Pre-Congestion Notification Using Single Marking for Admission and Termination
draft-charny-pcn-single-marking-02.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on January 10, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

Pre-Congestion Notification described in [I-D.eardley-pcn-architecture]draft-eardly-pcn-architecture-00 and
earlier in [I-D.briscoe-tsvwg-cl-architecture] approach proposes the use of an Admission Control mechanism to limit the amount of real-time PCN traffic to a configured level during the normal operating conditions, and the use of a Flow Termination mechanism to tear-down some of the flows to bring the PCN traffic level down to a desirable amount during unexpected events such as network failures, with the goal of maintaining the QoS assurances to the remaining flows. In [I-D.eardley-pcn-architecture], Admission and Flow Termination use two different markings and two different metering mechanisms in the internal nodes of the PCN region. This draft proposes a mechanism using a single marking and metering for both Admission and Flow Termination, and presents a preliminary analysis of the tradeoffs. A side-effect of this proposal is that a different marking and metering Admission mechanism than that proposed in [I-D.eardley-pcn-architecture] may be also feasible, and may result in a number of benefits. In addition, this draft proposes a migration path for incremental deployment of this approach as an intermediate step to the dual-marking approach.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].
Table of Contents

1.  Introduction .................................................. 5
   1.1.  Changes from -01 version ............................ 5
   1.2.  Terminology ............................................. 5
   1.3.  Background and Motivation ........................... 5

2.  The Single Marking Approach ................................. 7
   2.1.  High Level description ................................ 7
   2.2.  Operation at the PCN-interior-node .................. 8
   2.3.  Operation at the PCN-egress-node .................... 8
   2.4.  Operation at the PCN-ingress-node ................... 8
   2.4.1.  Admission Decision ................................. 8
   2.4.2.  Flow Termination Decision .......................... 9


4.  Impact on PCN Architectural Framework ....................... 10
   4.1.  Impact on the PCN-Internal-Node ..................... 11
   4.2.  Impact on the PCN-boundary nodes .................... 11
   4.2.1.  Impact on PCN-Egress-Node ........................ 11
   4.2.2.  Impact on the PCN-Ingress-Node ................... 12
   4.3.  Summary of Proposed Enhancements Required for Support
         of Single Marking Options ............................ 13
   4.4.  Proposed Optional Renaming of the Marking and Marking
         Thresholds ............................................. 13
   4.5.  An Optimization Using a Single Configuration
         Parameter for Single Marking ........................ 14

5.  Incremental Deployment Considerations ....................... 14

6.  Tradeoffs, Issues and Limitations of Single Marking
    Approach .................................................... 16
   6.1.  Restrictions on Termination-to-admission Thresholds .... 49
   6.2.  Assumptions on Loss .................................... 50
   6.3.  Effect of Reaction Timescale of Admission Mechanism .... 50
   6.4.  Performance Implications and Tradeoffs ................. 50
   6.5.  Effect on Proposed Anti-Cheating Mechanisms ........... 51
   6.6.  ECMP Handling .......................................... 51
   6.7.  Traffic Engineering Considerations .................... 52

7.  Performance Evaluation Comparison ............................ 55
   7.1.  Relationship to other drafts .......................... 55
   7.2.  Limitations, Conclusions and Direction for Future Work .. 55
   7.2.1.  High Level Conclusions ............................. 56
   7.2.2.  Future work ......................................... 57

8.  Appendix A: Simulation Details ............................... 57
   8.1.  Network and Signaling Models ........................ 57
   8.2.  Traffic Models ......................................... 60
   8.2.1.  Voice Traffic Models ............................... 60
   8.2.2.  "Synthetic Video": High Rate ON-OFF traffic with
           Video-like Mean and Peak Rates ("SVD") .......... 61
   8.2.3.  Real Video Traces (VTR) ............................ 62
   8.3.  Randomization of Base Traffic Models .................. 63
8.4. Parameter Settings .............................................. 63
  8.4.1. Queue-based settings ........................................ 63
  8.4.2. Token Bucket Settings ...................................... 63
8.5. Simulation Details ................................................. 64
  8.5.1. Sensitivity to EWMA weight and CLE ......................... 64
  8.5.2. Effect of Ingress-Egress Aggregation ....................... 66
  8.5.3. Effect of Multiple Bottlenecks .............................. 72
9. Appendix B. Controlling The Single Marking Configuration
    with a Single Parameter ......................................... 75
  9.1. Details of the Proposed Enhancements to PCN
      Architecture .................................................. 75
    9.1.1. PCN-Internal-Node ......................................... 76
    9.1.2. PCN-Egress-Node .......................................... 76
    9.1.3. PCN-Ingress-Node .......................................... 77
    9.2. Impact on PCN-Egress-Node ................................... 78
10. Security Considerations ........................................... 79
11. References .......................................................... 79
    11.1. Normative References ........................................ 79
    11.2. Informative References ...................................... 79
    11.3. References .................................................. 80
Authors’ Addresses ..................................................... 80
Intellectual Property and Copyright Statements .......................... 82
1. Introduction

1.1. Changes from -01 version

- Added miscellaneous clarifications based on comments received on version -01
- Removed Terminology section and replaced it with a pointer to [I-D.eardley-pcn-architecture].
- Added a section on standards implications and considerations for incremental deployment
- Added a section on ECMP handling
- Added a section on traffic engineering considerations and tradeoffs.
- Undated the Appendix to include new results and consolidate some of the old ones

1.2. Terminology

This draft uses the terminology defined in [I-D.eardley-pcn-architecture]

1.3. Background and Motivation

Pre-Congestion Notification [I-D.eardley-pcn-architecture] approach proposes to use an Admission Control mechanism to limit the amount of real-time PCN traffic to a configured level during the normal operating conditions, and to use a Flow Termination mechanism to tear-down some of the flows to bring the PCN traffic level down to a desirable amount during unexpected events such as network failures, with the goal of maintaining the QoS assurances to the remaining flows. In [I-D.eardley-pcn-architecture], Admission and Flow Termination use two different markings and two different metering mechanisms in the internal nodes of the PCN region. Admission Control algorithms for variable-rate real-time traffic such as video have traditionally been based on the observation of the queue length, and hence re-using these techniques and ideas in the context of pre-congestion notification is highly attractive, and motivated the virtual-queue-based marking and metering approach specified in [I-D.briscoe-tsvwg-cl-architecture] for Admission. On the other hand, for Flow Termination, it is desirable to know how many flows need to be terminated, and that in turn motivates rate-based Flow Termination metering. This provides some motivation for employing different metering algorithm for Admission and for Flow Termination.
Furthermore, it is frequently desirable to trigger Flow Termination at a substantially higher traffic level than the level at which no new flows are to be admitted. There are multiple reasons for the requirement to enforce a different configured-admissible-rate and configured-termination-rate. These include, for example:

- End-users are typically more annoyed by their established call dying than by getting a busy tone at call establishment. Hence decisions to terminate flows may need to be done at a higher load level than the decision to stop admitting.

- There are often very tight (possibly legal) obligations on network operators to not drop established calls.

- Voice Call Routing often has the ability to route/establish the call on another network (e.g., PSTN) if it is determined at call establishment that one network (e.g., packet network) can not accept the call. Therefore, not admitting a call on the packet network at initial establishment may not impact the end-user. In contrast, it is usually not possible to reroute an established call onto another network mid-call. This means that call Termination can not be hidden to the end-user.

- Flow Termination is typically useful in failure situations where some loads get rerouted thereby increasing the load on remaining links. Because the failure may only be temporary, the operator may be ready to tolerate a small degradation during the interim failure period. This also argues for a higher configured-termination-rate than configured-admissible-rate

- A congestion notification based Admission scheme has some inherent inaccuracies because of its reactive nature and thus may potentially over admit in some situations (such as burst of calls arrival). If the Flow Termination scheme reacted at the same rate threshold as the Admission, calls may get routinely dropped after establishment because of over admission, even under steady state conditions.

These considerations argue for metering for Admission and Flow Termination at different traffic levels and hence, implicitly, for different markings and metering schemes.

Different marking schemes require different codepoints. Thus, such separate markings consume valuable real-estate in the packet header, especially scarce in the case of MPLS Pre-Congestion Notification [I-D.davie-ecn-mpls]. Furthermore, two different metering techniques involve additional complexity in the data path of the internal routers of the PCN-domain.
To this end, [I-D.briscoe-tsvwg-cl-architecture] proposes an approach, referred to as "implicit Preemption marking" in that draft, that does not require separate termination-marking. However, it does require two separate measurement schemes: one measurement for Admission and another measurement for Flow Termination. Furthermore, this approach mandates that the configured-termination-rate be equal to a drop rate. This approach effectively uses dropping as the way to convey information about how much traffic can "fit" under the configured-termination-rate, instead of using a separate termination marking. This is a significant restriction in that it results in flow termination only taking effect once packets actually get dropped.

This document presents an approach that allows the use of a single PCN marking and a single metering technique at the internal devices without requiring that the dropping and flow termination thresholds be the same. We also argue that this approach can be used as intermediate step in implementation and deployment of a full-fledged dual-marking PCN implementation.

2. The Single Marking Approach

2.1. High Level description

The proposed approach is based on several simple ideas:

- **Replace virtual-queue-based marking for Admission Control by excess rate marking:**
  * meter traffic exceeding the configured-admissible-rate and mark *excess* traffic (e.g. using a token bucket with the rate configured with the rate equal to configured-admissible-rate)
  * at the PCN-boundary-node, stop admitting traffic when the fraction of marked traffic for a given edge-to-edge aggregate exceeds a configured threshold (e.g. stop admitting when 3% of all traffic in the edge-to-edge aggregate received at the ingress is marked)

- **Impose a PCN-domain-wide constraint on the ratio U between the configured-admissible-rate on a link and level of the PCN load on the link at which Flow Termination needs to be triggered (but do not explicitly configure configured-termination-rate).** For example, one might impose a policy that Flow Termination is triggered when PCN traffic exceeds 120% of the configured-admissible-rate on any link of the PCN-domain).
The remaining part of this section describes the possible operation of the system.

2.2. Operation at the PCN-interior-node

The PCN-interior-node meters the aggregate PCN traffic and marks the excess rate. A number of implementations are possible to achieve that. A token bucket implementation is particularly attractive because of its relative simplicity, and even more so because a token bucket implementation is readily available in the vast majority of existing equipment. The rate of the token bucket is configured to correspond to the configured-admissible-rate, and the depth of the token bucket can be configured by an operator based on the desired tolerance to PCN traffic burstiness.

Note that no configured-termination-rate is explicitly configured at the PCN-interior-node, and the PCN-interior-node does nothing at all to enforce it. All marking is based on the single configured rate threshold (configured-admissible-rate).

2.3. Operation at the PCN-egress-node

The PCN-egress-node measures the rate of both marked and unmarked traffic on a per-ingress basis, and reports to the PCN-ingress-node two values: the rate of unmarked traffic from this ingress node, which we deem Sustainable Admission Rate (SAR) and the Congestion Level Estimate (CLE), which is the fraction of the marked traffic received from this ingress node. Note that Sustainable Admission Rate is analogous to the sustainable Preemption rate of [I-D.briscoe-tsvwg-cl-architecture], except in this case it is based on the configured-admissible- rather than termination threshold, while the CLE is exactly the same as that of [I-D.briscoe-tsvwg-cl-architecture]. The details of the rate measurement are outside the scope of this draft.

2.4. Operation at the PCN-ingress-node

2.4.1. Admission Decision

Just as in [I-D.briscoe-tsvwg-cl-architecture], the admission decision is based on the CLE. The ingress node stops admission of new flows if the CLE is above a pre-defined threshold (e.g. 3%). Note that although the logic of the decision is exactly the same as in the case of [I-D.briscoe-tsvwg-cl-architecture], the detailed semantics of the marking is different. This is because the marking used for admission in this proposal reflects the excess rate over the configured-admissible-rate, while in [I-D.briscoe-tsvwg-cl-architecture], the marking is based on
exceeding a virtual queue threshold. Notably, in the current proposal, if the average sustained rate of admitted traffic is 5% over the admission threshold, then 5% of the traffic is expected to be marked, whereas in the context of [I-D.briscoe-tsvwg-cl-architecture] a steady 5% overload should eventually result in 100% of all traffic being admission marked. A consequence of this is that for "smooth" constant-rate traffic, the approach presented here will not mark any traffic at all until the rate of the traffic exceeds the configured admission threshold by the amount corresponding to the chosen CLE threshold.

At first glance this may seem to result in a violation of the pre-congestion notification premise that attempts to stop admission before the desired traffic level is reached. However, in reality one can simply embed the CLE level into the desired configuration of the admission threshold. That is, if a certain rate \( X \) is the actual target admission threshold, then one should configure the rate of the metering device (e.g. the rate of the token bucket) to \( X - y \) where \( y \) corresponds to the level of CLE that would trigger admission blocking decision.

A more important distinction is that virtual-queue based marking reacts to short-term burstiness of traffic, while the excess-rate based marking is only capable of reacting to rate violations at the timescale chosen for rate measurement. Based on our investigation, it seems that this distinction is not crucial in the context of PCN when no actual queuing is expected even if the virtual queue is full. More discussion on this is presented later in the draft.

