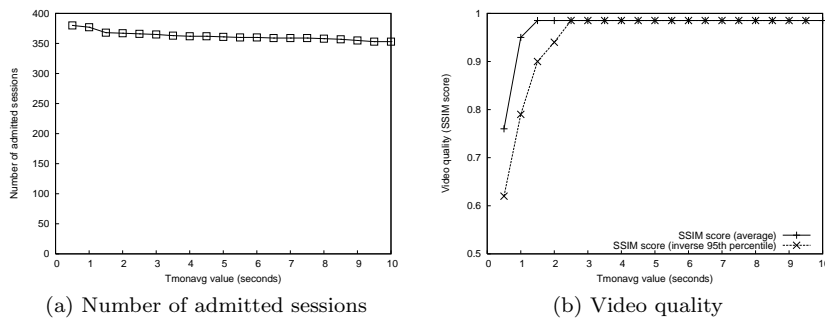


Fig. 12: Impact of the t_{monavg} parameter on the number of admitted sessions and video quality. Setting the value too low, results in over-admission.

Table 1: Overview of the investigated video qualities.

	Video resolution	Average bitrate	SSIM Score
YouTube low	320x240	200 Kbps	0.80
YouTube high	480x360	900 Kbps	0.88
Standard Definition	720x576	2.5 Mbps	0.93
HD Ready	1040x720	8 Mbps	0.95
Full HD	1920x1080	11 Mbps	0.98

6.2 Video quality differentiation

6.2.1 Evaluation scenario

In this section, we investigate how the video quality differentiation function has an impact on the video quality of the received videos. We investigated the transmission of different video qualities ranging from videos used in web streaming to HD videos. Table 1 summarizes the investigated video resolutions, corresponding average bit rate and corresponding SSIM score. These SSIM scores were obtained by taking the average of all SSIM scores per quality level. As in the previous experiments, each video was encoded as an H.264 video and through a constant quality encoding scheme, resulting in a VBR video with a framerate of 25 fps.

We investigated a scenario where, similar to the previous tests, a uniform random distribution of requests for videos arrive. We assume that each client always requests the highest possible video quality. Hence, the admission control mechanism should start with admitting HD videos and switch to lower video qualities once the network load increases. As discussed in Section 5.2, the moment to switch between video qualities should be set as a policy by the network operator in terms of a percentage of the network load. In our experiments, we varied the policies and number of video quality levels. To evaluate the effect on the QoE, we define two metrics: the total SSIM score, obtained by summing up the SSIM scores of all admitted sessions, and the average SSIM

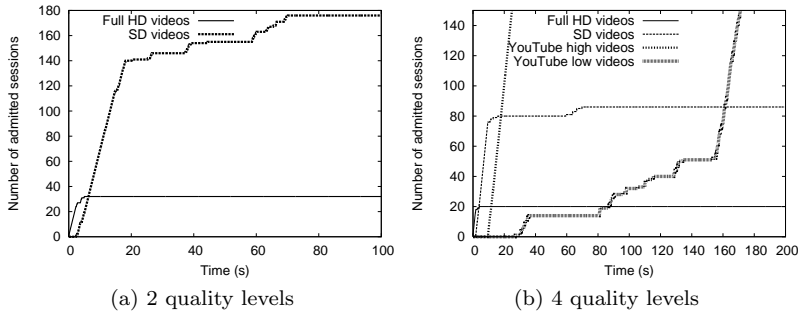


Fig. 13: Evolution of the number of admitted sessions over time using the video quality differentiation mechanism for 2 and 4 different video quality levels.

score per session, obtained by dividing the total SSIM score by the number of admitted session.

6.2.2 Video quality differentiation: Results description

The effect of the video quality differentiation function on the admission control over time is illustrated in Figure 13 for 2 and 4 quality levels. In the case of 2 quality levels, the policy was set to allow 40% of HD traffic and the remainder as SD videos. When using 4 quality levels, each share was set to 25% of the network load. Figure 13 shows that the video quality differentiation function makes a clear switch in decision in moving from one quality level to the other, regardless of the number of video quality levels being used. When using 4 quality levels, occasional new sessions are admitted at the lowest quality if the bandwidth of the aggregate decreases somewhat due to the variability. The transition between switching from admitting one quality level to the other, only takes a few requests. Thus, the video quality differentiation function is able to reach the policy set by the operator.

