Performance Evaluation of PCN-Based Admission Control

Michael Menth and Frank Lehrieder
University of Würzburg, Institute of Computer Science, Germany

***Abstract*— Pre-congestion notification (PCN) marks packets when the PCN traffic rate exceeds an admissible link rate and this marking information is used as feedback from the network to take admission decisions for new flows. This idea is currently under standardization in the IETF. Different marking algorithms are discussed and various admission control algorithms are proposed that decide based on the packet markings whether further flows should be accepted or blocked. In this paper, we propose a two-layer architecture that makes the coexistence of various algorithms explicit. We propose novel control algorithms, investigate their behavior under various conditions, and compare them with existing approaches.**

I. Introduction

Internet service providers (ISPs) recently offer increased access speeds, e.g., by digital subscriber lines (DSL), cable TV (CATV), and fiber to the home (FTTH). These technologies significantly increased the traffic volume in carrier networks and in 2005, the major traffic in Japan was already produced by residential users [1]. Popular video services like YouTube produce large traffic volumes, but are only weak precursors of high-quality IP-TV services. They present a challenge for ISPs which need to offer triple play, i.e. the integration of the transport of data, voice, and video. However, the resource management for triple play becomes more and more difficult due to the emerging interactive Web2.0 since residential users also become content providers. In particular, [2] has shown that normal users get accustomed with new services, change access technologies, and become "heavy hitters" such that the majority of the overall traffic is produced by a minority of residential users.

Today, ISPs rely on capacity overprovisioning (CO) to enforce quality of service (QoS) in terms of packet loss and delay. However, triple play requires guarantees that cannot be given by CO [3]. In [4] admission control (AC) was proposed for IP networks, but so far such techniques are only applied locally, they are rarely in use, and not deployed in core networks. If congestion occurs in core networks, this is mainly due to failures and redirected traffic, and only to a minor degree due to increased user activity [5]. Thus, both AC and CO require backup capacity that can be used under failure-free conditions to improve the transmission quality [6]. Taking this into account, CO seems a viable alternative to AC in practice for networks with static traffic. However, the dynamic behavior of users and services sketched above leads to an unpredictability of future demands such that QoS provisioning remains difficult. Therefore, ISPs see the need for AC to offer premium services over integrated IP networks in the future.

This work was funded by Nortel Networks, Ottawa, and Deutsche Forschungsgemeinschaft (DFG) under grant TR257/18-2. The authors alone are responsible for the content of the paper.

As a consequence, the "Congestion and Pre-Congestion Notification" (PCN) working group [7] of the Internet Engineering Task Force (IETF) is about to standardize a new lightweight AC for the Internet based on feedback from the network which is called pre-congestion notification (PCN). Each link l of a PCN domain is associated with an admissible rate $AR(l)$ and if the traffic on a link l exceeds the corresponding rate threshold $AR(l)$, its traffic is marked. The egress nodes evaluate the markings of the packets. A new flow is rejected if packets on its prospective path are marked; otherwise, it is accepted. Currently, two different marking algorithms are discussed: exhaustive marking (aka threshold marking [8]) and excess marking. Various AC algorithms are proposed that decide whether further requests should be admitted or blocked: one is based on congestion level estimates (CLE-based AC, CLEBAC) and the other is triggered by the observation of marked packets (observation-based AC, OBAC). As an alternative, probing may be used for AC purposes.

The contribution of this paper is manyfold. We formulate the current concept for PCN-based AC and flow termination (FT) as a two-layer architecture making its modularity more explicit. We present observation-based AC as a new control algorithm for AC. And we investigate the behavior of different AC algorithms in combination with different packet marking mechanisms under various conditions. The results provide valuable input for the standardization of PCN-based AC.

The paper is structured as follows. Section II reviews related work showing the historic roots of PCN. Section III introduces PCN using a new two-layer architecture to separate between marking and AC algorithms. This simplifies the adaptation of PCN-based AC to various application layers. Furthermore, we present different options to instantiate these layers. Section IV studies the behavior of various AC methods based on different marking mechanisms. Finally, Section V summarizes this work and draws conclusions.

II. Related Work

We review related work regarding random early detection (RED), explicit congestion notification (ECN), and stateless core concepts for AC as they can be viewed as historic roots of PCN.

A. Random Early Detection (RED)

RED was originally presented in [9], and in [10] it was recommended for deployment in the Internet. It was designed to detect incipient congestion by measuring a time-dependent average buffer occupation *avg* in routers and to take appropriate countermeasures. That means, packets are dropped or marked to indicate congestion to TCP senders and the probability for that action increases linearly with the average queue length *avg*. The value of *avg* relates to the physical queue size which is unlike PCN metering that relates to the configured admissible or supportable rate.

B. Explicit Congestion Notification

Explicit congestion notification (ECN) is built on the idea of RED to signal incipient congestion to TCP senders in order to reduce their sending window [11]. Packets of non-ECN-capable flows can be differentiated by a "not-ECN-capable transport" (not-ECT, '00') codepoint from packets of a ECN-capable flow which have an "ECN-capable transport" (ECT) codepoint. In case of incipient congestion, RED gateways possibly drop not-ECT packets while they just switch the codepoint of ECT packets to "congestion experienced" (CE, '11') instead of discarding them. This improves the TCP throughput since packet retransmission is no longer needed. Both the ECN encoding in the packet header and the behavior of ECN-capable senders and receivers after the reception of a marked packet is defined in [11]. ECN comes with two different codepoints for ECT: ECT(0) ('10') and ECT(1) ('01'). They serve as nonces to detect cheating network equipment or receivers [12] that do not conform to the ECN semantics. The four codepoints are encoded in the (currently unused) bits of the differentiated services codepoint (DSCP) in the IP header which is a redefinition of the type of service octet [13]. The ECN bits can be redefined by other protocols and [14] gives guidelines for that. This may be useful for the encoding of PCN codepoints, but this aspect is not the focus of this paper.

C. Admission Control

We briefly review some specific AC methods that can be seen as forerunners of the PCN principle.

1) Admission Control Based on Reservation Tickets: To keep a reservation for a flow across a network alive, ingress routers send reservation tickets in regular intervals to the egress routers. Intermediate routers estimate the rate of the tickets and can thereby estimate the expected load. If a new reservation sends probe tickets, intermediate routers forward them to the egress router if they have still enough capacity to support the new flow and the egress router bounces them back to the ingress router indicating a successful reservation; otherwise, the intermediate routers discard the probe tickets and the reservation request is denied. The tickets can also be marked by a packet state. Several stateless core mechanisms work according to this idea [15]–[17].

2) Admission Control Based on Packet Marking: Gibbens and Kelly [18], [19] theoretically investigated AC based on the feedback of marked packets whereby packets are marked by routers based on a virtual queue with configurable bandwidth. This core idea is adopted by PCN. Marking based on a virtual instead of a physical queue also allows to limit the utilization of the link bandwidth by premium traffic to arbitrary values between 0 and 100%. Karsten and Schmitt [20], [21] integrated these ideas into the IntServ framework and implemented a prototype. They point out that the marking can also be based on the CPU usage of the routers instead of the link utilization if this turns out to be the limiting resource for packet forwarding.

3) Resilient Admission Control: Resilient admission control admits only so much traffic that it still can be carried after rerouting in a protected failure scenario [6]. It is necessary since overload in wide area networks mostly occurs due to link failures and not due to increased user activity [5]. It can be implemented with PCN by setting the admissible rate thresholds $AR(l)$ low enough such that the PCN rate $r(l)$ on a link l is lower than the supportable rate threshold $SR(l)$ after rerouting.

III. Admission Control (AC) and Flow Termination (FT) based on Pre-Congestion Notification

We explain the general idea of PCN using the nomenclature of [22] and propose a new two-layer architecture for PCN-based admission control (AC) and flow termination (FT). We present currently discussed mechanisms for the packet marking layer, and review existing and suggest new mechanisms for the AC layers.