### 2.4.2. Flow Termination Decision

When the ingress observes a non-zero CLE and Sustainable Admission Rate (SAR), it first computes the Sustainable Termination Rate (STR) by simply multiplying SAR by the system-wide constant \( u \), where \( u \) is the system-wide ratio between Preemption and admission thresholds on all links in the PCN domain: \( STR = SAR \times U \). The PCN-ingress-node then performs exactly the same operation as is proposed in [I-D.briscoe-tsvwg-cl-architecture] with respect to STR: it preempts the appropriate number of flows to ensure that the rate of traffic it sends to the corresponding egress node does not exceed STR. Just as in the case of [I-D.briscoe-tsvwg-cl-architecture], an implementation may decide to slow down the termination process by preempting fewer flows than is necessary to cap its traffic to STR by employing a variety of techniques such as safety factors or hysteresis. In summary, the operation of Termination at the ingress node is identical to that of [I-D.briscoe-tsvwg-cl-architecture], with the only exception that the sustainable Termination rate is computed from the sustainable admission rate rather than derived from a separate
marking. As discussed earlier, this is enabled by imposing a system-wide restriction on the termination-to-admission thresholds ratio and changing the semantics of the admission marking.


The following is a summary of benefits associated with enabling the Single Marking Approach. Some tradeoffs will be discussed in section 7 below.

- Reduced implementation requirements on core routers due to a single metering implementation instead of two different ones.

- Ease of use on existing hardware: given that the proposed approach is particularly amenable to a token bucket implementation, the availability of token buckets on virtually all commercially available routers makes this approach especially attractive.

- Enabling incremental implementation and deployment of PCN (see section 4).

- Reduced number of codepoints which need to be conveyed in the packet header. If the PCN-bits used in the packet header to convey the congestion notification information are the ECN-bits in an IP core and the EXP-bits in an MPLS core, those are very expensive real-estate. The current proposals need 5 codepoints, which is especially important in the context of MPLS where there is only a total of 8 EXP codepoints which must also be shared with DiffServ. Eliminating one codepoint considerably helps.

- A possibility of using a token-bucket-, excess-rate- based implementation for admission provides extra flexibility for the choice of an admission mechanism, even if two separate markings and thresholds are used.

Subsequent sections argue that these benefits can be achieved with a relatively minor enhancements to the proposed PCN architecture as defined in [I-D.eardley-pcn-architecture], allow simpler implementations at the PCN-interior nodes, and trivial modifications at the PCN-boundary nodes.

4. Impact on PCN Architectural Framework

The goal of this section is to propose several minor changes to the PCN architecture framework as currently described in [I-D.eardley-pcn-architecture] in order to enable the single marking.
4.1. Impact on the PCN-Internal-Node

No changes are required to the PCN-internal-node in architectural framework in [I-D.eardley-pcn-architecture] in order to support the Single Marking Proposal. The current architecture [I-D.eardley-pcn-architecture] already allows only one marking and metering scheme rather than two by supporting either "admission only" or "termination only" functionality. To support the Single Marking proposal a single threshold (i.e. Configured-termination-rate) must be configured at the PCN-internal-node, and excess-rate marking as described in should be used to mark packets as described in [I-D.briscoe-tsvwg-cl-architecture]. Note however that the meaning of this single threshold and the marking in this case is no related to termination function, but to admission function. The configuration parameter(s) described in section 4.2 below at the PCN-ingress-nodes and PCN-egress-node will determine whether the marking should be interpreted as the admission-marking (as appropriate for the Single Marking approach) or as termination-marking (as appropriate for the dual marking approach of [I-D.briscoe-tsvwg-cl-architecture]).

We note that from the implementation standpoint, a PCN-ingress-node supporting Single Marking implements only a subset of the functionality needed for Dual Marking.

4.2. Impact on the PCN-boundary nodes

We propose an addition of one global configuration parameter MARKING_MODE to be used at all PCN boundary nodes. IF MARKING_MODE = DUAL_MARKING, the behavior of the appropriate PCN-boundary-node as described in the current version of [I-D.eardley-pcn-architecture]. If MARKING_MODE = SINGLE_MARKING, the behavior of the appropriate boundary nodes is as described in the subsequent subsections.

4.2.1. Impact on PCN-Egress-Node

If MARKING_MODE=SINGLE_MARKING, the Congestion-Level_Estimete (CLE) is measured against termination-marked packets. If MARKING_MODE=DUAL_MARKING, the CLE is measured against admission_marked packets. The method of measurement does not depend on the choice of the marking against which the measurement is performed.

Regardless of the setting of the MARKING_MODE parameter, Sustainable-Aggregate-Rate is measured against termination_marked packets, as currently defined in [I-D.briscoe-tsvwg-cl-architecture].
We note that from the implementation point of view, the same two functions (measuring the CLE and measuring the Sustainable-Aggregate-Rate are required by both the Single Marking approach and the approach in, so the difference in the implementation complexity of the PCN-egress-node is quite negligible.

4.2.2. Impact on the PCN-Ingress-Node

If MARKING_MODE=DUAL_MARKING, the PCN-ingress-node behaves exactly as described in [I-D.eardley-pcn-architecture]. If MARKING_MODE = SINGLE_MARKING, then an additional global parameter U is defined. U must be configured at all PCN_ingress nodes and has the meaning of the desired ratio between the traffic level at which termination should occur and the desired admission threshold, as described in section 2.4 above. The value of U must be greater than or equal to 1. The value of this constant U is used to multiply the Sustainable Aggregate Rate received from a given PCN-egress-node to compute the rate threshold used for flow termination decisions.

In more detail, if MARKING_MODE=SINGLE_MARKING, then

- A PCN-ingress-node receives CLE and/or Sustainable Aggregate Rate from each PCN-egress-node it has traffic to. This is fully compatible with PCN architecture as described in [I-D.eardley-pcn-architecture].

- A PCN-ingress-node bases its admission decisions on the value of CLE. Specifically, once the value of CLE exceeds a configured threshold, the PCN-ingress-node stops admitting new flows. It restarts admitting when the CLE value goes down below the specified threshold. This is fully compatible with PCN architecture as described in [I-D.eardley-pcn-architecture].

- A PCN-ingress node receiving a Sustainable Rate from a particular PCN-egress node measures its traffic to that egress node. This again is fully compatible with PCN architecture as described in draft-earley-pcn-architecture-00.

- The PCN-ingress-node computes the desired Termination Rate to a particular PCN-egress-node by multiplying the Sustainable Aggregate Rate from a given PCN-egress-node by the value of the configuration parameter U. This computation step represents a proposed change to the current version of [I-D.eardley-pcn-architecture].

- Once the Termination Rate is computed, it is used for the flow termination decision in a manner fully compatible with [I-D.eardley-pcn-architecture]. Namely the PCN-ingress-node
compares the measured traffic rate destined to the given PCN-egress-node with the computed Termination rate for that egress node, and terminates a set of traffic flows to reduce the rate exceeding that Termination rate. This is fully compatible with [I-D.eardley-pcn-architecture].

We note that as in the case of the PCN-egress-node, the change in the implementation of the PCN-ingress-node to support Single Marking is quite negligible (a single multiplication per ingress rate measurement interval for each egress node).

4.3. Summary of Proposed Enhancements Required for Support of Single Marking Options

The enhancements to the PCN architecture as defined in [I-D.eardley-pcn-architecture], in summary, amount to:

- defining a global (within the PCN domain) configuration parameter MARKING_MODE at PCN-boundary nodes
- Defining a global (within the PCN domain) configuration parameter U at the PCN-ingress_nodes. This parameter signifies the implicit ratio between the termination and admission thresholds at all links
- Multiplication of Sustainable-Aggregate-Rate by the constant U at the PCN-ingress-nodes if MARKING_MODE=SINGLE_MARKING
- Using the MARKING_MODE parameter to guide which marking is used to measure the CLE (but the measurement functionality is unchanged)

4.4. Proposed Optional Renaming of the Marking and Marking Thresholds

Previous work on example mechanisms [I-D.briscoe-tswg-cl-architecture] implementing the architecture of [I-D.eardley-pcn-architecture] assumed that the semantics of admission control marking and termination marking differ. Specifically, it was assumed that for termination purposes the semantics of the marking is related to the excess rate over the configured (termination) rate, or even more precisely, the amount of traffic that remains unmarked (sustainable rate) after the excess traffic is marked. Some of the recent proposals assume yet different marking semantics [I-D.babiarz-pcn-3sm], [I-D.westberg-pcn-load-control].

Even though specific association with marking semantics and function (admission vs termination) has been assumed in prior work, it is important to note that in the current architecture draft
[I-D.eardley-pcn-architecture], the associations of specific marking semantics (virtual queue vs excess rate) with specific functions (admission vs termination) are actually *not* directly assumed. In fact, the architecture document does not explicitly define the marking mechanism, but rather states the existence of two different marking mechanisms, and also allows implementation of either one or both of these mechanisms in a PCN-domain.

We argue that this separation of the marking semantics from the functional use of the marking is important to make sure that devices supporting the same marking can interoperate in delivering the function which is based on specific supported marking semantics.

To divorce the function (admission vs termination) and the semantics (excess rate marking, virtual queue marking), it may be beneficial to rename the marking to be associated with the semantics rather than the function to explicitly disassociate the two functions. Specifically, it may be beneficial to change the "admission-marking" and "termination-marking" currently defined in the architecture as "Type Q" or "virtual-queue-based" marking, and "Type R" or "excess-rate-based" marking. Of course, other choices of the naming are possible (including keeping the ones currently used in [I-D.eardley-pcn-architecture]).

With this renaming, the dual marking approach in [I-D.briscoe-tsvwg-cl-architecture] would require PCN-internal-nodes to support both Type R and Type Q marking, while Single Marking would require support of Type-R marking only.

We conclude by emphasizing that the changes proposed here amount to merely a renaming rather than a change to the proposed architecture, and are therefore entirely optional.


We note finally that it is possible to use a single configuration constant \( U \) instead of two constants (\( U \) and MARKING_TYPE). Specifically, one can simply interpret the value of \( U=1 \) as the dual-marking approach (equivalent to MARKING_TYPE=DUAL_MARKING) and use \( U>1 \) to indicate Single Marking.

5. Incremental Deployment Considerations

As most of today’s routers already implement a token bucket, implementing token-bucket based excess-rate marking at PCN-ingress nodes is a relatively small incremental step for most of today’s
implementations. Implementing an additional metering and marking scheme in the datapath required by the dual-marking approach without encountering performance degradation is a larger step. The single-marking approach may be used as an intermediate step towards the deployment of a dual-marking approach in the sense that routers implementing single-marking functionality only may be incrementally deployed.

The deployment steps might be as follows:

- Initially all PCN-ingress-nodes might implement Excess-rate (Type R) type marking and metering only.

- All PCN-boundary nodes implement the full functionality as described in this document (including the configuration parameters MARKING_TYPE and U) from the start. Since the PCN-boundary-node behavior is enabled by simply changing the values of the configuration parameters, all boundary nodes become immediately compatible with both dual-marking and single-marking.

- Initially all boundary nodes are configured parameter settings indicating Single Marking option.

- When a PCN-internal node with dual-marking functionality replaces a subset of PCN-internal-nodes, the virtual-queue-based (Type Q) marking is simply ignored by the boundary nodes until all PCN-internal-nodes in the PCN-domain implement the dual-marking metering and marking. At that time the value of the configuration parameters may be reset to at all boundary nodes to indicate the Dual Marking configuration.

- Note that if a subset of PCN-boundary-nodes communicates only with each other, and all PCN-internal-nodes their traffic traverses have been upgraded, this subset of nodes can be upgraded to two dual-marking behavior while the rest of the PCN-domain can still run the single marking case. This would entail configuring two thresholds at the PCN-internal-nodes, and setting the value of the configuration parameters appropriately in this subset.

- Finally note that if the configuration parameter U is configured per ingress-egress-pair rather than per boundary node, then each ingress-egress pair can be upgraded to the dual marking simultaneously. While we do not recommend that U is defined on a per-ingress-egress pair, such possibility should be noted and considered.