The most important parameter to set is the policy of the video quality differentiation function. Which configuration performs best depends on the objective a network operator has for the network. When the network load is high, the operator can either favor a large number of clients watching a lower quality video or only a few clients watching at the highest quality. This trade-off is illustrated in Figure 14a, which shows the effect an increasing share of Full HD videos has on the total and average SSIM score for a scenario with two quality levels (Full HD and SD videos). As illustrated, increasing the share of HD videos will have a negative effect on the total SSIM score but will increase the average SSIM score. These results can be used to determine what the share of a certain video quality level should be: typically, an operator should choose which is the lowest average SSIM score the customers should tolerate and check if this average SSIM score does not constrain the total SSIM score the operator targets in the network.

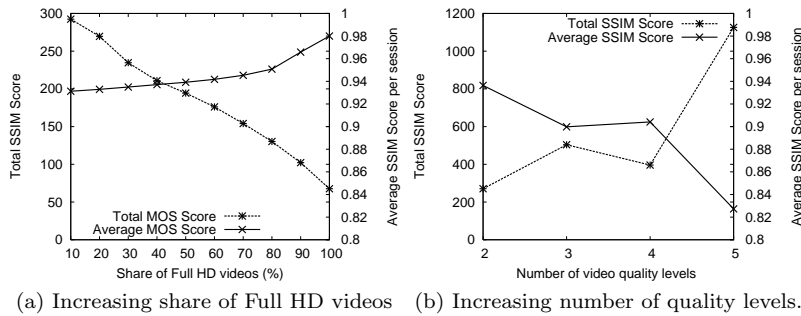


Fig. 14: Influence of an increasing share of Full HD videos (a) and an increasing number of quality levels (b) on the total and average SSIM score.

Table 2: Overview of the investigated video quality levels and corresponding share, stated through policies.

Number of levels	Full HD	HD Ready	SD	YouTube High	YouTube low
2	50%		50%		
3	33,33%	33,33%	33,33%		
4	25%	25%	25%	25%	
5	20%	20%	20%	20%	20%

Figure 14b shows the impact of an increasing number of quality levels on the total and average SSIM scores. In this case, each quality level is assigned an equal share of the link capacity. An overview of which quality levels are used in each situation is given in Table 2. The effect on the average SSIM score can easily be explained: when increasing from 2 to 3 and 4 to 5 quality levels, the newly introduced quality level always has a lower video quality than those already in the system. Therefore, the system will accept more of those videos as well, which results in a drop in average SSIM score. When increasing from 3 to 4 quality levels, the newly introduced quality level (HD Ready) has a SSIM score between those already present in the system (Full HD and SD). As there will be less Full HD and SD sessions admitted in favor of the HD ready sessions, the average SSIM score does not dramatically change.

Increasing the number of quality levels up to three results in an increase in total SSIM score. This is because the difference between the reduction in needed resources of the newly introduced YouTube high quality level is so high that the loss in SSIM score is compensated. As, in this case, a lot more sessions will be admitted with only minor quality drops, the total SSIM score is considerably higher. For the same reason, the total SSIM score drops again when the HD ready option is added as a new quality level. The minor gain in SSIM score obtained by admitting HD ready sessions is compensated by the loss of HD, SD and especially YouTube High sessions. Finally, the total SSIM score increases again considerably because of the introduction of the YouTube Low quality level. The reduction in bandwidth obtained from introducing YouTube Low is so high that the minor loss of SSIM score is compensated.

The results show that introducing a lower quality level to an already existing pool of quality levels increases the total SSIM score. However, it is important to take the average SSIM score into account as well as this also decreases the QoE of each user. This trade-off between a large number of low qualities videos or a small number of high qualities videos, depends on the objective of the operator. The introduced video quality differentiation mechanism, is able to achieve the operator's objective through the policies.

7 Conclusions

We designed and evaluated a measurement based admission control mechanism, specifically targeted at protecting video services. Our mechanism is based on the IETF's PCN architecture but features a number of changes that make PCN more adaptive and able to protect bursty traffic, a traffic type which is currently not studied in the PCN working group.