A. Pre-Congestion Notification (PCN)

PCN defines a new PCN traffic class that receives preferred treatment by PCN nodes. It provides information to support admission control (AC) and flow termination (FT) for this traffic type. FT is a new control function that tears down already admitted traffic in case of imminent overload which can occur in spite of AC due to rerouted traffic in failure cases or other unexpected events.

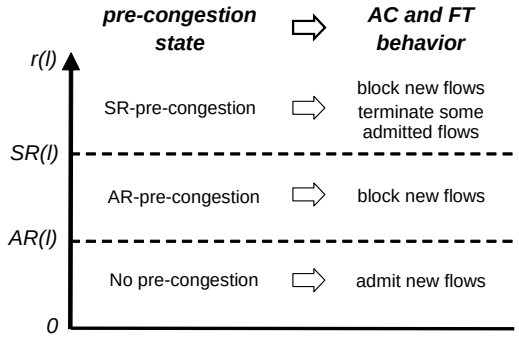


Fig. 1. The admissible and the supportable rate $(AR(l), SR(l))$ define three pre-congestion states concerning the PCN traffic rate $r(l)$ on a link.

PCN introduces an admissible and a supportable rate threshold ($AR(l)$, $SR(l)$) for each link l of the network which imply three different link states as illustrated in Figure 1. If the PCN traffic rate $r(l)$ is below $AR(l)$, there is no pre-congestion and further flows may be admitted. If the PCN traffic rate $r(l)$ is above $AR(l)$, the link is *AR*-pre-congested and the rate

above $AR(l)$ is *AR*-overload. In this state, no further flows should be admitted. If the PCN traffic rate $r(l)$ is above $SR(l)$, the link is *SR*-pre-congested and the rate above $SR(l)$ is *SR*-overload. In this state, some already admitted flows should be terminated. PCN nodes monitor the PCN rate on their links and they remark packets depending on their pre-congestion states. The PCN egress nodes evaluate the packet markings and their essence is reported to the AC and FT entities of the network such that they can take appropriate actions. Therefore, this concept is called pre-congestion notification.

B. Applicability of PCN-Based AC and FT

PCN implements AC and FT for a network with PCN-enabled nodes, i.e. for a so-called PCN domain. It is simple since it does not require per-flow states inside the PCN domain as classical link-by-link reservation protocols do (e.g. RSVP [23]). Therefore, it is an attractive means to perform resource admission control for individual PCN domains on behalf of higher layer end-to-end resource or application signalling protocols or frameworks such as RSVP, SIP, or the IP Multimedia Subsystem (IMS) (cf. Figure 2).

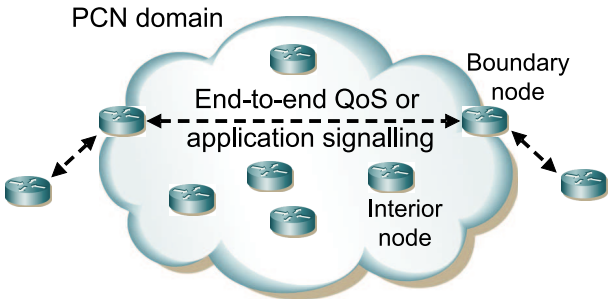


Fig. 2. PCN performs admission control (AC) and flow termination for a single PCN domain. Admission requests are triggered by higher-layer resource signaling protocols.

C. A Two-Layer Architecture for PCN-Based AC and FT

The PCN concept for AC and FT has been presented in [8]. We describe it by a new two-layer approach to make the modularity and decomposability of different network functions more obvious. The concept can be subdivided into a packet marking layer (PML) and an admission control and flow termination layer (ACFTL). The PML associates with each link l of the PCN domain an *AR* and *SR* threshold. Packets are marked when the PCN traffic $r(t)$ on a link l exceeds the corresponding rate threshold. Within a single PCN domain, the same PML must be applied by the PCN nodes, i.e., all PCN nodes require the same marking behavior. However, the parameters for the marking algorithms may be link specific, in particular, every link l can be configured with its own rate thresholds $AR(l)$ and $SR(l)$. The ACFTL is implemented in the PCN ingress and egress nodes: they monitor the markings and decide whether further flows can be accepted or whether already admitted flows need to be terminated. The implementation of the ACFTL may take advantage of specific transport architectures and may be tailored for various signaling architectures with which it needs to interoperate. To support different signaling architectures in a single network, it makes sense to deploy different instances of the ACFTL as long as they respect the semantics of the packet markings and coexist in a fair way. This is depicted in Figure 3. In contrast to the ACFTL, the PML can have only a single implementation in a PCN domain.

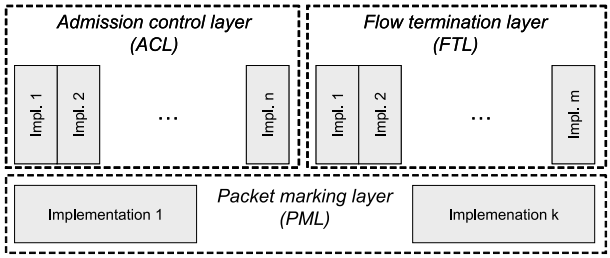


Fig. 3. Within a single network, only one PML must be deployed, but several ACFTLs may coexist.

In the following, we present two different implementations for the packet marking layer (PML) and several algorithms for the AC and FT layer (ACFTL). We consider only the AC part of the ACFTLs in this paper and call it the AC layer (ACL).

D. Methods for the Packet Marking Layer

Packets are marked with a "no-pre-congestion" (NP) codepoint when entering the PCN domain. PCN nodes remark NP-marked packets to "admission-stop" (AS) in case of *AR*-pre-congestion and NP- or AS-marked packets to "excess-traffic" (ET) in case of *SR*-pre-congestion. Various marking behaviors for both objectives exist that mark differently many packets. In the following we discuss two major options for AS-marking that are used by the majority of the current PCN proposals [22], [24], [25]: while excess marking marks only those PCN packets that exceed the admissible rate $AR(l)$ on a link, exhaustive marking marks all packets in case of *AR*-pre-congestion. We use the token bucket principle for their presentation, but there are equivalent virtual queue based formulations [22].

1) Excess Marking: Excess marking uses a token bucket (TB) with a bucket size S and a rate R to control whether the PCN traffic rate exceeds the admissible rate $AR(l)$ on a specific link l. Furthermore, it records the time when the TB was last updated by the variable lU. The variable F tracks the fill state, i.e. the number of tokens in the bucket, and the global variable *now* indicates the current time. Algorithm 1 is called for each packet p. First, the fill state F of the TB is updated and so is lU. If F is smaller than the size B of the packet p, its marking M is set to AS. Otherwise, the number of tokens in the bucket is reduced by the packet size B.

This type of marking behavior is used in the Single Marking (SM) proposal [25], [26] for AS-marking and has the great advantage that it is readily available in today's routers.

2) Exhaustive Marking: The basic operation of exhaustive marking is similar to the one of excess marking. However, packets are marked if the fill state F of the TB is lower than a configured threshold T and tokens are always removed from

Input: token bucket parameters S, R, lU, F, packet size B and marking M, current time *now*

$F = \min(S, F + (now - lU) \cdot R)$;
$lU = now$;
if $(F < B)$ **then**
  $M = \text{AS}$
else
  $F = F - B$;
end if

Algorithm 1: EXCESS MARKING: only those packets exceeding the admissible rate $AR(l)$ are marked.

the bucket if possible. Algorithm 2 explains the mechanism in detail.