This draft presents the results of a follow-up simulation study. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].
2.3.3. Real Video Traces (VTR) ......................... 10
2.3.4. Randomization of Base Traffic Models .............. 10
2.4. Performance Metrics .................................. 11
2.5. Simulation Environment ............................... 11
3. Admission Control ...................................... 11
3.1. Parameter Settings ................................... 11
3.1.1. Virtual queue settings ............................. 11
3.1.2. Egress measurements ............................... 12
3.2. What Bottleneck Aggregation is Sufficient? .......... 12
3.3. Sensitivity to Call Arrival Assumptions ............. 14
3.4. Sensitivity to Marking Parameters at the Bottleneck .. 16
3.4.1. Ramp vs Step Marking ............................... 16
3.4.2. Sensitivity to Virtual Queue Marking Thresholds .. 16
3.5. Effect of Ingress-Egress Aggregation ................. 20
3.6. Effect of Multiple Bottlenecks ....................... 21
3.7. Utilization of overloaded bottlenecks ............... 21
3.8. Fairness Between Long-haul and Short-haul flows ... 22
4. Termination Control .................................... 25
4.1. Termination Model and Key Parameters ............... 25
4.2. Effect of RTT Difference ............................. 26
4.3. Ingress-Egress Aggregation Experiments ............. 29
4.3.1. Motivation for the Investigation ................. 29
4.3.2. Detailed results .................................. 30
4.4. Multiple Bottlenecks Experiments ................. 35
4.4.1. Motivation for the Investigation ................. 35
4.4.2. Detailed Results .................................. 36
4.5. Sensitivity to Call Arrival Assumptions ........... 41
5. Summary of Results ..................................... 41
5.1. Summary of Admission Control Results ............... 41
5.2. Summary and Discussion of Termination Results ..... 42
6. Future work ............................................. 43
IANA Considerations ....................................... 44
Security Considerations ................................... 44
References ............................................... 44
9.1. Normative References ................................. 44
9.2. Informative References ............................... 44
Authors’ Addresses ...................................... 45
Intellectual Property and Copyright Statements .......... 46

Introduction Pre-Congestion Notification approach (draft-eardley-pcn-architecture, [I-D.briscoe-tsvwg-cl-architecture]) proposes Admission Control to limit the amount of real-time PCN traffic to a configured level during the normal operating conditions, and Flow Termination used to tear down some of the flows to bring the PCN traffic level down to a desirable...
amount during unexpected events such as network failures, with the goal of maintaining the QoS assurances to the remaining flows. In draft-eardley-pcn-architecture, Admission and Termination use two different markings and two different metering mechanisms in the internal nodes of the PCN region. Here and elsewhere in this document we will omit "Flow" and refer to Flow Termination simply as Termination. An initial simulation study was reported in [I-D.briscoe-tsvwg-cl-phb], where it was shown that both Admission and Termination mechanisms discussed there have reasonable performance in a limited set of experiments performed there. This draft reports the next installment of the simulation results. For completeness and convenience of exposition, most of the results earlier presented in [I-D.briscoe-tsvwg-cl-phb] have been moved into this draft. The new results presented in the current draft further confirm that Admission and Termination algorithms of [I-D.briscoe-tsvwg-cl-phb] perform well under a range of operating conditions and are relatively insensitive to parameter variations around a chosen operation range. Perhaps the most interesting (and somewhat unexpected) conclusion that can be drawn from these results is that both Admission and Termination algorithms appear to be not as sensitive to low per ingress-egress-pair aggregation as one might fear. This result is quite encouraging: while it seems reasonable to assume sufficient bottleneck link aggregation, it is not very clear whether one can safely assume high levels of aggregation on a per ingress-egress-pair basis. Yet, low levels of ingress-egress aggregation remain a potential concern, especially for the Termination mechanism. More discussion on this is presented in section 4. Other conclusions are presented in Section 5.) Section 2 describes simulation environment and models, Admission and termination simulation results are presented in sections 3 and 4, and section 5 summarizes the results of the simulations so far and lists areas for further study. Zhang, et al. Expires January 6, 2008

1.1. Changes from the previous version
- Refined the analysis of low aggregation effect on Termination
- Added batch arrivals experiments for Termination
- Added Fairness analysis for Admission
- Added experiments with different voice codecs mixes sharing the bottleneck
- Replaced the Terminology section with a pointer to draft-eardley-pcn-architecture
- Miscellaneous editorial changes and clarifications based on feedback to the previous version.

1.2. Terminology
This draft uses the terminology as defined in draft-eardley-pcn-architecture-00.

2. Simulation Setup and Environment
2.1. Network Models
We use three types of topologies, described in this section. In the simplest topology shown in Fig. 2.1 the network is modelled as a single link between an ingress and an egress node, all flows sharing the same link. Figure 2.1 shows the modelled network. A is the ingress node and B is the egress node. A-----B Fig. 2.1 Simulated Single Link Network (Referred to as
Single Link Topology) A subset of simulations uses a network structured similarly to the network shown on Figure 2.2. A set of ingresses (A, B, C) connected to an interior node in the network (D) with links of different propagation delay. This node in turn is connected to the egress (F). In this topology, different sets of flows between each ingress and the egress converge on the single link, where Pre-congestion notification algorithm is enabled. The ingress link capacity is assumed to be sufficiently large so that neither Admission nor Zhang, et al. Expires January 6, 2008 [Page 5]
Termination mechanisms have any effect on them. All links are assigned a propagation delay. The point of congestion (link (D-F) connecting the interior node to the egress node) is modeled with a 1ms or 10ms propagation delay. In our simulations, the number of ingress nodes in the network range from 2 to 1800 nodes, each connected to the interior node with a range of propagation delay (1ms to 100ms). In some experiments all ingress links have the same propagation delay, and in some experiments the delay of different ingresses vary in the range from 1 to 100 ms.

A B D F / C

Fig. 2.2. Simulated Multi-Link Network (Referred to as RTT Topology) Another type of network of interest is multi-bottleneck topology that we call Parking Lot (PLT). The simplest PLT with 2 bottlenecks is illustrated in Fig. 2.3(a). An example traffic matrix with this network on this topology is as follows:

- an aggregate of "2-hop" flows entering the network at A and leaving at C (via the two links A-B-C)
- an aggregate of "1-hop" flows entering the network at D and leaving at E (via A-B)
- an aggregate of "1-hop" flows entering the network at E and leaving at F (via B-C)

In the 2-hop PLT of Fig. 2.3(a) the points of congestion are links A--B and B--C. Capacity of all other links is not limiting. This topology and traffic matrix models the network where some flows cross multiple bottlenecks, each with substantial amount of cross-traffic. A--B--C A--B--C--D A--B--C--D--E--F E F G H I J K L

(b) (c) Figure 2.3: Simulated Multiple-bottleneck (Parking Lot) Topologies. We also experiment with larger PLT topologies with 3 bottlenecks (see Fig 2.3(b)) and 5 bottlenecks (Fig 2.3 (c)). In all cases, we...
Internet-Draft CL Simulation Study July 2007 simulated one ingress-egress pair that carries the aggregate of "long" flows traversing all the N bottlenecks (where N is the number of bottleneck links in the PLT topology, shown as "horizontal" links in Fig. 2.3), and N ingress-egress pairs that carry flows traversing a single bottleneck link and exiting at the next "hop". In all cases, capacities of all "vertical" links are non-limiting, so neither Termination nor Admission mechanisms are never triggered on these links. Propagation delays for all links in all PLT topologies are set to 1ms. These topologies aim to model the cross traffic and congestion that can occur in the hierarchically structured networks deployed by many
Due to time limitations, other possible traffic matrices (e.g. some of the flows traversing a subset of several bottleneck links in Fig 2.3) have not yet been considered and remain the area for future investigation. Our simulations concentrated primarily on the range of capacities of ‘bottleneck’ links with sufficient level of bottleneck aggregation - above 10 Mbps for voice and 622 Mbps for "video", up to 2.4 Gbps. But we also investigated slower ‘bottleneck’ links down to 512 Kbps in some experiments. 2.2. Call Signaling Model In the simulation model of Flow Admission Control, a flow request arrives at the ingress and immediately sends a message to the egress. The message arrives at the egress after the propagation time plus link processing time (but no queuing delay). When the egress receives this message, it immediately responds to the ingress with the current Congestion Level Estimate (CLE). If the CLE is below the specified CLE- threshold, the flow is admitted, otherwise it is rejected. For Termination, once the ingress node of a PCN region decides to terminate a flow, that flow is terminated immediately and sends no more packets from that time on. The life of a flow outside the domain described above is not modelled. 2.3. Traffic Models We simulated four models of real-time traffic - two voice models and two video models. The voice models included CBR voice and on-off traffic approximating voice with silence compression. For video, we Zhang, et al. simulated on-off traffic with peak and mean rates corresponding to an MPEG-2 video stream (we termed the latter Synthetic Video (SVD)), and a real video trace (VTR). The distribution of flow duration was chosen to be exponentially distributed with mean 1min, regardless of the traffic type. In most of the experiments flows arrived according to a Poisson distribution with mean arrival rate chosen to achieve a desired amount of overload over the configured-admission-rate in each experiment. Overloads in the range 1x to 5x and underload with 0.95x have been investigated. For on-off traffic, on and off periods were exponentially distributed with the specified mean. Traffic parameters for each flow are summarized below. 2.3.1. Voice Traffic Models The table below describes all voice codecs we modeled in our simulation results. The first two rows correspond to our two basic models (they correspond to the older G.711 encoding with and without silence compression). These two models are referred simply as "CBR" and "VBR" in the reported simulation results. We also simulated several "mixes" of the different codecs reported in the table below. The primary mix consists of equal proportion of all voice codecs list below. We have also simulated various other mix consist different proportion of the subset of all codecs. Though these result are not reported in this draft due to their similarities to the primary mix result. Zhang, et al. Expired January 6, 2008 [Page 7] Internet-Draft CL Simulation Study July 2007
<table>
<thead>
<tr>
<th>Name/Codecs</th>
<th>Packet Size</th>
<th>Inter-Arrival</th>
<th>On/Off Period</th>
<th>Average Rate</th>
<th>Time (ms)</th>
<th>Ratio</th>
<th>(kbps)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;CBR&quot;</td>
<td>160</td>
<td>20</td>
<td>1</td>
<td>64</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;VBR&quot;</td>
<td>160</td>
<td>20</td>
<td>0.34</td>
<td>21.75</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G.711 CBR</td>
<td>200</td>
<td>20</td>
<td>1</td>
<td>80</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G.711 VBR</td>
<td>200</td>
<td>20</td>
<td>0.4</td>
<td>32</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G.711 CBR</td>
<td>120</td>
<td>10</td>
<td>1</td>
<td>96</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G.711 VBR</td>
<td>120</td>
<td>10</td>
<td>0.4</td>
<td>38.4</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G.729 CBR</td>
<td>60</td>
<td>20</td>
<td>1</td>
<td>24</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G.729 VBR</td>
<td>60</td>
<td>20</td>
<td>0.4</td>
<td>9.6</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1. Simulated Voice Codecs.

2.3.2. Synthetic "Video" - High Peak-to-Mean Ratio VBR Traffic (SVD)

This model is on-off traffic with video-like mean-to-peak ratio and mean rate approximating that of an MPEG-2 video stream. No attempt is made to simulate any other aspects of a video stream, and this model is merely that of on-off traffic. Although there is no claim that this model represents the performance of video traffic under the algorithms in question adequately, intuitively, this model should be more challenging for a measurement-based algorithm than the actual MPEG video, and as a result, 'good' or "reasonable" performance on this traffic model indicates that MPEG traffic should perform at least as well. We term this type of traffic SVD for "Synthetic Video". Parameters used for this traffic models are: o Long term average rate 4 Mbps o On Period mean duration 340ms; during the on-period the packets are sent at 12 Mbps o 1500 byte packets, packet inter-arrival: 1ms o Off Period mean duration 660ms

from random locations in the trace with duration chosen to be exponentially distributed with mean 1min. The results show that the expected rate of flow is roughly the same as the trace's average. Traffic characteristics are summarized below: o Average rate 769 Kbps o Each frame is sent with packet length 1500 bytes and packet inter-arrival time 1ms o No traffic is sent between frames. 2.3.4. Randomization of Base Traffic Models To emulate some degree of network-introduced jitter, in some experiments we implemented limited randomization of the base models by randomly moving the packet by a small amount of time around its transmission time in the corresponding base traffic model. More specifically, for each packet we chose a random number R, which is picked from uniform distribution in a randomization-interval, and delayed the packet by R compared to its ideal departure time. We choose randomization-interval to be a fraction of packet-inter-arrive-time of the CBR portion of the corresponding base model. To simulate a range of queueing delays, we varied this fraction from 0.0001 to 0.1. While we do not claim this to be an adequate model for network-introduced jitter, we chose it for the simplicity of implementation as a means to gain insight on any simulation artifacts of strictly CBR traffic generation. We implemented randomized versions of all 5 traffic streams (CBR, VBR, MIX, SVD and VTR) by randomizing the CBR portion of each model.

Zhang, et al. Expires January 6, 2008 [Page 10] Internet-Draft CL Simulation Study July 2007 2.4. Performance Metrics In all our experiments we use as performance metric the percent deviation of the mean rate achieved in the experiment from the expected load level. We term these "over-admission" and "over-termination" percentages, depending on the type of the experiment. More specifically, our experiments measure the actual achieved throughput at 50 ms intervals, and then compute the average of these 50ms rate samples over the duration of the experiment (where relevant, excluding warmup/startup conditions). We then compare this experiment average to the desired traffic load. Initially in our experiments we also computed the variance of the traffic around the mean, and found that in the vast majority of the experiments it was quite small. Therefore, in this draft we omit the variance and limit the reporting to the over-admission and over-termination percentages only. 2.5. Simulation Environment The simulation study reported here used purpose built discrete-event simulator implemented in ECLiPSe Language (http://eclipse.crosscoreop.com/eclipse). The latter is intended for general programming tasks, and is especially suitable for rapid prototyping. Simulations were run on Enterprise Linux Red Hat, IBM eServer x335, 3.2GHz Intel Xeon, 4GB RAM. 3. Admission Control 3.1. Parameter Settings 3.1.1. Virtual queue settings Unless otherwise specified, most of the simulations were run with the following Virtual Queue thresholds: o min-marking-threshold: 5ms at virtual queue rate o max-marking-threshold: 15ms at virtual queue rate o virtual-queue-upper-limit: 20ms at virtual queue rate

Most of the simulations were set with the configured-admissible-rate at half the link speed. Note that as long as there is no packet loss, the admission control scheme successfully keeps the load of admitted flows at the desired level regardless of the actual setting of the configured-admissible-rate. However, it is not clear if this remains true when the configured-admissible-rate is close to the link speed/actual queue service rate. Further work is necessary to quantify the performance of the scheme with smaller service rate/virtual queue rate ratio, where packet loss may be an issue.