To investigate how the original PCN mechanism is able to protect bursty traffic we first evaluated PCN's performance in a Video on Demand scenario. We found that the variability of the aggregate, a direct consequence of the use of bursty traffic, complicates PCN's configuration in three ways: (1) finding an accurate aggregation interval is crucial: the aggregation interval should be taken high enough to smooth out oscillations in the measurements (2) the configured rate should be lower than the goal rate to compensate the variability and (3) the delay introduced by the measurement function should be as low as possible to support flash crowds.

The evaluation of PCN's performance led to the introduction of a new measurement algorithm, and a novel algorithm, aPCN+, which adaptively sets the configured rate based on the measured variability. We showed that aPCN+ is superior than any static configuration of PCN as it can both maximize the link utilization and avoid packet loss, regardless of the traffic being transmitted over the network and is still robust against flash crowds.

Finally, we introduced the concept of video quality differentiation in the PCN architecture. This introduces a more fine grained admission control where the network operator can choose how much of the link capacity to reserve for each video quality level, while the requests are admitted at the best qualities possible instead of simply blocking a session if the number of resources available is lower than those requested. We showed that a video quality differentiation based admission control mechanism works seamlessly with the PCN architecture with only a limited overhead. Additionally, with respect to the perceived video quality, the evaluation results show the impact of increasing the share of one quality level on one hand and introducing additional quality levels on the other hand.

Acknowledgements The research was performed partially within the framework of the EUREKA CELTIC RUBENS project. Steven Latré is funded by Ph.D grant of the Fund for Scientific Research, Flanders (FWO-V).

References

1. P. Eardley, "Pre-Congestion Notification (PCN) Architecture." RFC 5559 (Informational), June 2009.
2. ETSI TS 182 019, "Resource and Admission Control Sub-system (RACS); Function Architecture," 2009.
3. "ITU-T Recommendation Y.2111: Resource and admission control functions in next generation networks," 2006.
4. K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet." RFC 2638 (Informational), July 1999.
5. R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview." RFC 1633 (Informational), June 1994.
6. P. Yuan, J. Schlembach, A. Skoe, and E. Knightly, "Design and implementation of scalable edge-based admission control," *Computer Networks*, 2001.
7. J. Yu Qiu, C. Cetinkaya, C. Li, and E. W. Knightly, "Inter-class resource sharing using statistical service envelopes," in *In Proceedings of IEEE Infocom*, pp. 36–42, 1999.
8. J. Rezgui, A. Hafid, and M. Gendreau, "A distributed admission control scheme for wireless mesh networks," in *5th International Conference on Broadband Communications, Networks and Systems. BROADNETS.*, 2008.
9. Z. Xia, W. Hao, I.-L. Yen, and P. Li, "A distributed admission control model for QoS assurance in large-scale media delivery systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, pp. 1143–1153, Dec. 2005.
10. M. Menth and F. Lehrieder, "Performance evaluation of PCN-based admission control," in *16th International Workshop on Quality of Service. IWQoS*, pp. 110–120, 2008.
11. M. Menth and M. Hartmann, "Threshold configuration and routing optimization for PCN-based resilient admission control," *Journal of Computer Networks*, 2009.
12. S. Latré, B. De Vleeschauwer, W. Van de Meerse, 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.
13. S. Latré, B. De Vleeschauwer, W. Van de Meerse, S. Perrault, F. De Turck, P. Demeester, K. De Schepper, C. Hublet, W. Rogiest, S. Custers, and W. Van Leekwijck, "An Autonomic PCN based Admission Control Mechanism for Video Services in Access Networks," in *3rd Workshop on Autonomic Communications and Network Management (ACNM)*, 2009.
14. 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.*, pp. 1–6, August 2008.
15. I. Kofler, R. Kuschig, and H. Hellwagner, "Improving iptv services by h.264/svc adaptation and traffic control," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, 2009. BMSB '09.*, pp. 1–6, May 2009.
16. S. Latré, F. De Turck, B. Dhoedt, and P. Demeester, "Scalable Simulation of QoE Optimization for Multimedia Services over Access Networks," in *The International Conference on Internet Computing (ICOMP)*, 2007.
17. Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication*, vol. 19, pp. 121–132, February 2004.
18. A. Rose, "BBC iPlayer goes HD, adds higher quality streams, releases iPlayer Desktop out of Labs," 2009. [on-line] http://www.bbc.co.uk/blogs/bbcinternet/2009/04/bbc.iplayer.goes.hd.adds_higher.html, Accessed 15th November 2009.