Input: token bucket parameters S, R, lU, F, T, packet size B and marking M, current time *now*

$F = \min(S, F + (now - lU) \cdot R)$;
$lU = now$;
if $(F < T)$ **then**
  $M = \text{AS}$
end if
$F = \max(0, F - B)$;

Algorithm 2: EXHAUSTIVE MARKING: all packets are marked if the PCN rate exceeds the admissible rate $AR(l)$.

If the PCN rate exceeds the admissible rate, the tokens are faster consumed than refilled and the fill state of the TB goes to zero and remains small. Therefore, the fill state F stays below the marking threshold T and all packets are marked. Exhaustive marking is applied for AS-marking by the Controlled Load (CL) [24] and the Three State Marking (3sm) proposal [22].

E. Methods for the Admission Control Layer

In the following we describe three fundamentally different AC algorithms that may be implemented in PCN edge nodes of a PCN domain to take admission decisions for new flows based on the received packet markings. Each of them may be applied in combination with excess and exhaustive marking in the packet marking layer.

1) Probe-Based Admission Control (PBAC): With probing, the PCN ingress node generates probe packets upon an admission request. They have the same IP header as future data packets. This is necessary when multipath routing (e.g. ECMP) distributes traffic over several paths depending on a header digest. The PCN egress node intercepts the packets to avoid that they leak out of the PCN domain, evaluates their markings, and communicates the result to the admission control entity. This idea has been pursued in [22] and [27].

a) PBAC with Exhaustive Marking: Exhaustive marking marks no packets with AS when the PCN traffic rate is clearly below $AR(l)$, and it marks all packets if the PCN traffic rate is clearly above $AR(l)$. As a consequence, a single probe packet suffices for exhaustive marking to find out whether the prospective path of a new flow is *AR*-pre-congested. If the probe packet is received by the egress node with AS mark, the new flow is blocked, otherwise it is accepted.

b) PBAC with Excess Marking: Excess marking marks no packets with AS when the PCN traffic rate is clearly below $AR(l)$, but it marks only those packets with AS that exceed the admissible rate if the PCN traffic rate is clearly above $AR(l)$. Thus, only a small fraction of the packets is marked in case of *AR*-pre-congestion. Therefore, several probe packets must be sent for a reliable test indicating whether the path is AR-pre-congested. If the egress node detects one or more marked probe packets, the new flow is rejected, otherwise it is accepted.

2) Observation-Based Admission Control (OBAC): With OBAC, the PCN egress node groups the flows under its control that share the same ingress and egress node into so-called ingress-egress aggregates (IEAs). Each IEA has a state K which is either blocking (*block*) or admitting (*admit*) new flows. The state K is controlled by the PCN egress node and communicated to the AC entitiy of the network (e.g. the ingress node or a central node) using "admission-stop" and "admission-continue" messages that are triggered based on previous packet markings.

OBAC keeps the state K for D^{min}_{block} time in the *block* mode whenever a marked packet is received for the corresponding IEA. This behavior can be technically achieved using a IEA-specific timer T^{min}_{block}. Algorithm 3 needs to be called when a packet arrives and Algorithm 4 when the timer expires. When a non-NP-marked packet is observed (AS- or ET-marked), Algorithm 3 sets the variable lM to the current time to record the instant of the last marked packet. If the IEA state is *admit*, it is switched to *block*, an admission-stop message is sent to the corresponding AC-entity, and the timer is set. Algorithm 4 switches the IEA state back to *admit* and sends an admission-continue message D^{min}_{block} time after the last marked packet was observed. OBAC is very simple. It has only the single configuration parameter D^{min}_{block} which may be IEA-specific and does not require any form of measurement. It can be used in combination with excess and exhaustive marking.

Input: packet marking M, time of last marked packet lM, IEA state K, minimum *block* time D^{min}_{block}, timer T^{min}_{block}, current time *now*

if $(M \neq \text{NP})$ **then**
  $lM = now$;
  if $(K == admit)$ **then**
    $K = block$;
    send admission-stop msg;
    set $(T^{min}_{block}, now + D^{min}_{block})$;
  end if
end if

Algorithm 3: OBSERVATION-BASED AC: routine called upon packet arrival.

Input: time of last marked packet lM, IEA state K, minimum *block* time D^{min}_{block}, timer T^{min}_{block}, current time *now*

if $((now - lM) \geq D^{min}_{block})$ **then**
  if $(K == block)$ **then**
    $K = admit$;
    send admission-continue msg;
  end if
else
  set $(T^{min}_{block}, lM + D^{min}_{block})$;
end if

Algorithm 4: OBSERVATION-BASED AC: routine called upon expiration of timer T^{min}_{block}.

3) CLE-Based Admission Control (CLEBAC): CLEBAC is an alternative algorithm for the egress nodes to trigger changes of the IEA state and to send admission-stop and admission-continue messages to the corresponding ingress nodes or other AC entities.

CLEBAC proceeds in measurement intervals and calculates for every interval a congestion level estimate (CLE) which is the proportion of AS- or ET-marked traffic rate. To that end, each IEA has counters n_{marked} and $n_{unmarked}$ that track the number of marked and unmarked bytes and these counters are updated by Algorithm 5 whenever a packet arrives. A timer T_{MI} indicates the end of a measurement interval whose duration D_{MI} may also be IEA-specific. Algorithm 6 is called when the timer expires. The CLE is computed, the counters are reset and the timer is set to the end of the next measurement interval. If the CLE is at least the CLE admission-stop threshold T^{AStop}_{CLE}, the IEA state is switched from *admit* to *block* or if the CLE is at most the CLE admission-continue threshold T^{ACont}_{CLE}, the IEA state is switched from *block* to *admit*. In addition, the respective control messages are sent.

In contrast to OBAC, CLEBAC has three IEA-specific configuration parameters: D_{MI}, T^{AStop}_{CLE}, and T^{ACont}_{CLE}. It performs measurements and takes into account the marking of every packet. Like OBAC, CLEBAC can be applied both with excess and exhaustive marking. With OBAC and CLEBAC, additional state refreshes may be sent to show the ingress node that the egress is alive or for reliability reasons.

Input: packet marking M, packet size B, counters n_{marked} and $n_{unmarked}$

if $(M == \text{NP})$ **then**
  $n_{unmarked} = n_{unmarked} + B$;
else
  $n_{marked} = n_{marked} + B$;
end if

Algorithm 5: CLEBAC: routine called upon packet arrival.

F. Need for Several Admission Control Layers (ACLs)

ACLs translate the marking results obtained from the packet marking layer to higher layer signalling protocols depending on what information is available from the requesting flow. In the following we discuss different deployment scenarios that call for different ACLs and which implement different AC algorithms. In a multi-service network, it makes sense to support several ACLs to provide resource admission control for various higher layer signalling architectures.

Input: counters n_{marked} and $n_{unmarked}$, measurement interval duration D_{MI}, measurement interval timer T_{MI}, IEA state K, CLE thresholds T^{ACont}_{CLE} and T^{AStop}_{CLE}

$CLE = \frac{n_{marked}}{n_{marked}+n_{unmarked}}$;
$n_{marked} = 0$; $n_{unmarked} = 0$;
set $(T_{MI}, now + D_{MI})$;
if $((K == block) \wedge (CLE \leq T^{ACont}_{CLE}))$ **then**
  $K = admit$;
  send admission-continue msg;
else if $((K == admit) \wedge (CLE \geq T^{AStop}_{CLE}))$ **then**
  $K = block$;
  send admission-stop msg;
end if

Algorithm 6: CLEBAC: routine called upon timeout at the end of a measurement interval.

1) Label Switched Paths: When traffic is carried over label switched paths (LSPs), flows are already classified into IEAs such that PCN egress nodes can classify packets in a rather simple manner when penultimate hop popping is not used. Then it is easy to implement observation- or CLE-based control of the admission process.