Egress measurements The CLE is computed as an exponential weighted moving average (EWMA) with a weight of 0.01. In the simulation results presented in sections 3.2 and 3.3 the CLE is computed on a per-packet basis as it is that setting that was used in [I-D.briscoe- tsvwg-cl-phb], from which these results are taken. For those experiments the CLE value 0.5 and EWMA weight of 0.01 are used unless otherwise specified. Our subsequent study indicated that there is no significant difference between the observed performance of interval-based and per-packet egress measurements. Since interval based measurements for a large number of ingresses are substantially easier for hardware implementations, subsequent studies reported in the rest of this draft concentrated on the interval based egress measurement. The measurement interval was chosen to be 100ms, and a range of CLE values and EWMA weights was explored, as specified in specific experiment descriptions.

3.2. What Bottleneck Aggregation is Sufficient? One of the assumptions in [I-D.briscoe-tsvwg-cl-architecture] is that there is sufficient aggregation on the "bottleneck" links. Our first set of experiments revolved around getting some preliminary intuition of what constitutes "enough bottleneck aggregation" for the traffic models we chose. To that end we fixed configured-admissible-rate at half the link speed in the range of T1 (1.5 Mbps) through 1Gbps, and examined the level of aggregation at different link speeds for different traffic models corresponding to the chosen configured admission rate at those speeds. Further, to eliminate the issue of whether ingress-egress pair aggregation has any significant effect, in the experiments performed in this section we used Single Link topology only, so that all flows shared the same ingress-egress pair. We found that on links of capacity from 10Mbps to OC3, admission control for CBR voice and ON-OFF voice (VBR) traffic work reliably with the range of parameters we simulated, both with Poisson and Batch call arrivals. As the performance of the algorithm was quite good at these speeds, and generally becomes the better the higher the Zhang, et al. Expires January 6, 2008 [Page 12] Internet-Draft CL Simulation Study July 2007 degree of aggregation of traffic, we chose to not...
investigate higher link speeds for CBR and VBR voice, within the time
constraints of this effort. The performance at lower link speeds was
substantially worse, and these results are not presented here. These
results indicate that a rule of thumb, admission control algorithm
described in [I-D.briscoe-tsvwg-cl-architecture] should not be used
at aggregations substantially below 5 Mbps of aggregate rate even for
voice traffic (with or without silence compression). For higher-rate
on-off SVD traffic, due to time limitations we simulated 1Gbps and
OC12 (622 Mbps) links and Poisson arrivals only. Note that due to
the high mean and peak rates of this traffic model, slower links are
unlikely to yield sufficient level of aggregation of this type of
traffic to satisfy the flow aggregation assumptions of [I-D.briscoe-
tsvwg-cl-architecture]. Our simulations indicated that this model
also behaved quite well at these levels of aggregation, although the
deviation from the configured-admissible-rate is slightly higher in
this case than for the less bursty traffic models. Recalling that
simulated SVD model is in fact just on-off traffic with high peak
rate and video-like peak ratio, we believe that the actual video will
behave only better, and hence it follows that with bottleneck
aggregation of the order of 150 SVD flows the admission control
algorithm is expected to perform reasonably well. Note however that
this statement assumes sufficient per ingress-egress pair aggregation
as well. Due to time limitations bottleneck aggregation experiments
were not performed for other traffic models. For the chosen link
speeds and traffic models, we investigated the demand overload of
2x-5x. By demand overload we mean that the sources generate traffic
with the aggregate mean rate exceeding the configured-admissible-rate
by the specified factor. Performance at lower levels of demand
overload is expected to be only better. Higher levels of overloads
have not been studied due to time limitations, especially given the
expectation that the 5x demand overload is already sufficiently rare
to expect in practice. Table 3.1 below summarizes the worst case
difference (in percent) between the admitted load and configured-
admissible-rate (we refer to as over-admission-perc). The worst case
difference was taken over all experiments with the corresponding
range of link speeds and demand overloads. In general, the higher
the demand, the more challenging it is for the admission control
algorithm due to a larger number of near-simultaneous arrivals at
higher overloads, and as a result the worst case results in Table 3.1
correspond to the 5x demand overload experiments. Zhang, et al.
Expires January 6, 2008 [Page 13] Internet-Draft CL Simulation Study
July 2007

<table>
<thead>
<tr>
<th>Link type</th>
<th>traffic</th>
<th>call</th>
<th>over-admission</th>
<th>standard</th>
<th>Link type</th>
<th>traffic</th>
<th>call</th>
<th>over-admission</th>
<th>standard</th>
<th>Link type</th>
<th>traffic</th>
<th>call</th>
<th>over-admission</th>
<th>standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>T3,100Mbps,OC3</td>
<td>CBR</td>
<td>POISSON</td>
<td>0.5%</td>
<td>0.005</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 3.1. Summary of the admission control results for links above T3 speeds. Note: T3 = 45Mbps, OC3 = 155Mbps, OC12 = 622Mbps.

Results correspond to 5x overload on a Single Link Topology. 3.3. Sensitivity to Call Arrival Assumptions In the previous section we reported that at sufficient levels of aggregation Poisson call arrivals assumption was not critical in the sense that even a burstier, batch arrival process resulted in a reasonable performance for all traffic models. In this section we investigate to what extent the Poisson call arrival assumption affect the accuracy of the admission control algorithm at lower levels of bottleneck aggregation. To that end we first investigated the comparative performance of the algorithm with Poisson and Batch call arrival processes for the CBR and VBR voice traffic. The mean call arrival rate was the same for both processes, with the demand overloads ranging from 2x to 5x. Table 3.2 below summarizes the difference between the admitted load and the configured-admissible-rate for CBR Voice in the case of Poisson and Batch arrivals. Table 3.3 provides a similar summary for on-off traffic simulating voice with silence compression. The results in the tables correspond to the worst case across all overload factors (and when multiple links speeds are listed, across all those link speeds).  

<table>
<thead>
<tr>
<th>Link type</th>
<th>arrival</th>
<th>over-admission</th>
<th>standard deviation to conf-adm-rate</th>
<th>ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1Mbps, T1</td>
<td>BATCH</td>
<td>30.0%</td>
<td>0.30</td>
<td></td>
</tr>
<tr>
<td>10 Mbps, T1</td>
<td>BATCH</td>
<td>5.0%</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td>1Mbps, T3,100Mbps,OC3</td>
<td>BATCH</td>
<td>1.0%</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>10 Mbps, T1</td>
<td>POISSON</td>
<td>5.0%</td>
<td>0.10</td>
<td></td>
</tr>
<tr>
<td>10 Mbps, T3,100Mbps,OC3</td>
<td>POISSON</td>
<td>0.5%</td>
<td>0.005</td>
<td></td>
</tr>
</tbody>
</table>
3.2. Comparison of Poisson and Batch call arrival models for CBR voice. Note: T1 = 1.5Mbps, T3 = 45Mbps, OC3 = 155Mbps, OC12 = 622Mbps. The results are for 5x overload on a Single Link Topology.

<table>
<thead>
<tr>
<th>Link type</th>
<th>arrival</th>
<th>over-admission</th>
<th>standard deviation to conf-adm-rate</th>
<th>model</th>
<th>percent</th>
<th>ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1Mbps, T1</td>
<td>BATCH</td>
<td>40.0%</td>
<td>0.30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Mbps</td>
<td>BATCH</td>
<td>8.0%</td>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3, 100Mbps, OC3</td>
<td>BATCH</td>
<td>3.0%</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1Mbps, T1</td>
<td>POISSON</td>
<td>15.0%</td>
<td>0.20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Mbps</td>
<td>POISSON</td>
<td>7.0%</td>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3, 100Mbps, OC3</td>
<td>POISSON</td>
<td>2.5%</td>
<td>0.025</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.3. Comparison of Poisson and Batch call arrival models for VBR voice with silence compression. Note: T1 = 1.5Mbps, T3 = 45Mbps, OC3 = 155Mbps, OC12 = 622Mbps. As can be seen, there is substantial sensitivity to Poisson call arrivals at lower bottleneck aggregation levels, but very little performance difference is observed as long as the aggregation levels are sufficiently high. Zhang, et al. Expires January 6, 2008 [Page 15] Internet-Draft CL Simulation Study July 2007 Subsequently we also investigated sensitivity to Poisson assumption with all other traffic models and other topologies. Due to time limitations, we investigated this only at higher levels of aggregation. Specifically, all voice experiments, including various codecs mixes are run on bottleneck link with OC3 (155 Mbps) bottleneck links, VTR traces are run on 1Gbps and SVD on OC48 (2.4Gbps) links. At these levels of aggregation we have run the experiments on the entire set of topologies and parameter settings reported in this draft, and fond that the performance with BATCH arrivals is very close to that of Poisson arrivals across the entire range of these experiments. This confirms that BATCH arrivals have little effect on the performance compared to Poisson at sufficient aggregation levels and demand overloads in the studied range. 3.4. Sensitivity to Marking Parameters at the Bottleneck 3.4.1. Ramp vs Step Marking Draft [I-D.briscoe-tsvwg-cl-architecture] gave an option of "ramp" and "step" marking at the bottleneck. The behavior of the congestion control algorithm in all simulation experiments we performed did not substantially differ depending on whether the marking was "ramp", i.e. whether a separate min-marking-threshold and max-marking- threshold were used, with linear marking probability between these thresholds, or whether the marking was "step" with the
min-marking-threshold and max-marking-threshold collapsed at the max-marking-threshold value, and marking all packets with probability 1 above this collapsed threshold. However, the difference between "ramp" and "step" may be more visible in the multiple congestion point case (evaluation of "ramp" vs "step" performance in the multi-bottleneck case remains an area for future work). Another possible reason for this apparent lack of difference between "ramp" and "step" may relate to the choice of CLE threshold and measurement timescale. Choosing a lower CLE threshold and a faster measurement timescale may result in a better sensitivity to lower levels of marked traffic. Investigating the interaction between settings of the marking thresholds, the CLE-threshold, and the measurement parameters at the egress remains an area of future investigation.

3.4.2. Sensitivity to Virtual Queue Marking Thresholds

The limited number of simulation experiments we performed indicate that the choice of the absolute value of the min-marking-threshold, the max-marking-threshold and the virtual-queue-upper-limit can have Zhang, et al. Expires January 6, 2008 [Page 16]
a visible effect on the algorithm performance. Specifically, choosing the min-marking-threshold and the max-marking-threshold too small may cause substantial under-utilization, especially on the slow links. However, at larger values of the min-marking-threshold and the max-marking-threshold, preliminary experiments suggest the algorithm’s performance is insensitive to their values. The choice of the virtual-queue-upper-limit affects the amount of over-admission (above the configured-admissible-rate threshold) in some cases, although this effect is not consistent throughout the experiments. The Table 3.4 below gives a summary of the difference between the admitted load and the configured-admissible-rate as a function of the virtual queue parameters, for the SVD traffic model. The results in the table represent the worst case result among the experiments with different degree of demand overloads in the range of 2x-5x. Typically, higher deviation of admitted load from the configured-admissible-rate occurs for the higher degree of demand overload. The sensitivity of smoother CBR and VBR voice traffic models to the variation of these parameters is not as significant as that presented in Table 3.4 for SVD.

<table>
<thead>
<tr>
<th>standard</th>
<th>Link type</th>
<th>min-threshold, max-threshold,</th>
<th>over-admission deviation to</th>
<th>percent</th>
<th>conf-adm-rate</th>
<th>upper-limit(ms)</th>
<th>ratio</th>
</tr>
</thead>
</table>
| 1Gbps     | 5, 15, 20 | 6.0% | 0.08 | 0.08  | 1Gbps
| 1Gbps     | 1, 5, 10 | 2.0% | 0.07 | 0.07  | 1Gbps
| 1Gbps     | 5, 15, 45 | 2.0% | 0.08 | 0.08  | OC12
|5, 15, 20 | 5.0% | 0.11 |
----------------------------------------------- | OC12
|1, 5, 10 | 2.0% | 0.13 |
----------------------------------------------- | OC12
|5, 15, 45 | 0.0% | 0.10 |
-----------------------------------------------

Table 3.4. Sensitivity of 4 Mbps on-off SVD traffic to the virtual queue settings. Note: T1 = 1.5Mbps, T3 = 45Mbps, OC3 = 155Mbps, OC12 = 622Mbps

3.5. Sensitivity to RTT We performed a limited amount of sensitivity analysis of the admission control algorithm used to the range of round trip propagation time (which is the dominant component of the control delay in the typical environment using Pre-congestion notification). Zhang, et al. Expires January 6, 2008 [Page 17]
We considered both the case when all flows in a given experiment had the same RTT from this range, and also when RTT of different flows sharing a single bottleneck link in a single experiment had a range of round trip delays between 22 and 220 ms. The results were good for all types of traffic tested, implying that the admission control algorithm is not sensitive to the either the absolute value of the round-trip propagation time or relative value of the round-trip propagation time, at least in the range of values tested. In addition, we found no sign of unfairness to the flows with large RTT. We expect this to remain true for a wider range of round-trip propagation times. It is important to note that these results relate to the difference in RTT of flows sharing a single bottleneck. One can expect that flows with longer RTT also traverse more bottleneck links. This effect of multiple bottlenecks is studied separately and is reported later in this draft.

3.6. Sensitivity to EWMA weight and CLE

This section represents the results of the investigation the combined effect of the EWMA weight and CLE setting at the egress in three types of settings on: o a Single Link topology of Fig. 2.1 o RTT topology of Fig. 2.2 with 100 ingress links o PLT topologies of Fig. 2.3 We experiment with 3 levels of CLE (0.05, 0.15, 0.25) in combination of EWMA weight ranging from 0.1 to 0.9 (in 0.2 step increase). The demand overload is taken to be 5x. For brevity, instead of listing all 15 values (for each combination of weight and CLE), we present the 4-tuple summaries across all experiments. For PLT topology with N bottlenecks, we have N over-admission-perc. values (each corresponds to one bottleneck link). We show here only the worse case values. That is, in the overload experiments (1-5x), the maximum of the N over-admission-perc is displayed. The results below are presented for non-randomized traffic models. Randomized versions of all traffic type were tested as well, but no meaningful difference were observed. The simulation results reveal that for all of the traffic models tested except SVD, the admission control is rather insensitive to the EWMA weight and CLE changes. These statistics show that over-
Internet-Draft CL Simulation Study July 2007

admission percentage values are rather similar, with the admitted load staying within -3%\/+2% range of the desired admission threshold, with quite limited variability.

<table>
<thead>
<tr>
<th>Type</th>
<th>Topo</th>
<th>Over Admission Perc Stats</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>S.Link</td>
<td></td>
<td></td>
<td>0.224</td>
<td>1.105</td>
<td>0.801</td>
<td>0.179</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.198</td>
<td></td>
<td></td>
<td>-0.93</td>
</tr>
<tr>
<td></td>
<td>CBR</td>
<td>RTT</td>
<td>0.200</td>
<td>1.192</td>
<td>0.851</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.528</td>
<td>0.559</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>S.Link</td>
<td></td>
<td>0.07</td>
<td>1.646</td>
<td>1.272</td>
<td>0.396</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.434</td>
<td></td>
<td></td>
<td>-1.48</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.803</td>
<td>1.329</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.798</td>
<td>0.958</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VBR</td>
<td>RTT</td>
<td>-0.11</td>
<td>1.830</td>
<td>1.329</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-0.93</td>
<td>0.990</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.528</td>
<td>0.559</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MIX</td>
<td>RTT</td>
<td>-0.46</td>
<td>1.803</td>
<td>1.171</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.363</td>
<td>0.798</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VTR</td>
<td>RTT</td>
<td>-0.57</td>
<td>1.313</td>
<td>0.855</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.585</td>
<td>0.606</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.363</td>
<td>0.798</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVD</td>
<td>RTT</td>
<td>-2.98</td>
<td>5.357</td>
<td>2.541</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.618</td>
<td>6.525</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.294</td>
<td>4.294</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.229</td>
<td>1.229</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.903</td>
<td>2.903</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Summarized performance for CBR, VBR, MIX, VTR, SVD across different parameter settings and topologies. For SVD, the algorithms do show certain sensitivity to parameters, which means that high peak-to-mean ratio SVD traffic is more stressful to the queue-based admission control algorithm, but a set of parameters exists that keeps the over-admission with about -3\/+7% of the expected load. Note that since the configured-admissible-rate is expected to be set substantially below the actual link capacity, and PCN traffic is typically expected to be served at high priority over non-PCN traffic, 10% overload does not result in any loss as long as the configured-admissible-rate is set below 90% of the link speed. Hence, we treat 10% overload as "reasonable" for practical purposes. A negative overload indicates that less traffic is admitted than the policy threshold would allow, indicating potential underutilization.


We can assess the effect of Ingress-Egress aggregation on the algorithm by comparing the SingleLink results in Table 3.5 with the corresponding RTT results. As discussed earlier, the actual choice of RTT values of different ingress links does not appear to have any significant effect on the simulation results. We believe that any appreciable difference between the two topologies relates to the degree of aggregation of each ingress-egress pair. One of the outcomes of the results presented in Table 3.5 is that the admission control algorithm of [I-D.briscoe-tsvwg-cl-architecture] seems relatively insensitive to the level of ingress-egress aggregation.
<table>
<thead>
<tr>
<th>Number of Ingresses and the over-admission perc.</th>
<th>Type</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CBR</td>
<td>1.003</td>
</tr>
<tr>
<td></td>
<td>-1.45</td>
<td>0.396</td>
</tr>
<tr>
<td></td>
<td>VBR</td>
<td>1.021</td>
</tr>
<tr>
<td></td>
<td>0.721</td>
<td>-0.85</td>
</tr>
<tr>
<td></td>
<td>MIX</td>
<td>1.080</td>
</tr>
<tr>
<td></td>
<td>1.132</td>
<td>1.098</td>
</tr>
<tr>
<td></td>
<td>VTR</td>
<td>1.109</td>
</tr>
<tr>
<td></td>
<td>0.856</td>
<td>0.862</td>
</tr>
<tr>
<td></td>
<td>SVD</td>
<td>-0.08</td>
</tr>
<tr>
<td></td>
<td>-1.56</td>
<td>0.914</td>
</tr>
</tbody>
</table>

3.6 Ingress aggregation effect. Each cell in the table shows the number of PCN-ingress-nodes generating the flows sharing the bottleneck (top number) and the corresponding over-admission percentage (bottom number). The results correspond to EWMA weight of 0.3, CLE=0.05, demand overload 5x) Table 3.6 summarizes the effect of ingress-egress aggregation. For each traffic type, the mean number of flows sharing the bottleneck is constant in all experiments. The number of ingresses therefore is inversely proportional to the level of ingress-egress aggregation. As can be seen, the right-most column represents the lowest aggregation level (expected 1 call/ingress), indicating that algorithm is rather insensitive toward the level of ingress-egress aggregation. These results are very encouraging: while the assumption of Zhang, et al. becomes more questionable as the aggregation of PCN traffic at an internal bottleneck seems a relatively safe one, it is much less clear that it is safe to assume that high per ingress-egress aggregation level is a safe assumption in reality. In particular, the SVD setup with only ~100 SVD flows taking up about 50% of a 1G bottleneck link bandwidth with all 100 flows coming from different ingresses seems entirely plausible. It is therefore encouraging that the algorithm seems sufficiently robust under these circumstances.

3.8. Effect of Multiple Bottlenecks In this section we report a set of experiments on the multi-bottleneck topology. 3.8.1. Utilization of overloaded bottlenecks Our first set of experiments (reported in Table 3.5) investigates whether multiple bottlenecks have any effect on the utilization of bottlenecks links all of which contain a mix of flows traversing multiple bottlenecks and small number of bottlenecks (in our case just one bottleneck). We term the former "long-haul" flows, and the latter "short-haul" flows. In these experiments, we use the PLT topology where the long-
haul flows traverse the entire length of the chain, and short-term flows traverse only one hop. The demands of all short- and long-haul flows are the same, and the demand overloads on each bottleneck link in the topology are also the same. We experiment with all sizes of PLT topologies from 2 to 5 and all demand overloads up to 5x, and a range of different parameter (weight and CLE) settings. For each one of them we report the utilization of all the bottleneck links. In Table 3.7, we show a snapshot of the behavior of all bottlenecks in a 5 bottleneck topology. Here, the over-admission-perc. displayed for each link is an average across all 15 experiments with different [weight, CLE] setting for a 5x overload. (We do observe very much the same behavior in each of the individual experiment, hence providing summarized results is meaningful). As seen from this table, there appears to be no significant difference in over-admission percentages across the different bottlenecks traversed by the "long-haul" flows in the PLT topologies. Furthermore, there is no visible performance difference in the case of multiple bottleneck topologies (PLT), compared to the case when only a single bottleneck is traversed (as in both SingleLink and RTT topologies) for the same demand overloads and parameters. We observed similar result all experiments we run. We ran these experiments for all traffic types, with similar results. Zhang, et al. Expires January 6, 2008 [Page 21] Internet-Draft CL Simulation Study July 2007

<table>
<thead>
<tr>
<th>Bottleneck LinkId</th>
<th>Type</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.286</td>
<td>0.238</td>
<td>0.332</td>
<td>0.306</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.420</td>
<td>0.257</td>
<td>0.341</td>
<td>0.254</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.394</td>
<td>0.312</td>
<td>0.268</td>
<td>0.205</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.309</td>
<td>0.223</td>
<td>0.363</td>
<td>0.317</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.420</td>
<td>0.257</td>
<td>0.341</td>
<td>0.254</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.7 Over-admission-percentage for PLT5 for all bottlenecks. The results are for CBR, 5x overload, averaged over all experiments with different parameter settings (there is no significant parameter sensitivity and the results for different settings are very close). 3.8.2. Fairness Between Long-haul and Short-haul flows Our next set of experiments targeted understanding the effect of multiple bottlenecks on the fairness of sharing the bottleneck links between the long- and the short-haul flows. It is generally known [Jamin, etc] that measurement-based admission control algorithms are susceptible to the effect when long-haul flows get a much smaller share of the bottleneck than short-haul flows. While the effect of unfairness is well known, the exact cause of it (and possibly the extent) depends
on the details of the algorithm. Our first goal was to understand the extent of the unfairness that might occur. Table 3.8 shows the ratio of the bandwidth achieved by the long-haul the short-haul aggregates with respect to the simulation time. As can be seen, the long-haul flow consistently loose bandwidth as a function of time, and this effect is the more pronounced the more bottlenecks are traversed by the long-haul flow. Zhang, et al. Expires January 6, 2008

<table>
<thead>
<tr>
<th>Topo</th>
<th>Weight</th>
<th>Simulation Time (s)</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLT5</td>
<td>0.5</td>
<td>1.02</td>
<td>1.07</td>
<td>1.19</td>
<td>1.24</td>
<td>1.30</td>
<td>1.34</td>
<td>1.33</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>1.02</td>
<td>1.09</td>
<td>1.23</td>
<td>1.41</td>
<td>1.65</td>
<td>2.10</td>
<td>2.63</td>
</tr>
<tr>
<td>PLT3</td>
<td>0.5</td>
<td>1.02</td>
<td>1.04</td>
<td>1.19</td>
<td>1.24</td>
<td>1.30</td>
<td>1.34</td>
<td>1.33</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>1.02</td>
<td>1.09</td>
<td>1.23</td>
<td>1.41</td>
<td>1.65</td>
<td>2.10</td>
<td>2.63</td>
</tr>
<tr>
<td>PLT2</td>
<td>0.5</td>
<td>1.02</td>
<td>1.06</td>
<td>1.14</td>
<td>1.17</td>
<td>1.15</td>
<td>1.31</td>
<td>1.41</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>1.02</td>
<td>1.04</td>
<td>1.11</td>
<td>1.30</td>
<td>1.56</td>
<td>1.61</td>
<td>1.62</td>
</tr>
</tbody>
</table>

Table 3.8. Unfairness ratio between long flow aggregate and short flow aggregate in time, for different PLT topologies and different EWMA rates. All results are for 5x overload, CBR traffic. Table 3.8 indicates that the bandwidth of the long-haul aggregate consistently declines in time, even though its demand remains constant. This effect is frequently referred as the "beatdown". Discouraging as it is, this effect is well known for Measurement-based admission control. The intuition behind the "beatdown" is that the long-haul aggregate can admit new flows only if all the links it transverse are not in the congestion state. Hence comparing to the short-haul aggregate, the long-haul ones see congestion more often, and is in the no-admission state substantially more often as well. If the demand loads of the short-haul and long haul flows are similar, and high enough to monopolize the entire bottleneck bandwidth, the long-haul flow repeatedly loses the competition and stays in the no-admission state most of the time. It is important to note that Table 3.8 indicates that the bandwidth of the long-haul aggregate consistently declines in time, even though its demand remains constant. In fact, in our simulation runs of about 80 simulation seconds long, we see that for all settings of the parameters and all PLT topologies we see a consistent decline of the share of the long-hauls aggregate. A question then arises on whether this effect continues (with the long-haul aggregate being eventually beaten-down
to zero) or whether the long-haul aggregate eventually stabilizes at some (perhaps low) value. Before attempting to answer this question we note that this effect is well known for Measurement-based admission control. In fact the authors of [Jamin] argue that for sufficiently large demands, in the Zhang, et al. Expires January 6, 2008 [Page 23] Internet-Draft CL Simulation Study July 2007 limit the long-haul flow is always beaten down completely. In our simulations, however, the demands at which the beatdown effect occurs is not at all infinitely large. We investigate why, even at demand overloads as small as 2X the configured-admissible-rate at the bottleneck, the long-haul aggregate is consistently beaten down. Our analysis indicates that for all parameter settings, the proportion of time at least one of the links in the topology is in the "pre-congestion state", i.e. marking enough packets to trigger no-admission is substantially higher than the percentage of time any one of the links spends in the pre-congestion state, and in many cases it is close to 100% of the time. It seems clear that the fraction of time the long-haul aggregate on the average sees congestion on at least one of the links is a critical parameter defining whether or not the long-haul flow will be eventually starved completely or not. Clearly if the fraction of time the long-haul sees no congestion on all links along the way is close to 1, then it simply never has a chance to admit new flows, and eventually gets beaten down to zero. On the other hand, if the long-term average fraction of time when it sees no congestion simultaneously on all of the links it traverses stays above some \( f > 0 \) for any long enough period of time, if the demand (or the rate of calls requesting admission) of the aggregate is constant, then the beatdown effect would never drive the long-haul flow below (call arrival rate) times \( f \) times (mean call duration). This can be easily seen by observing that if for the fraction of time \( f \) the aggregate is allowed to admit, then it effective long-term call acceptance rate is (call arrival rate) \( \times f \). When there are \( N \) flows of the aggregate in the system, the mean call departure rate is \( N/(\text{mean call duration}) \). Therefore, if the number of admitted flows in the aggregate ever reaches or goes below \( k=(\text{call arrival rate}) \times f \times (\text{mean call duration}) \), the mean call arrival rate will become larger than the mean departure rate, and hence the number of flows will be increasing on the average. How realistic is it that at least one of the links traversed by the long-haul aggregate is always in congestion/marking state? While we do not know the general answer, we can argue that if congestion states of different links were independent, and each link \( j \) is in the state of congestion some fraction of time \( p(j) \), then the fraction of time \( p \) that a flow traversing \( n \) bottlenecks sees congestion on at least one link is \( p = 1 - (1 - p(1)) * (1 - p(2)) * ... * (1 - p(n)) \). If \( n \) is large and/or demands on the bottlenecks are large, this \( p \) is close to 1. In our simulations, with 5 bottlenecks and 5x overload, the fraction of time when at least one of the links was marking packets was close to 100%. Of course in general we...
cannot assume that the "congestion" state in different links is completely independent. Yet, in all our simulations, it appears that even when Zhang, et al. Expires January 6, 2008 [Page 24] Internet-Draft CL Simulation Study July 2007 the system is started in a synchronized state where all the bottlenecks are "congested" at the same time, the system tends to get desynchronized in time, so that congestion periods at different bottlenecks spread in time, and in many experiments with 5-PLT almost all the time at least one of the links was in the congestion state. We observed consistent beatdown effect across all experiments, although the exact extent of the unfairness depends on the demand overload, topology and parameters settings. To further quantify the effect of these factors remains an area of future work. We also note that the cause of the beatdown effect appears to be largely independent of the specific algorithm, and is likely to be relevant to other PCN proposals as well. Finally we note that the for the beatdown effect to be significant, not only the demand overload amount should be substantial, but also the duration of the demand overload should be long enough. Under "normal" conditions, one should not expect prolonged substantial overloads. In the exceptional cases where high overloads do occur, they are likely to not be of very large duration. In those cases, unfairness and even starvation of some aggregates is still preferential to indiscriminately dropping packets of all flows that would occur in the absence of admission control. Hence, in practice, the effect of the beatdown effect we report here is probably limited.