2) Arbitrary Flows: In case of arbitrary flows, it is rather hard to associate an admission request with an IEA at the ingress node and to map its packets to the corresponding IEA at the egress node. Therefore, probe-based admission control which does not require the concept of IEAs is possibly a more attractive solution. The ingress node generates probe packets containing information about the ingress node such that the egress node can return the probing result to the correct ingress node. The disadvantage of that method is a possibly long probing delay when several probe packets are required.

3) End-to-End Reservations: In case of end-to-end reservations controlled by RSVP, the initial PATH message can be reused as a probe message. The initial PATH message travels along the prospective path through the PCN domain. However, it is not RSVP-processed by interior PCN nodes since the PCN egress node is the next regular RSVP node processing the message after the PCN ingress node. Nevertheless, the message is subject to the PCN metering and marking process of the interior PCN nodes on its path. If the PATH message is marked, the egress node returns a PATH ERROR message to the ingress node which rejects the reservation. If the PATH message arrives unmarked at the PCN egress node, it is forwarded. Thus, a corresponding RESV message can only return to the PCN ingress node if the probing procedure for the initial PATH message was successful. Therefore, all reservations can be admitted when the first RESV message returns. This is a very lightweight implementation of probing since

no extra probe messages need to be created and intercepted by boundary nodes. However, this approach does not work with excess marking since then probing requires several probe messages.

4) Central Node as Admission Control Entity: In some architectures, a central node is in charge of admitting or blocking flows. They need an adaptation of the above sketched ACLs. In case of AC methods that rely on IEAs, a copy of the state for each IEA is stored at the central node to local decisions, and control messages are sent from the egress nodes to the central node to update the state. With probing, the central node triggers the ingress node to issue probe messages and the egress node reports the results to the central node.

5) Dealing with Flash Crowds and Delayed Media: In practice, flash crowds in the sense of exceptionally large bursts of call arrivals are observed [28]–[31], e.g., in case of normal telephony, tele-voting, file download, or realtime video transmission of sport events. In addition, media start delayed after admission is granted and this delay may be several seconds in case of telephony applications. In such cases, many flows may be accepted before the feedback of previously accepted flows is reflected in the PCN traffic rate and the markings. This may lead to overload when they start sending. One solution to avoid this problem is sending characteristic dummy packets as soon as a flow is admitted in order to reflect the newly admitted flow in load of the network which is used to decide whether more flows can be admitted. The injection of dummy packets may be done by the PCN ingress node, by some application signalling proxy like SIP, or by the application itself. This idea has been proposed in [32]. It is not clear yet whether this concept requires its own ACL.

IV. Performance Evaluation of PCN-Based AC Methods

We investigate the blocking behavior of the three presented AC methods (PBAC, OBAC, and CLEBAC) in combination with excess and exhaustive marking under various load conditions and in the presence of smooth and bursty traffic. We start with a description of the simulation setup and study packet marking probabilities before we analyze the AC methods.

A. Simulation Setup

We assume a single bottleneck link between PCN ingress and egress node such that packets receive potential AS-marks only from this link. This assumption allows to limit the simulation of this scenario to the bottleneck link, i.e. a PCN ingress is connected via a single link with a PCN egress node. The link has an admissible rate of $AR = 8$ Mbit/s and carries n flows with a rate of 80 kbit/s. Simple voice codecs produce strictly periodic traffic with constant packet sizes while other applications like video lead to significantly more traffic variability. Therefore, we use two simple traffic types in our simulations: smooth and bursty traffic. Smooth traffic consists of flows with Gamma-distributed packet inter-arrival times A with a mean of $E[A] = 20$ ms and a coefficient of variation of $c_{var}[A] = 0.1$, i.e. the flows have an almost periodic structure. Packet sizes B are distributed according to the sum of a constant part of 50 bytes and a negative-binomial random variable such that their mean is $E[B] = 200$ bytes and their coefficient of variation $c_{var}[B] = 0.5$. Bursty traffic looks similar, but has $E[A] = 100$ ms and $E[B] = 1000$ bytes. For the production of simulated data we run so many experiments that the confidence intervals for a confidence level of 95% are very small. Therefore, we omit them in the figures.

B. Packet Marking Probabilities

We investigate the impact of the packet size variability and the marking parameters on the packet marking probability p_{AS}.

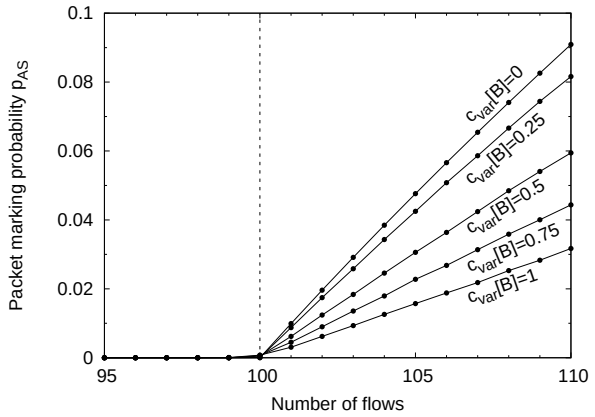
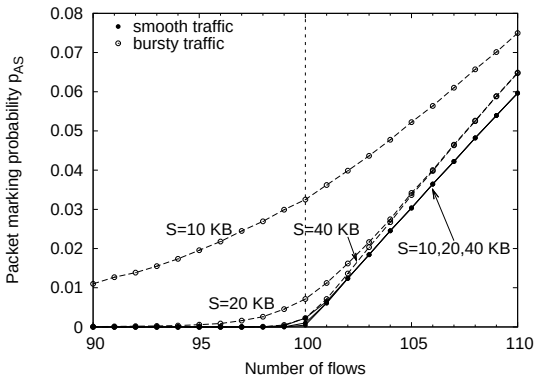


Fig. 4. Impact of the packet size variability on the packet marking probability in case of excess marking for smooth traffic.

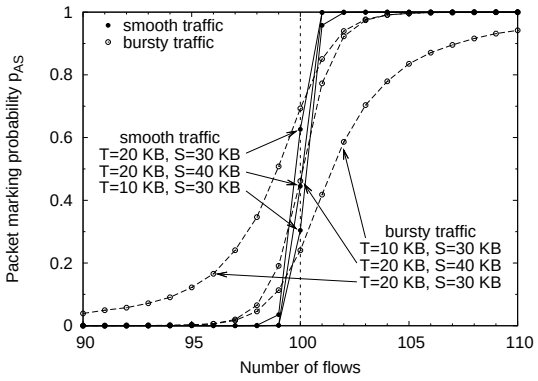
We first consider different packet size variability with excess marking. We set the token bucket size to $S = 40$ KB. We simulate smooth flows, but modify the coefficient of variation $c_{var}[B]$ of the packet sizes from 0 to 1. Figure 4 shows the packet marking probability for different numbers of flows n. 100 flows correspond to a bottleneck utilization of 100% with respect to the admissible rate AR. The figure reveals that excess marking does not mark packets when the link is not pre-congested and that the marking probability rises almost linearly with the number of flows n. The theoretical marking probability is $p_{AS} = \frac{\max(0,n-100)}{n}$, i.e. for 110 flows the theoretical value is $p_{AS} = 0.091$. This value is achieved only for $c_{var}[B] = 0$ while $c_{var}[B] = 0.5$ yields only $p_{AS} = 0.059$ and $c_{var}[B] = 1.0$ only $p_{AS} = 0.031$. Reason for that phenomenon is the fact that the marking probability for large packets is higher than for small packets when excess marking is used. Looking at the percentage of marked bytes, the simulation results meet the theoretical values. As CLEBAC calculates its CLE based on marked bytes, its CLE is not sensitive to packet sizes.

With exhaustive marking, the marking probabilities quickly increase from 0 to 1 when the number of flows increases from 99 to 101. In addition, all packets have the same marking probability as the marking decision in Algorithm 2 is independent of the packet size.