4. Termination Control 4.1. Termination Model and Key Parameters We evaluate the termination algorithm on all the topologies described in Section 2. In the simulation, the router implementing PCN termination Marking operates as described in [I-D.briscoe-tsvwg-cl-architecture], marking all packets which find no token in the token bucket. In the case of multiple bottlenecks, only previously unmarked traffic is metered against the token bucket. When an egress gateway receives a marked packet from the ingress, it will start measuring its Sustainable-Aggregate-Rate for this ingress, if it is not already in the Termination mode. If a marked packet arrives while the egress is already in the Termination mode, the packet is ignored. The measurement is interval based, with 100ms measurement interval chosen in all simulations. At the end of the measurement interval, the egress sends the measured Sustainable-Aggregate-Rate to the ingress, and leaves the termination mode. When the ingress receives the sustainable rate from the egress, it starts its own interval immediately (unless it is already in a measurement interval), and Zhang, et al. Expires January 6, 2008 [Page 25]
measures its sending rate to that egress. Then at the end of that measurement interval, it terminates the necessary amount of traffic. The ingress then leaves the termination mode until the next time it receives the sustainable rate estimate from the egress. In all our simulations
the ingress used the same length of the measurement interval as the egress. Token bucket depth was set to 256 packets in all experiments presented here. We evaluate the performance of the algorithms using a metric called "over-termination-percentage", which is defined as (actual-termination - optimal-termination) expressed in percentage of the optimal termination value. We apply this metric in two contexts: (1) the aggregate amount of terminated traffic on a given bottleneck link, and (2) the aggregate amount of terminated traffic of an ingress-egress traffic aggregate. The former relates to bottleneck utilization, and is quite straightforward: the optimal Termination would terminate all traffic above the configured-termination-rate, so "optimal" Termination is defined only by the configured-termination-rate. For the ingress-egress aggregates, the notion of optimality is closely related to the notion of fairness. In general, fairness can be defined in many different ways, and we do not attempt to argue for one being "more optimal" than the other. In this draft we call the per-ingress-egress Termination amounts optimal if the amount of terminated traffic is distributed among all ingress-egress pairs sharing a bottleneck link in proportion to their rates prior to Termination. For brevity, we omit the details of the definition for the multiple bottleneck case here as it is not central to the discussion in this draft.

4.2. Effect of RTT Difference

Our experiments indicate that absolute value of RTT within the chosen range (up to 220 ms) has no effect on the performance of the termination algorithm, as long as the RTTs of the different ingress-egress pairs are comparable. This section investigates the impact of the relative difference or RTTs of different flows sharing a single bottleneck. We show that in principle, when both short- and long-RTT ingress-egress pairs are present, the difference in RTT may cause over-termination. To demonstrate that we consider a simple RTT topology with two ingresses, with CBR traffic. Table 4.3 shows the experiment setup and termination results. The overall traffic on the bottleneck during the event is 1761 CBR flows, which constitutes 75% of OC3 link. Ingress 2 has a RTT that around 50ms larger than Ingress 1. The actual termination (termination) and the over-termination percentage are listed for each ingress separately. The results shows that Ingress 1 over-terminates about 10% of its traffic, which Zhang, et al. Expires January 6, 2008 [Page 26]
Internet-Draft CL Simulation Study July 2007 results in about 6% of the overall over-termination at the bottleneck.

<table>
<thead>
<tr>
<th>Ingress</th>
<th>Bottleneck</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTT</td>
<td>Actual</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>1178</td>
</tr>
<tr>
<td>2</td>
<td>583</td>
</tr>
</tbody>
</table>

Table 4.3. Summary of the RTT difference Results. Figure 4.3 shows a time vs. load graph that is intended to capture the effect of the flow termination.

algorithm in this experiment. The X-axis is the time, where a number of important time points are labeled (actual time is listed in table due to lack of space). The Y-axis is the load on the bottleneck link. The stacked graph on the right shows the behavior of each individual ingress. (The shade region is the load contributes to Ingress 1 and the clear region corresponds to Ingress 2). Finally, the dotted line represent the configured-termination-rate. Zhang, et al. 

Study July 2007 |_L1| | | | | | | | | | | | | | | | |___ | | | | | L2|.....|......|___............. |___...
| | L ** | | o | | o | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | _|_________________ t1 t2 t3 t4 t1 t2 t3 t4 Time Time
--------------------------------- | t1 | t2 | t3 | t4 |
--------------------------------- | 200.0 | 200.2 | 200.25 | 200.40 |
------------------------- ------- Fig 4.4. Time series of termination events in the RT Difference experiment As the simulated failure event occur at time t1 (200s), the load on the bottleneck goes over the configured-termination-rate by 1/3, thereby activating the termination algorithm. 200ms afterward at t2, which is sum of the measurements of sustainable rate at the egress (100 ms) and the consequent ingress measurement of its current sending rate, Ingress1 with negligible RTT (1ms) start terminating its traffic. 50ms later at t3, Ingress 2 terminates its share of traffic. Note, at this point, both of ingresses had terminated the correct amount, which is why the load on bottleneck between time t3 and t4 is exactly at the configured-termination-rate. However the stacked graph shows that Ingress1 did another around of termination at t4 (200.4), which corresponds to its 10% over-termination. The reason for this effect is that during the interval between t2 and t3, when Ingress1 finishes its flow terminations, and Ingress2 has not yet started due to its longer RTT, the non-terminated traffic from Ingress2 will cause a further decrement in Ingress1’s sustainable rate during the measurement interval (t2, t2+100ms). This will in turn cause Ingress1 to terminate at time t4 to compensate for that 50ms of excess traffic from Ingress2. Our follow-up results indicate Zhang, et al. Study July 2007 that this RTT effect exists to some degree in every experiment that has sufficient Ingress RTT difference, independent of the traffic type. Although for burstier traffic the over-termination may be worse than shown above, in our experiments we did not see over-termination that would be drastically larger. However, further investigation is needed to access whether other scenarios might lead to more substantial over-termination. 4.3.
Ingress-Egress Aggregation Experiments 4.3.1. Motivation for the Investigation While sufficiently high bottleneck aggregation is listed as one of the underlying assumptions of [I-D.briscoe-tsvwg-cl-architecture], there remains a question of whether or not sufficient degree of aggregation of traffic on a per ingress-egress pair is also necessary. We saw that in our admissions experiments, the virtual-queue-based admission algorithm performed reasonably well even with small ingress-egress aggregation levels, as long as the bottleneck aggregation level was sufficiently high. A similar investigation is performed for the case of termination. Assuming a large degree of aggregation on a per ingress-egress pair is less attractive, as one can easily imagine that a bottleneck link in a PCN region may carry traffic from hundreds or thousands of ingresses, and there is evidence to believe that in practice cases when per- ingress-egress pair traffic is generated by a relatively small number of flows may not be uncommon. If indeed the number of flows in an ingress-egress pair is small, theoretically there exists a concern that the granularity of termination (which can operate on integer number of flows only) will result in large inaccuracies of the amount of traffic terminated in a per-ingress-egress aggregate, and consequently a large amount of over-termination. As an example of a situation creating this problem suppose that a bottleneck link is shared by 2N flows, each one of them coming from a different ingress-egress pair. Suppose that only N flows can be supported at the configured-termination-rate, so N out of 2N flows must be terminated. This means that half of the packets will get termination marked. If these marked packets are more or less uniformly distributed among the flows sharing the bottleneck, one should expect that every one of the 2N flows will have half of its packets marked. That in turn would imply that each ingress would need to terminate half of its traffic, and since it only has one flow, it would have to terminate that flow (assuming that the number of flows to terminate is rounded up to the nearest flow) or not terminate any flow at all (if the rounding down to the nearest flow is done). In either case the outcome is quite pessimistic- either all flows are terminated, or the termination will not take any effect at all. Clearly, a similar Zhang, et al. effect would be if a few flows rather than one constitute an ingress-egress pair. The effect quickly disappears when the rate of an individual flow is sufficiently small compared to the total rate of the ingress-egress aggregate. While a number of possible changes to the ingress behavior could be considered to solve or alleviate this problem, we set out to investigate whether this problem does in fact occur in practice. The key question in that respect is whether or not the packets do indeed get marked more or less uniformly among different flows sharing a bottleneck over the timescale of the ingress and egress measurement intervals. The results of this investigation are
presented in the following subsections. 4.3.2. Detailed results To investigate the effect of small ingress-egress aggregation, we first performed the experiments with three traffic types (CBR, VBR and SVD) at different degrees of ingress aggregation. All the experiments in this section are carried out on RTT topology; the different ingress aggregation levels are obtained by varying the number of ingress links in the topology. All links’ RTT are set to 1ms (to eliminate the potential RTT influence). CBR and VBR voice used an OC3 bottleneck link while SVD used an OC48 link, with configured-termination-rate set at 50% of the link bandwidth in all cases. The bottleneck aggregation was therefore quite high (with respect to the corresponding link bandwidth), but the ingress-egress aggregation was varied from 1 flow to about 1/3 of the number of flows at the bottleneck in each ingress-egress pair. The results are summarized in Table 4.1 below. Zhang, et al. Expires January 6, 2008 [Page 30]
<table>
<thead>
<tr>
<th>Model</th>
<th>Load</th>
<th>Ingress</th>
<th>Ingress</th>
<th>Threshold</th>
<th>Actual</th>
<th>Over-termin</th>
<th>Perc</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBR</td>
<td>1789</td>
<td>2</td>
<td>582</td>
<td>0.321</td>
<td>0.05%</td>
<td>0.321</td>
<td>0.05%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.328</td>
<td>1.41%</td>
<td>0.328</td>
<td>1.41%</td>
</tr>
<tr>
<td>VBR</td>
<td>5336</td>
<td>2</td>
<td>1759</td>
<td>0.333</td>
<td>0.35%</td>
<td>0.333</td>
<td>0.35%</td>
</tr>
<tr>
<td></td>
<td>3574</td>
<td></td>
<td></td>
<td>0.364</td>
<td>2.84%</td>
<td>0.364</td>
<td>2.84%</td>
</tr>
<tr>
<td>SVD</td>
<td>450</td>
<td>2</td>
<td>135</td>
<td>0.404</td>
<td>8.07%</td>
<td>0.404</td>
<td>8.07%</td>
</tr>
<tr>
<td></td>
<td>452</td>
<td></td>
<td></td>
<td>0.406</td>
<td>8.02%</td>
<td>0.406</td>
<td>8.02%</td>
</tr>
</tbody>
</table>

Table 4.1: Effect of ingress-egress aggregation. In this table, bottleneck load at failure is represented as the number of flows at the bottleneck after the simulated failure event has occurred and before the termination takes place. The "Number Ingress" column shows the number of ingresses in the RTT topology. In all cases, ideally, the algorithm should terminate roughly 1/3 of the traffic after the failure event has occurred (the exact percentage differs slightly from experiment to experiment due to some variability of load generation implementation). The second to last column shows the actual termination percentage in each experiment, and the last column shows how far it deviates from the optimal value in terms of over-termination percentage (where the optimal value is computed based on the actual traffic generated in each experiment). The first conclusion that can be drawn from Table 4.1 is that in these experiments flow termination worked quite well for CBR and VBR, and even in the SVD case with just 1 flows per ingress the over-termination is quite bounded. The second - far more unexpected - outcome of these results is that for all traffic types in these experiments the result show no appreciable effect of the ingress.
aggregation on the degree of ingress aggregation, as all the over-
termination percentage do not differ significantly. Given the
discussion in the previous section that predicted substantial
inaccuracy of flow termination in the case of a small number of flows
per ingress, this result appears both unexpected and encouraging, but
does require explanation and discussion. Further analysis of the
simulation traces of CBR traffic of Zhang, et al. turned out that in all the simulation runs with CBR traffic, contrary
to our expectation that marking will be more or less
uniformly distributed among active flows, what actually happens is
that some flows get all their packets marked, while other flows get
no packets marked at all (we refer to this effect loosely as
"synchronization" in the rest of this document). It is this
phenomenon that, in the case of a single flow per ingress, made only
the ingresses whose flows were marked terminate these flows,
resulting in correct amount of termination. Further analysis showed
that in fact this effect is not a simulation artifact, and is a
direct consequence of periodicity of individual CBR flows in
combination with incidental choice of several parameters. As it
happens, if the number of tokens arriving in the token bucket in an
inter-packet interval of a single CBR flow is an integer multiple of
a packet size, then if a packet of a flow is marked once, all the
subsequent packets will find the same number of tokens in the token
bucket and will also be marked. The proof of this fact is provided
in the companion technical report. It seems clear that in general
this synchronization cannot be relied upon, and we expected that for
the VBR case we will see much less of it. Again, we were in for a
surprise, as trace investigation of our initial results reported in
Table 4.1 revealed that even though the token bucket state
encountered by the packets of the same VBR flow was not quite the
same, it was close enough so that again a large number of flows were
either fully marked or fully unmarked. We realized that the reason
for that is that the number of flows which are in the on-period
during the relevant measurement intervals is relatively stable, and
hence much of the effects observed for the CBR flows approximately
holds for the on-off traffic we use for our VBR model. Since the on-
period had the same rate as our CBR model, and the packet size was
the same for the two models, similar behavior was observed in both
sets of experiments. In our quest to further understand the
unexpectedly reasonable performance at small ingress-egress
aggregation we then tested the hypothesis that randomizing the packet
inter-arrival time must surely break synchronization, and to that end
we rerun the same set of experiments on the randomization version of
all traffic. The results are summarized in Table 4.2 Note, the
column label with f (e.g. 0.0001) correspond to randomized traffic
with a randomization interval of f x packet-inter-arrival-time. It
also means that on average, the packets are delayed by $f \times \text{packet-inter-arrival-time} / 2$. Zhang, et al. Expires January 6, 2008

Table 4.2 Effect of ingress-egress aggregation v.s. deviation. The table entries correspond to the over-termination-percentage at different aggregation levels, different randomization interval, for different traffic types. As can be seen, the over-termination-percentage shown in Table 4.2 Zhang, et al. Expires January 6, 2008 exhibits different trends depending on the traffic types. First for CBR,

<table>
<thead>
<tr>
<th>No.</th>
<th>Deviation Interval</th>
<th>Ingre</th>
<th>No-Rand</th>
<th>0.0001</th>
<th>0.001</th>
<th>0.005</th>
<th>0.01</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.050</td>
<td>0.390</td>
<td>1.047</td>
<td>0.224</td>
<td>0.757</td>
<td>1.072</td>
<td>10</td>
<td>0.495</td>
</tr>
<tr>
<td>0.771</td>
<td>0.769</td>
<td>1.016</td>
<td>0.975</td>
<td>0.819</td>
<td>35</td>
<td>1.157</td>
<td>1.615</td>
<td></td>
</tr>
<tr>
<td>1.817</td>
<td>2.448</td>
<td>1.938</td>
<td>2.300</td>
<td>CBR</td>
<td>70</td>
<td>0.841</td>
<td>2.428</td>
<td>3.693</td>
</tr>
<tr>
<td>3.098</td>
<td>3.710</td>
<td>3.528</td>
<td></td>
<td>140</td>
<td>1.577</td>
<td>3.089</td>
<td>4.962</td>
<td>5.271</td>
</tr>
<tr>
<td>-----</td>
<td>------------------</td>
<td>-------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>2</td>
<td>2.663</td>
<td>3.002</td>
<td>2.350</td>
<td>2.320</td>
<td>2.240</td>
<td>2.217</td>
<td>10</td>
<td>1.856</td>
</tr>
<tr>
<td>3.629</td>
<td>3.369</td>
<td>1.460</td>
<td>2.493</td>
<td>3.712</td>
<td>35</td>
<td>2.831</td>
<td>3.385</td>
<td></td>
</tr>
<tr>
<td>3.931</td>
<td>4.128</td>
<td>4.918</td>
<td>4.490</td>
<td>VBR</td>
<td>100</td>
<td>3.952</td>
<td>6.257</td>
<td>5.018</td>
</tr>
<tr>
<td>5.732</td>
<td>5.099</td>
<td>5.568</td>
<td></td>
<td>300</td>
<td>2.421</td>
<td>4.846</td>
<td>5.435</td>
<td>5.651</td>
</tr>
<tr>
<td>4.856</td>
<td></td>
<td>1800</td>
<td>2.500</td>
<td>0.435</td>
<td>2.248</td>
<td>2.727</td>
<td>1.698</td>
<td>2.077</td>
</tr>
<tr>
<td>-----</td>
<td>------------------</td>
<td>-------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>2</td>
<td>1.173</td>
<td>1.863</td>
<td>1.185</td>
<td>1.341</td>
<td>1.736</td>
<td>1.164</td>
<td>10</td>
<td>2.377</td>
</tr>
<tr>
<td>1.579</td>
<td>1.207</td>
<td>2.656</td>
<td>2.321</td>
<td>3.097</td>
<td>35</td>
<td>2.914</td>
<td>3.484</td>
<td></td>
</tr>
<tr>
<td>2.232</td>
<td>2.665</td>
<td>2.642</td>
<td>3.874</td>
<td>MIX</td>
<td>140</td>
<td>4.066</td>
<td>3.009</td>
<td>4.600</td>
</tr>
<tr>
<td>3.221</td>
<td>4.263</td>
<td>4.907</td>
<td></td>
<td>300</td>
<td>2.806</td>
<td>5.436</td>
<td>4.088</td>
<td>4.113</td>
</tr>
<tr>
<td>4.895</td>
<td>5.602</td>
<td></td>
<td>600</td>
<td>1.995</td>
<td>0.527</td>
<td>3.295</td>
<td>2.982</td>
<td>4.851</td>
</tr>
<tr>
<td>4.424</td>
<td></td>
<td>1000</td>
<td>0.733</td>
<td>1.076</td>
<td>1.729</td>
<td>3.022</td>
<td>3.955</td>
<td>2.731</td>
</tr>
<tr>
<td>-----</td>
<td>------------------</td>
<td>-------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>2</td>
<td>2.544</td>
<td>2.610</td>
<td>1.217</td>
<td>3.055</td>
<td>2.889</td>
<td>1.906</td>
<td>10</td>
<td>3.741</td>
</tr>
<tr>
<td>6.629</td>
<td>7.014</td>
<td>7.644</td>
<td>6.610</td>
<td>VTR</td>
<td>70</td>
<td>7.607</td>
<td>7.541</td>
<td>8.217</td>
</tr>
<tr>
<td>7.002</td>
<td>7.916</td>
<td>8.113</td>
<td></td>
<td>140</td>
<td>11.16</td>
<td>9.162</td>
<td>12.50</td>
<td>10.74</td>
</tr>
<tr>
<td>10.49</td>
<td>10.44</td>
<td></td>
<td>300</td>
<td>12.15</td>
<td>14.41</td>
<td>13.57</td>
<td>14.32</td>
<td>15.62</td>
</tr>
<tr>
<td>16.35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>------------------</td>
<td>-------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>2</td>
<td>8.071</td>
<td>10.62</td>
<td>8.235</td>
<td>10.22</td>
<td>10.62</td>
<td>8.531</td>
<td>10</td>
<td>10.66</td>
</tr>
<tr>
<td>12.13</td>
<td>11.16</td>
<td>10.81</td>
<td>11.35</td>
<td>10.61</td>
<td>35</td>
<td>10.69</td>
<td>10.14</td>
<td></td>
</tr>
<tr>
<td>6.008</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

we expected, the "randomization" indeed breaks in the
synchronization, so that at low aggregation, we observe substantially
more over-termination (~14%), confirming our expectation that the
unexpectedly good performance cannot be expected in general for low
ingress-egress aggregation levels. On the other hand, it also shows
that at least a certain amount of randomization is required to break
the "synchronization". For instance, with a randomization interval
of the 0.0001 x packet-inter-arrival-time, no substantial increment
in over-termination is observation. From this aspect, the
"synchronization" effect can not be merely regarded as a simulation
artifact. A final note is that given sufficient amount of
aggregation (~10 call/ingress or above), the difference caused by
synchronization goes away. VBR shows a different trend. It seems
that given enough randomization, the effect of aggregation (over-
termination) starts to emerge at the transition from medium to low
aggregation level (around 100 or 300 ingress in the graph). However
the effect then diminishes, so that at the lowest aggregation
(expected 1 flow per ingress), we no longer observe appreciable over-
termination. We believe the reason for this outcome the
following: at medium aggregation levels, even though there are a few
flows per ingress, it's not enough to smooth out the burstiness in
the aggregated flows. This causes each ingress-egress-pair to over-
terminate a little due to occasional under-estimation of the
Sustainable-Aggregate-Rate, which results in the net over-termination at
the bottleneck link. At the low aggregation level (with 1 or 2 flow
per ingress), each VBR flow spends a large portion of its time in
off-period. Once the aggregate of an ingress-egress pair is in its
off period, it will send no packets, get no marking, hence will not
react to the termination algorithms. Since a substantial portion of
the ingress-egress aggregates can be in the off-period, only those
ingresses that are in the on-period terminate their traffic. The MIX
essentially shows the added up effect of both CBR and VBR, in the
sense that it shows a clear increasing of over-termination-
percentage as the level of aggregation decreases, yet at the lowest
aggregation, instead of having the highest over-termination (like
CBR), the on-off effect of VBR dominates, hence again don't see a
significant over-termination For TRC, a clear aggregation effect is
observed, but the trends seems to be irrelevant to the degree of
randomization. In fact, the result for TRC looks like a complete
randomized version of CBR. We hypothesize this is indeed the case,
since trace is implemented as constant-frame-rate, that’s why it
doesn’t exhibit what appears in the VBR (namely the on-off effect),
in addition, different frame Zhang, et al. Expires January 6, 2008
size, and packetization provide enough randomization. The trace analysis of
the SVD experiments indicates that there are a large number of of
partially marked flows, which indicates that synchronization could
not have been responsible for the relatively bounded over-termination
of about 10% Table 4.2. We believe this performance should be traced to the burstiness of our crude SVD traffic model at the time scales commensurate with the measurement period. In addition, just as for VBR, the on-off nature of the model dominates at low aggregation, which can also be used to explain why no aggregation effect is observed. In summary, the over-termination can be expected at low aggregation for a variety of traffic, but in practice the degree of this over-termination is not as bad as the worst-case analysis might indicate. The over-termination vanishes as the level of ingress-egress aggregation becomes sufficiently large. 4.4. Multiple Bottlenecks Experiments 4.4.1. Motivation for the Investigation In this section, we focus our analysis on the multi-bottleneck effect. That is, how would termination algorithm perform when the flows from one (or more) ingress-egress pairs traverse multiple bottleneck links. For the rest of section, we use the term "IE- aggregate" (IEA for short) to refer to the flow aggregates of a certain ingress-egress pair. In theory, we expect the IE-aggregate that travel more bottlenecks will be penalized more, which would result in over-termination on a per-ingress-egress basis. We refer to this as a "beat-down" effect. The main consequence of the beat-down effect is the excessive termination at the up-stream bottlenecks, leading to underutilization of those bottlenecks. To illustrate the beat-down effect, consider the setup with 2 bottleneck PLT in Figure 2.3(a). Recall the two bottlenecks are links A - B and B - C. Both links have the same capacity. There are two short IE-aggregates, one from Ingress D to Egress E (IEA2); the other from Ingress E to Egress F (IEA3); each traversing a single bottleneck. At the time of the failure event, each short IEA carries the traffic load that equals 1/4 of the bottleneck link size (or 1/2 of the configured-termination-rate, which in this case is set to 50% of the link bandwidth). The long IE-aggregate (IEA1), from Ingress A to Egress C, traverses both of bottlenecks and carries twice as much traffic as the short ones. Given that we set the configured-termination-rate to be 1/2 of link size, it’s easy to see that letting all IEAs terminate 1/3 of their flows will give the optimal results (which we refer to as "optimal-termination") in the sense that all bottleneck links will be fully utilized.
Internet-Draft CL Simulation Study July 2007 utilized. However, what we expect to happen is the following. When the long IE-aggregate (IEA1) traverses through the first bottleneck link, assuming uniformly random marking, 1/3 of its traffic will get termination-marked. (The short IEA2 will also get 1/3 of its traffic marked). Next, 2/3 of IEA1’s unmarked traffic together with IEA3’s traffic will result a load of (2/3)*(1/2) + 1/4 = 7/12 on the second bottleneck. This implies that for the aggregate IEA1, an additional (7/12-1/2) / (7/12) = 1/7 percentage of remaining unmarked traffic will be marked. And for IEA3, only 1/7 (instead of 1/3) of its traffic will be marked. To summarize, a beat-down effect in this
simple setting means we should see the following termination behaviors:

- **EA1**: \( \frac{1}{3} + \frac{2}{3} \times \frac{1}{7} = \frac{3}{7} > \frac{1}{3} \)
- **IEA2**: \( \frac{1}{3} \)
- **IEA3**: \( \frac{1}{7} < \frac{1}{3} \)

**Bottleneck1**: \( \left( \frac{3}{7} \times \frac{1}{2} + \frac{1}{3} \times \frac{1}{4} \right) / \left( \frac{3}{4} \right) = \frac{25}{63} > \frac{1}{3} \)

We refer to the above values as "expected-termination". In general, the more bottlenecks an IEA traverses, the more over-termination occurs at both the long IEA and the upstream bottlenecks. The goal of our experiments was to validate to what extent the beat-down effect is visible in practice, and how much underutilization on up-stream links will actually be seen. To that end, we used 2, 3 and 5 PLT topologies with various traffic types. We are interested in whether the actual-termination exhibits the multi-bottleneck effect comparing to the optimal-termination, and also how much does the actual-termination deviate from our expected-termination. The results of this investigation are presented in the following subsections.