We now study the impact of marking parameters for excess marking. Figure 5(a) shows the impact of the token bucket size S on the packet marking probability for $c_{var}[B] = 0.5$. The curves for smooth traffic are independent of the TB size $S \in$



(a) Excess marking.



(b) Exhaustive marking.

Fig. 5. Impact of marking parameters on the packet marking probabilities for smooth and bursty traffic.

$\{10,20,40\}$ KB. The packet marking probabilities for bursty traffic exceed these curves significantly for TB size of 10 and 20 KB. In particular, packets are already marked when there is no pre-congestion, yet. Only for $S = 40$ KB or larger, packet marking starts only with *AR*-overload and the packet marking probability is independent of the marking parameter S. Thus, the token bucket size must be set to a sufficiently large value.

Figure 5(b) presents the same study for exhaustive marking. For smooth traffic, the marking probability follows the ideal step function when the admissible rate *AR* is exceeded by the PCN rate. The marking threshold T and the size of the token bucket S have hardly any influence on p_{AS}. For bursty traffic, the marking probability depends on the marking parameters T and S. When $S - T$ is too low, it starts marking early, and when T is low, only a fraction of packets is marked for light *AR*-overload. The ideal step function at $n = 100$ is best approximated with $T = 20$ KB and $S = 40$ KB.

In the remainder of this work, we use $S = 40$ KB for excess marking and $T = 20$ KB and $S = 40$ KB for exhaustive marking.

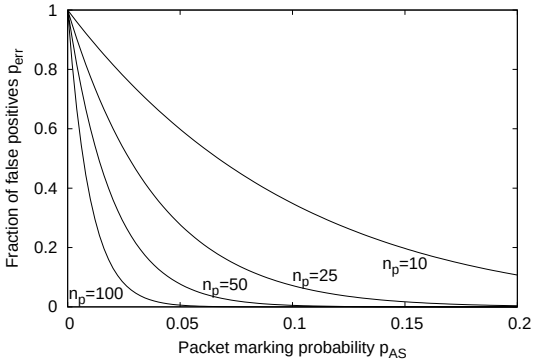
In [33] we compared the packet marking probability of exhaustive marking (aka threshold marking) and ramp marking. We did not consider the impact of any AC method but the impact of many more traffic parameters. Note that [33] uses a virtual queue description of the marking algorithms while we use a token bucket approach in this work. This difference is important when comparing partial results of both studies because the marking threshold T has different semantics for token bucket and virtual queue based markers.

C. Probe-Based Admission Control (PBAC)

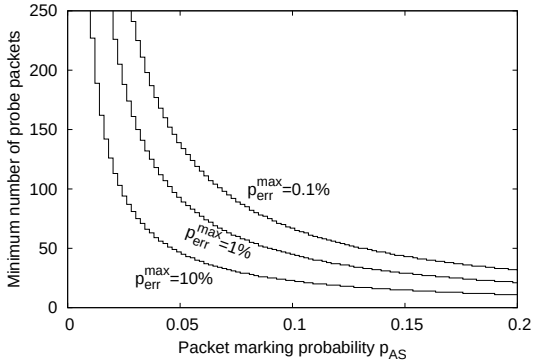
When exhaustive marking is used for probing, only one probe packet is needed for the AC decision. Thanks to the design of Algorithm 2, the marking probability of a packet is independent of its size. Therefore, probe packets can be arbitrarily small and the flow blocking probability is still exactly the packet marking probability presented in Figure 5(b), i.e., all flows are admitted in case of no pre-congestion and blocked in case of *AR*-pre-congestion.

This is different when probing is used in combination with excess marking. Then, several probe packets are required because only a fraction of the packets is marked in case of *AR*-pre-congestion. Furthermore, probe packets must be large enough to face a typical marking probability. We derive the flow blocking probability analytically depending on the marking probability p_{AS} and the number of probe packets n_p. A flow is blocked if at least one out of n_p probe packets is marked:

$$p_{block} = 1 - (1 - p_{AS})^{n_p}. \quad (1)$$



(a) Impact of the number of probes on the fraction of false positives.



(b) Impact of the desired maximum fraction of false positives p_{err}^{max} on the required number of probe packets.

Fig. 6. Probe-based AC (PBAC) with excess marking.

We calculate the fraction of flows that are falsely admitted, i.e. the fraction of false positives, by $p_{err} = 1 - p_{block}$. Figure 6(a) shows this value depending on the packet marking

probability p_{AS} and the number of sent probe packets n_p. If $n_p = 10$ packets are used for probing, the fraction of false positives is unacceptably high like 10% for an *AR*-overload of about 20%. That means, 1 out of 10 requests is admitted such that blocking is not effective if the request rate is high. The error probability is still high for $n_p = 25$ and 50 probes per admission decision. False positives cannot be avoided with PBAC and excess marking, but their fraction can be limited by using a sufficiently large number of probe messages n_p. We compute the required number of probe packets $n_{min}(p_{AS}, p_{err}^{max})$ to meet a maximum fraction of false positives p_{err}^{max} for a marking probability of p_{AS} by

$$\begin{aligned} n_{min}(p_{AS}, p_{err}^{max}) &= \min_{n_p}(p_{err}(p_{AS}, n_p) \leq p_{err}^{max}) \\ &= \lceil \frac{\log(p_{err}^{max})}{\log(1-p_{AS})} \rceil. \end{aligned} \quad (2)$$

Figure 6(b) shows this value depending on the packet marking probability p_{AS} and the maximum fraction of false positives p_{err}^{max}. An error probability of 10% is certainly too high in practice. In the presence of a packet marking probability of $p_{AS} = 0.05$, about $n_{min}(0.05, 0.01) = 90$ probe packets are needed to meet a maximum error probability of $p_{err}^{max} = 0.01$ and $n_{min}(0.05, 0.001) = 140$ probe packets for $p_{err}^{max} = 0.001$. When packet loss is estimated by probe messages, their inter-arrival times should be exponentially distributed [34]. Therefore, we propose that also the inter-arrival time of PCN probe messages should be exponentially distributed instead of sending all probe messages in one shot. However, this introduces significant delay for probing in the presence of excess marking.

D. Observation-Based Admission Control (OBAC)

With OBAC, the PCN egress node switches the IEA state *K* to the *block* mode when it observes a single AS- or ET-marked packet. After an interval of D_{block}^{min} since the observation of the last marked packet, the PCN node switches the IEA state *K* back to the *admit* mode. D_{block}^{min} is OBAC's only configuration parameter.

We simulate the flow blocking probability for the scenario in Section IV-A. The flow blocking probability is the time fraction for which the IEA is in the *block* mode. In Figures 7(a) and 7(b) we study the impact of D_{block}^{min} on the blocking probability of OBAC in combination with excess marking for smooth and bursty traffic. For smooth traffic, OBAC starts blocking only with incipient *AR*-pre-congestion and reliably blocks all new flows when the PCN rate exceeds *AR* by 2% up to 5% depending on D_{block}^{min}. With bursty traffic, OBAC starts blocking already when the PCN rate is slightly below *AR* and reliably blocks all new traffic when the PCN rate exceeds *AR* by 4% up to 12% depending on D_{block}^{min}. The desired behavior of AC is to admit new flows when the PCN rate is below *AR* and to block them when it is above (cf. Section III-A). Apparently, longer minimum *block* intervals lead to a better approximation of the desired behavior. However, D_{block}^{min} should not be chosen too long because otherwise the responsiveness of the system becomes slow such that it cannot continue admitting new flows when old flows have stopped. Another aspect is a smooth operation of the system, i.e., the IEA state *K* should not change too frequently to avoid excessive signalling. At most 2 state changes can occur within D_{block}^{min} time, but Figure 7(c) shows that even the maximum of the average state change rates is lower. Obviously it significantly depends on D_{block}^{min} and clearly decreases with increasing D_{block}^{min}.

Figures 8(a) and 8(b) are analogous to Figures 7(a) and 7(b), but show the blocking of OBAC in combination with exhaustive marking. Exhaustive marking marks more packets and leads, therefore, to larger blocking probabilities under the same conditions. For smooth traffic, there are hardly any false positives or negatives and the minimum *block* interval D_{block}^{min} has no impact on the blocking probability. For bursty traffic, OBAC already blocks a significant fraction of requests when the PCN rate is below *AR* and the impact of D_{block}^{min} is visible. Figure 8(c) shows that exhaustive marking leads to clearly lower state change rates for OBAC than excess marking.

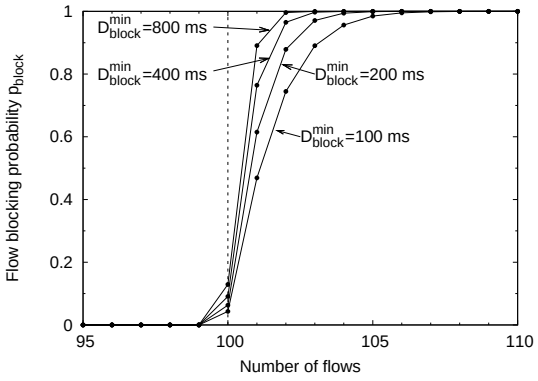
E. CLE-Based Admission Control (CLEBAC)

With CLEBAC, the PCN egress node turns the IEA state *K* to the *block* or *admit* when it measures a CLE at least T_{CLE}^{AStop} or at most T_{CLE}^{ACont}, respectively. These two CLE thresholds and the duration of the measurement intervals D_{MI} are the three configuration parameters of CLEBAC.

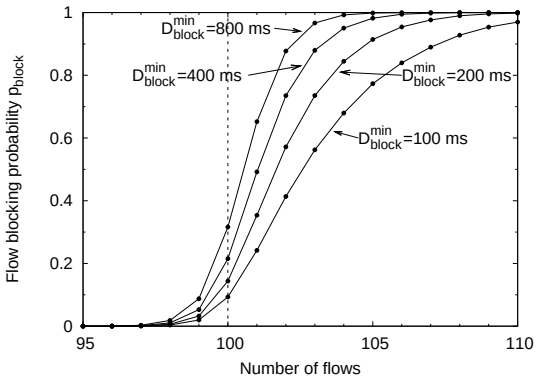
We set $D_{MI} = 200$ ms for excess marking and $D_{MI} = 100$ ms for exhaustive marking, and study the impact of the CLE thresholds T_{CLE}^{AStop} and T_{CLE}^{ACont}. Figures 9(a) and 9(b) show the blocking probability for CLEBAC together with excess marking. We consider the curves with the admission-continue threshold $T_{CLE}^{ACont} = 0$. The curve for $T_{CLE}^{AStop} = 0.01$ is the left-most one and any smaller values for T_{CLE}^{AStop} can hardly move the curve any further to the left (not shown). In contrast, increasing values of T_{CLE}^{AStop} moves the curve significantly to the right. A comparison of the curves $(T_{CLE}^{AStop}, T_{CLE}^{ACont}) \in \{(0, 0.04), (0.1, 0.04)\}$ shows that increasing T_{CLE}^{ACont} moves the curves also to the right. In any case, new flows are rejected only with a sufficiently high probability for at least 2% *AR*-overload. For bursty traffic we get essentially the same results, but the slopes of the curves are not so steep and, as a consequence, there are more false positives.

Figures 10(a) and 10(b) show the average blocking probability for exhaustive marking. For smooth traffic, CLEBAC produces the same blocking probabilities as OBAC with only few false negatives and positives. More bursty traffic leads to slightly more false negatives and positives. In both cases, the curves are independent of the CLE thresholds T_{CLE}^{AStop} and T_{CLE}^{ACont} for a wide range of parameters.

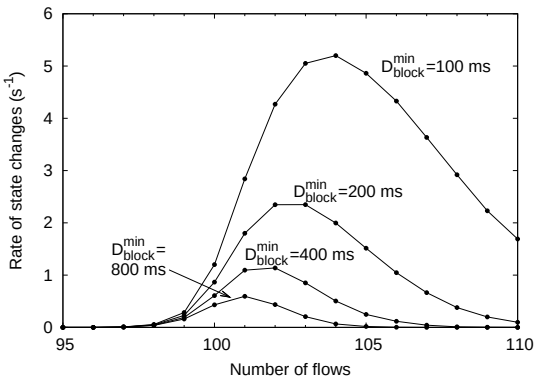
Figures 9(c) and 10(c) show the state change rate for $T_{CLE}^{AStop} = 0.01$ and $T_{CLE}^{ACont} = 0$ in case of excess marking and for $T_{CLE}^{AStop} = 0.9$ and $T_{CLE}^{ACont} = 0.1$ in case of exhaustive marking. To achieve a maximum state change rate of about 1 change per second, the duration of the measurement interval D_{MI} should be 200 ms or longer for excess marking and 50 ms or longer for exhaustive marking.



(a) Impact of the minimum *block* duration D^{min}_{block} for smooth traffic.

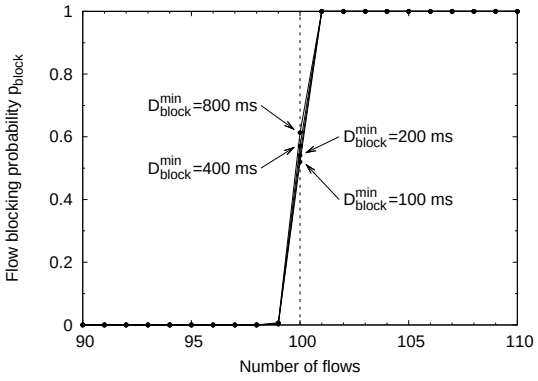


(b) Impact of the minimum *block* duration D^{min}_{block} for bursty traffic.

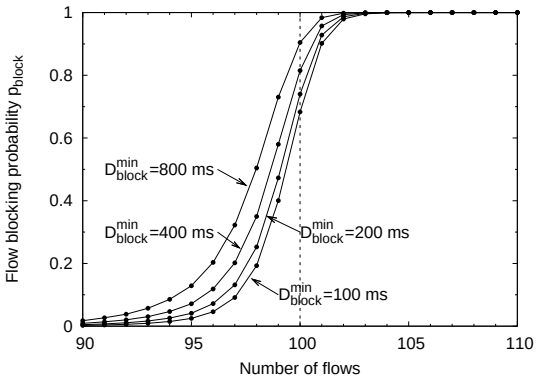


(c) Impact of the marking parameters for bursty traffic.

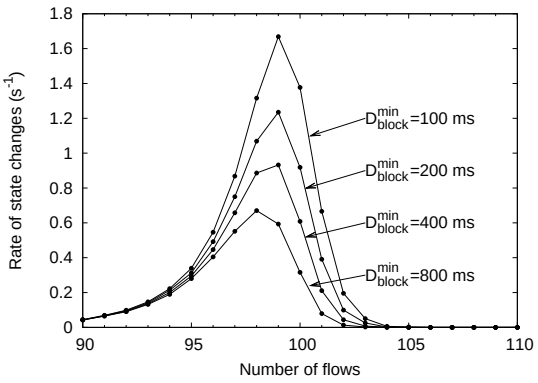
Fig. 7. OBAC and excess marking.



(a) Impact of the minimum *block* duration D^{min}_{block} for smooth traffic.



(b) Impact of the minimum *block* duration D^{min}_{block} for bursty traffic.



(c) Impact of the marking parameters for bursty traffic.

Fig. 8. OBAC and exhaustive marking.

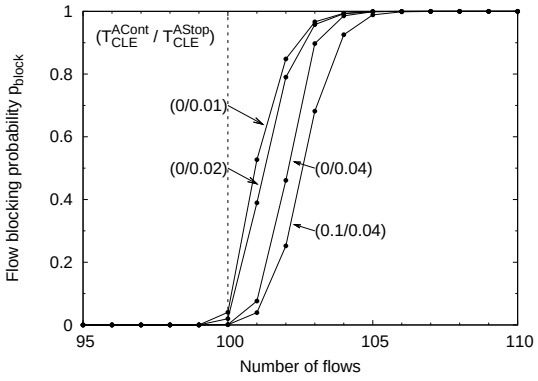
F. Reaction Speed of AC Methods

We consider a link that is suddenly faced with significant overload that may be caused by rerouted traffic in case of a network failure or due to a flash crowd. We are interested in the reaction time of the different AC mechanisms. None of the mechanisms can react before the marking algorithms start AS- or ET-marking. However, their reaction time is rather fast. In case of an overload of k flows with a rate of r_f each, it takes $\frac{S}{k \cdot r_f}$ or $\frac{S-T}{k \cdot r_f}$ time until excess or exhaustive marking start marking packets. For an overload of 100%, this leads to $\frac{40\text{ KB}}{100 \cdot 80\text{kbit/s}} = 4$ ms in our examples for excess marking and to 2 ms for exhaustive marking.

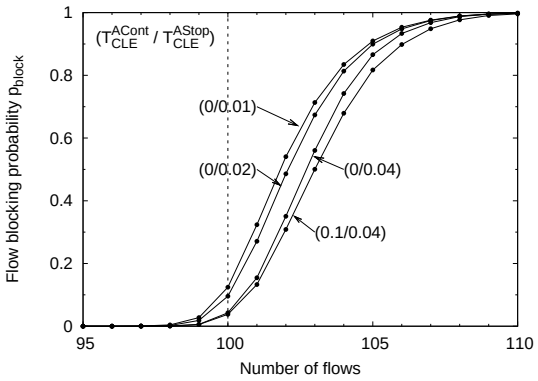
PBAC reacts as soon as packets are marked. OBAC also triggers admission-stop when the first marked packet is observed. In contrast, CLEBAC induces some delay as it sends control messages only at the end of measurement intervals whose duration is D_{MI}. However, the proportion of marked packets might not be large enough in the current measurement interval to trigger admission-stop. Therefore, it takes up to two measurement intervals until the admission process for IEAs going over pre-congested links is reliably stopped.

G. Fair Coexistence of Different AC Methods

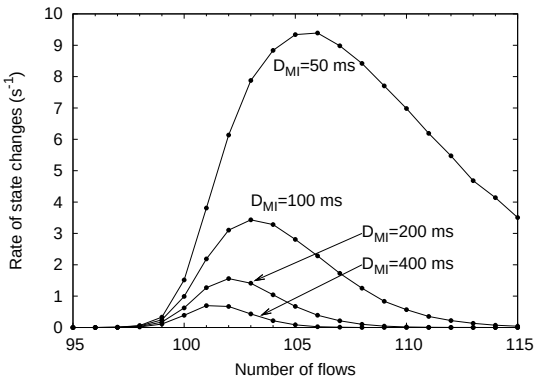
PCN-based AC allows only one marking behavior in a single PCN domain, but several different AC mechanisms



(a) Impact of the T_{CLE}^{ACont} and T_{CLE}^{AStop} parameters for smooth traffic.

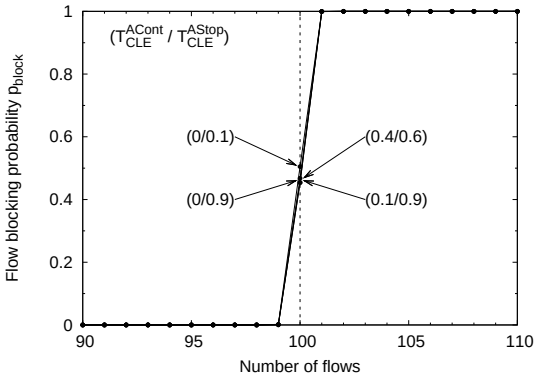


(b) Impact of the T_{CLE}^{ACont} and T_{CLE}^{AStop} parameters for bursty traffic.

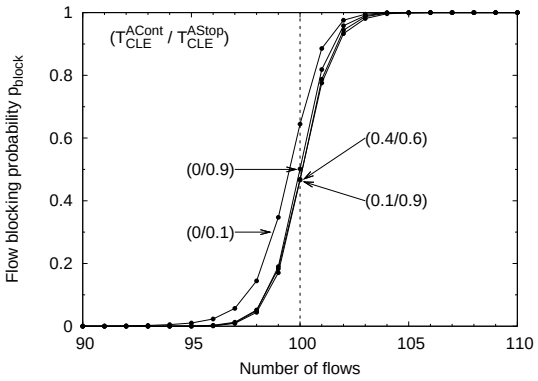


(c) Impact of the measurement interval duration D_{MI} on the state change rate for bursty traffic with $T_{CLE}^{AStop} = 0.01$ and $T_{CLE}^{ACont} = 0$.

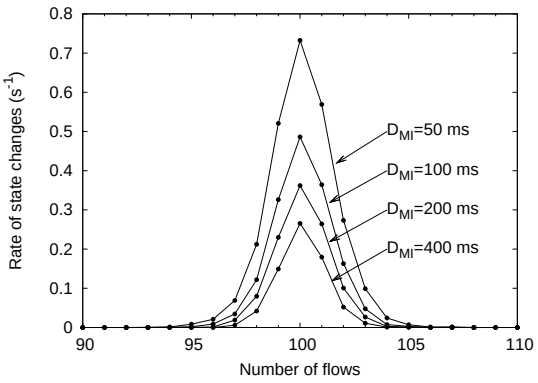
Fig. 9. CLEBAC and excess marking.



(a) Impact of the T_{CLE}^{ACont} and T_{CLE}^{AStop} parameters for smooth traffic.



(b) Impact of the T_{CLE}^{ACont} and T_{CLE}^{AStop} parameters for bursty traffic.



(c) Impact of the measurement interval duration D_{MI} on the state change rate for bursty traffic with $T_{CLE}^{AStop} = 0.9$ and $T_{CLE}^{ACont} = 0.1$.

Fig. 10. CLEBAC and exhaustive marking.

when they coexist in a fair way. For instance, if one AC method blocks flows when the PCN rate equals *AR*, but another AC method starts blocking only when the PCN rate is 5% larger, the PCN rate on the link will be 5% larger than *AR*. If some flows terminate, their freed bandwidth is seized by new flows controlled by the second AC method, thus starving flows subject to the first AC method which is clearly unfair. As a consequence, coexisting AC methods should have similar utilization-dependent flow blocking probabilities, otherwise the coexistence will not be fair.

We achieve similar load-dependent blocking curves for OBAC and CLEBAC in the presence of excess marking when we use a minimum *block* interval of $D_{block}^{min} = 200$ ms for OBAC and CLE thresholds $T_{CLE}^{AStop} = 0.01$ and $T_{CLE}^{ACont} = 0$ for CLEBAC. PBAC yields similar curves for $n_p = 100$ probes. Thus, suitable parameters possibly facilitate a fair coexistence of OBAC and CLEBAC in the same network.

For exhaustive marking, this is easier to achieve because for smooth traffic all AC methods implement a rather steep ascent of the packet marking probability. This steep ascent is softened by bursty traffic for all AC methods. OBAC achieves a similar behavior for small values of D_{block}^{min}, but smaller values than $D_{block}^{min} = 100$ or 200 ms should not be taken to avoid potential oscillation of the IEA state *K*.

V. Conclusion

We have proposed a two layer architecture for PCN-based admission control (AC) and flow termination (FT) that makes the modularity of the PCN concept more explicit by introducing a packet marking layer (PML) and an AC and FT layer (ACFTL). We proposed observation-based AC (OBAC) as new PCN-based AC method and reviewed probing-based AC (PBAC) and the AC method based on congestion level estimates (CLE, CLEBAC). All three methods can be combined with both excess and exhaustive marking and can coexist in the same network if they lead to fair admission results. We studied the blocking probabilities of all six combinations under various load conditions and tested the impact of traffic characteristics and configuration parameters.

With excess marking, the marking probability of a packet depends on its size while the marking probability is independent for exhaustive marking. The packet marking probabilities for both marking algorithms depend on their configured parameters. PBAC requires 50 – 150 probes to reliably admit new flows when excess marking is used while a single probe packet is sufficient in case of exhaustive marking. With excess marking, both OBAC and CLEBAC tend to block traffic at higher load conditions than with exhaustive marking such that they produce similar load-dependent blocking probabilities. This is an important finding as it encourages the idea to use different AC methods within a single network without starving some traffic in a high load regime. Further simulations are required to confirm this hypothesis, in particular for links carrying multiple IEAs where each of them has only a small number of flows.

Acknowledgements

The authors would like to thank Joe Babiarz, Anna Charny, and Francois le Faucheur for their fruitful discussions.

References

[1] K. Fukuda, K. Cho, H. Esaki, and A. Kato, "The Impact of Residential Broadband Traffic on Japanese ISP Backbones," *ACM SIGCOMM Computer Communications Review*, vol. 35, no. 1, pp. 15–22, Jan. 2005.

[2] K. Cho, K. Fukuda, H. Esaki, and A. Kato, "The Impact and Implications of the Growth in Residential User-to-User Traffic," in *ACM SIGCOMM*, Pisa, Italy, Sept. 2006.

[3] D. M. Johnson, "QoS Control versus Generous Dimensioning," *British Telecom Technology Journal*, vol. 23, no. 2, pp. 81–96, Apr. 2005.

[4] S. Shenker, "Fundamental Design Issues for the Future Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1176–1188, Sept. 1995.

[5] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot, "An Approach to Alleviate Link Overload as Observed on an IP Backbone," in *IEEE Infocom*, San Francisco, CA, April 2003.

[6] M. Menth, R. Martin, and J. Charzinski, "Capacity Overprovisioning for Networks with Resilience Requirements," in *ACM SIGCOMM*, Pisa, Italy, Sept. 2006.

[7] IETF Working Group on Congestion and Pre-Congestion Notification (pcn), "Description of the Working Group," http://www.ietf.org/html.charters/pcn-charter.html, Feb. 2007.

[8] P. Eardley (ed.), "Pre-Congestion Notification Architecture," http://www.ietf.org/internet-drafts/draft-ietf-pcn-architecture-03.txt, Feb. 2008.

[9] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[10] B. Braden et al., "RFC2309: Recommendations on Queue Management and Congestion Avoidance in the Internet," Apr. 1998.

[11] K. Ramakrishnan, S. Floyd, and D. Black, "RFC3168: The Addition of Explicit Congestion Notification (ECN) to IP," Sept. 2001.

[12] N. Spring, D. Wetherall, and D. Ely, "RFC3540: Robust Explicit Congestion Notification (ECN)," June 2003.

[13] K. Nichols, S. Blake, F. Baker, and D. L. Black, "RFC2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," Dec. 1998.

[14] S. Floyd, "RFC4774: Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field," Feb. 2007.

[15] W. Almesberger, T. Ferrari, and J.-Y. Le Boudec, "SRP: A Scalable Resource Reservation for the Internet," *Computer Communications*, vol. 21, no. 14, pp. 1200–1211, Nov. 1998.

[16] I. Stoica and H. Zhang, "Providing Guaranteed Services without per Flow Management," in *ACM SIGCOMM*, Boston, MA, Sept. 1999.

[17] R. Szábó, T. Henk, V. Rexhepi, and G. Karagiannis, "Resource Management in Differentiated Services (RMD) IP Networks," in *International Conference on Emerging Telecommunications Technologies and Applications (ICETA 2001)*, Kosice, Slovak Republic, Oct. 2001.

[18] R. J. Gibbens and F. P. Kelly, "Distributed Connection Acceptance Control for a Connectionless Network," in 16^{th} *International Teletraffic Congress (ITC)*, Edinburgh, UK, June 1999, pp. 941 – 952.

[19] F. Kelly, P. Key, and S. Zachary, "Distributed Admission Control," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2617–2628, 2000.

[20] M. Karsten and J. Schmitt, "Admission Control based on Packet Marking and Feedback Signalling – Mechanisms, Implementation and Experiments," Darmstadt University of Technology, Technical Report 03/2002, 2002.

[21] ——, "Packet Marking for Integrated Load Control," in *IFIP/IEEE Symposium on Integrated Management (IM)*, 2005.

[22] J. Babiarz, X.-G. Liu, K. Chan, and M. Menth, "Three State PCN Marking," http://www.ietf.org/internet-drafts/draft-babiarz-pcn-3sm-01.txt, Nov. 2007.

[23] B. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "RFC2205: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification," Sept. 1997.

[24] B. Briscoe, P. Eardley, D. Songhurst, F. L. Faucheur, A. Charny, J. Babiarz, K. Chan, S. Dudley, G. Karagiannis, A. Bader, and L. Westberg, "An Edge-to-Edge Deployment Model for Pre-Congestion Notification: Admission Control over a DiffServ Region," http://www.cs.ucl.ac.uk/staff/bbriscoe/projects/ipe2eqos/gqs/papers/draft-briscoe-tsvwg-cl-architecture-04.txt, Oct. 2006.

[25] A. Charny, F. L. Faucheur, V. Liatsos, and J. Zhang, "Pre-Congestion Notification Using Single Marking for Admission and Pre-emption," http://www.ietf.org/internet-drafts/draft-charny-pcn-single-marking-03.txt, Nov. 2007.

[26] J. Zhang, A. Charny, V. Liatsos, and F. L. Faucheur, "Performance Evaluation of CL-PHB Admission and Pre-emption Algorithms," http://www.ietf.org/internet-drafts/draft-zhang-pcn-performance-evaluation-02.txt, July 2007.

[27] L. Westberg, A. B. an A. Bader, and G. Karagiannis, "LC-PCN: The Load Control PCN Solution," http://www.ietf.org/internet-drafts/draft-westberg-pcn-load-control-03.txt, Feb. 2008.

[28] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites," in *International World Wide Web Conference (WWW)*, Honolulu, Hawaii, USA, May 2002.

[29] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. E. Long, "Managing Flash Crowds on the Internet," in *International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Orlando, FL, USA, Oct. 2003.

[30] ——, "Modeling, Analysis and Simulation of Flash Crowds on the Internet," Jack Baskin School of Engineering, University of California, Santa Cruz, Technical Report, No. UCSC-CRL-03-15, Feb. 2004.

[31] D. S. Seibel (Broadcasting & Cable), "American Idol Outrage: Your Vote Doesn't Count," http://www.broadcastingcable.com/article/CA417981.html, May 2004.

[32] J. Babiarz, K. Chan, G. Karagiannis, and P. Eardley, "SIP Controlled Admission and Preemption," http://www.ietf.org/internet-drafts/draft-babiarz-pcn-sip-cap-00.txt, Oct. 2006.

[33] M. Menth and F. Lehrieder, "Comparison of Marking Algorithms for PCN-Based Admission Control," in 14^{th} *GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB)*, Dortmund, Germany, Mar. 2008.

[34] J. Sommers, P. Barford, N. G. Duffield, and A. Ron, "Improving Accuracy in End-to-End Packet Loss Measurement," in *ACM SIGCOMM*, 2005.