### 4.4.2. Detailed Results

For the first set of experiments, we use the similar setup as the example described in last subsection. That is, at failure event time, all bottleneck links have a load of roughly \( \frac{3}{4} \) of its link size. In addition, the long IEA constitutes \( \frac{2}{3} \) of this load, while the short one is \( \frac{1}{3} \).

Table 4.7 shows the sample output for the multi-bottleneck experiments (In this case, it’s with CBR traffic and 5 PLT topology). The first row (labeled IEA1) represents the long IE-Aggregate that travels multiple bottlenecks (the exact count of the bottlenecks is given in the parenthesis after the IEA’s name). Zhang, et al. Expires January 6, 2008 [Page 36] Internet-Draft CL Simulation Study July 2007 The rest of IEA rows are the short IE-Aggregates that each travels only one bottleneck. The IEA rows are ordered based on the bottleneck it traverses (from upstream to downstream). The same information is shown for both IEAs and bottlenecks. The last two columns are of most interests in that they shows the how far the actual-termination deviates from the optimal, and from the expectation.

<table>
<thead>
<tr>
<th>Optimal</th>
<th>Expected</th>
<th>Actual</th>
<th>A - O</th>
<th>A - E</th>
<th>term</th>
<th>term</th>
<th>term</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEA1 (5H)</td>
<td>0.3090</td>
<td>0.4432</td>
<td>0.4446</td>
<td>13.56</td>
<td>0.14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3090</td>
<td>0.3090</td>
<td>0.3231</td>
<td>1.42</td>
<td>1.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1181</td>
<td>0.1601</td>
<td>-14.33</td>
<td>4.20</td>
<td>5 IEA1 (5H)</td>
<td>0.3048</td>
<td>0.0541</td>
<td></td>
</tr>
<tr>
<td>0.0947</td>
<td>-21.01</td>
<td>4.07</td>
<td>IEA1 (5H)</td>
<td>0.3073</td>
<td>0.0293</td>
<td>0.0641</td>
<td></td>
</tr>
<tr>
<td>-24.32</td>
<td>3.48</td>
<td>BN1 IEA1 (5H)</td>
<td>0.3031</td>
<td>0.0049</td>
<td>0.0307</td>
<td>-27.24</td>
<td></td>
</tr>
<tr>
<td>2.57</td>
<td>R BN1</td>
<td>0.3090</td>
<td>0.3995</td>
<td>0.4051</td>
<td>9.61</td>
<td>0.56</td>
<td></td>
</tr>
<tr>
<td>0.3034</td>
<td>0.3392</td>
<td>0.0348</td>
<td>0.3182</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3322</td>
<td>2.73</td>
<td>1.40</td>
<td>BN4</td>
<td>0.3073</td>
<td>0.3214</td>
<td>1.41</td>
<td>1.22</td>
</tr>
<tr>
<td>0.3031</td>
<td>0.3031</td>
<td>0.3123</td>
<td>0.92</td>
<td>0.92</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.7 Over-termination percentage with 5-PLT topology and CBR

The following Table 4.8 summarizes the main results for multi-bottleneck
experiments. For each combination of the traffic type and PLT topology, it shows (actual-termination - optimal-termination)*100% (labeled as 'A-O') and (actual-termination - expect-termination)*100% (labeled as 'A-E'). Zhang, et al. Expires January 6, 2008 [Page 37]
Table 4.8 Summary of the PLT results for 2:1 long-to-short load ratio. It's clear from the 'A-O' results that the beat-down effect is visible across all PLT topologies and traffic types. For instance, for the long IE-aggregate (IEA1), as it travels 2, 3, 5 bottlenecks, the degree of over-termination increases (7.61, 10.85, 13.56 respectively for CBR traffic); so does the most upstream bottleneck link (BN1). Furthermore, all the downstream short IEAs (IEA3 and above) have experienced under-termination compared to their "optimal" value, while the long IEA terminated more than the "optimal" value. Next we compare the actual-termination with the level of termination predicted by the theoretical beat-down effect based on the assumption of uniformly random marking. Our experience reported in the previous section shows that the assumption of uniform marking may not always hold in the case of bursty traffic. Zhang, et
As seen from Table 4.8, the results for CBR, VBR and VTR are reasonably close to those predicted by the beat-down effect (within 1% for CBR and within 3% for VBR and VTR). The larger discrepancy between the expected and the actual results for SVD are most likely the consequence of the same burstiness effect that we observed in the previous section with respect to ingress-egress aggregation experiments. Recall that in all of above experiments, we had the long IE-aggregate carries the traffic twice as much as the short ones. Now we investigate what will happen if this load ratio changes. We can use the same method (as the one illustrated in the last subsection) to obtain the expected-termination for any given PLT topology. The expected trend is that, keeping all other conditions the same, the smaller portion the long IEA is, the more relative unfairness towards it (percentage-wise) will be displayed. In following set of experiments we chose the 1:1 as the load ratio (instead of 2:1) of the long and short aggregates, while keeping other the settings unchanged. The results, (actual-termination - optimal- termination)*100%, are summarized in Table 4.9.

<table>
<thead>
<tr>
<th>CBR</th>
<th>VTR</th>
<th>2:1 1:1</th>
<th>2:1 1:1</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEA1(2H)</td>
<td>7.61</td>
<td>10.74</td>
<td>9.19</td>
</tr>
<tr>
<td>IEA3(1H)</td>
<td>-14.49</td>
<td>-10.27</td>
<td>-14.49</td>
</tr>
<tr>
<td>IEA3(1H)</td>
<td>-12.50</td>
<td>-7.23</td>
<td>-9.71</td>
</tr>
<tr>
<td>IEA4(1H)</td>
<td>5.45</td>
<td>9.19</td>
<td>7.24</td>
</tr>
<tr>
<td>IEA5(1H)</td>
<td>3.18</td>
<td>2.18</td>
<td>3.17</td>
</tr>
<tr>
<td>IEA6(1H)</td>
<td>2.85</td>
<td>3.85</td>
<td>3.18</td>
</tr>
</tbody>
</table>

Table 4.9 Summary of the PLT results for 1:1 long-to-short load ratio. The results confirm our expected behavior. For instance, the row that gives the over-termination of the IEA1 that goes through 3 bottleneck links shows that in the 1:1 ratio setup, the over-termination of the long aggregate is much larger comparing to 2:1 setup. And the problem grows severely when the number of bottleneck link increases (see IEA1(5H)). Furthermore, the increment in over-termination of the long IEA also reflects on the bottleneck link, that is, the aggregated over-termination perc. on the bottleneck link increases accordingly.
The 'A-E' part of the results is very similar to the ones in Table 4.5. That is, for CBR, VBR, VTR, we have the actual-termination close to expectation. A high-level conclusion of the results presented in this section is that the actual results confirm the predicted beat-down effect Zhang, et al. Expires January 6, 2008

4.5. Sensitivity to Call Arrival Assumptions In this section we investigate to what extent the Poisson call arrival assumption affect the accuracy of the termination control algorithm. To that end we investigated the comparative performance of the algorithm with Poisson and BATCH call arrival processes for the all traffic. The mean call arrival rate was the same for both processes, with a batch mean equal to 5. With sufficient level ingress-egress aggregation, the BATCH arrivals experiments give very similar results to the ones with Poisson arrivals. However, contrary to what we expected, in the case of low ingress-egress aggregation, the BATCH model actually performs better. This is simply because the combination of the BATCH arrival and low aggregation makes the traffic aggregate more "on/off"- like. Hence the same reason that VBR and SVD is not affected by the low aggregation (discussed in Section 4.3), can be applied here.

5. Summary of Results The study presented here demonstrated that overall, both admission control and termination algorithms of [I-D.briscoe-tsvwg-cl-architecture] work reasonably well and are relatively insensitive to parameter variations. We can summarize the conclusions of the study so far as follows.

5.1. Summary of Admission Control Results

- We observed no significant benefit of using "ramp" marking instead of a simpler "step" marking.
- There appears to be no appreciable sensitivity of the admission algorithm to either the absolute value of the round-trip time or the relative value of the round-trip time between different flows.
- As a rule of thumb, the level of bottleneck aggregation necessary to demonstrate tolerable performance even in the simplest network topology corresponds to links of about 10 Mbps or higher for voice traffic (CBR of VBR with silence compression), assuming at least 50% of the link speed is allocated to the PCN traffic. For higher rate bursty SVD flows, 50% of the OC48 of higher appears to be a Zhang, et al. Expires January 6, 2008
reasonable rule of thumb. The higher the degree of bottleneck aggregation, the better
the performance. o Even though larger per ingress-egress pair
aggregation results in better performance of admission control
algorithm, performance remains reasonable even for really low
ingress-egress aggregation levels (i.e. a single or a small number of
bursty SVD flow per ingress). o Poisson call arrival has a visible
effect on performance at lower levels of aggregation (10 Mbps for
voice or lower), but is of less significance at the higher levels of aggregation/link speeds. The algorithm is relatively insensitive to variation of key parameter settings at the internal node or the ingress of the PCN domain, as long as the variations are kept within a reasonable range around "sensible" parameter settings. As expected, synthetic video traffic SVD was the most challenging for all topologies, and the performance of real video traces (VTR) was substantially better. Even for the SVD, however, a range of parameters exist for which performance across all experiments considered is within reasonable bounds. The algorithms is relatively insensitive to the level of ingress-egress aggregation. No performance degradation is observed on the bottleneck link, in a multi-bottleneck topology where some flows traverse multiple bottlenecks in the presence of cross-traffic on each of the bottleneck links. However, the algorithm suffers from the well-known phenomenon of unfairness towards flows traversing multiple bottlenecks.

5.2. Summary and Discussion of Termination Results

The simulations results presented in this installment of the simulation study further demonstrated that at least in a simple one-bottleneck topology case the termination mechanism works reasonably well for a wide range of parameters for all traffic models we considered. The key thrust of this study was the investigation of how much ingress-egress aggregation is needed for tolerable performance of the algorithm (assuming sufficient degree of bottleneck aggregation). We demonstrated that contrary to our expectations, it was not easy to find cases with sufficiently bad performance. We traced some of this better-than-expected performance to the effect of synchronization of the token bucket state for certain combinations of parameter values, and we demonstrated this effect cannot be simply regarded as a simulation artifact. A question of whether this synchronization can be explored to the benefit of the general operation for voice-only PCN regions remains open, but seems of substantial interest. Further investigation with other codecs and in a broader set of network conditions is warranted to address this question. Our experiments demonstrated that the absolute value of RTT of the flows sharing the same bottleneck did not have any appreciable effect as long as the RTT of all flows were the same (or close). However, we have demonstrated that if RTTs of different flows are substantially different, longer RTT flows tend to over-terminate, resulting in overall over-termination as well. In the multi-bottleneck case, the "beatdown" of long-haul discussed in the context of admission, cause a certain degree of over-termination. In addition, unlike the case of admission when under-admission of long-haul aggregates was compensated by the over-admission of the short-haul aggregates keeping the bottlenecks utilized, in the case of termination, any over-termination of long-haul aggregates is likely to result in under-utilization of some bottleneck links.
rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights. This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Intellectual Property The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79. Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr. The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org. Acknowledgment Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA). Zhang, et al. Expires January 6, 2008
6.1. Restrictions on Termination-to-admission Thresholds

An obvious restriction necessary for the single-marking approach is that the ratio of (implicit) Preemption and admission thresholds remains the same on all links in the PCN region. While clearly a limitation, this does not appear to be particularly crippling, and does not appear to outweigh the benefits of reducing the overhead in the router implementation and savings in codepoints in the case of a single PCN domain, or in the case of multiple concatenated PCN regions. The case when this limitation becomes more inconvenient is when an operator wants to merge two previously separate PCN regions (which may have different admission-to-preemption ratios) into a single PCN region. In this case it becomes necessary to do a network-wide reconfiguration to align the settings.

The fixed ratio between the implicit termination rate and the configured-admissible-rate also has an implications on traffic
engineering considerations. Those are discussed in section 7.7 below.

6.2. Assumptions on Loss

Just as in the case of [I-D.briscoe-tsvwg-cl-architecture], the approach presented in this draft assumes that the configured-admissible-rate is configured at each link below the service rate of the traffic using PCN. This assumption is significant because the algorithm relies on the fact that if admission threshold is exceeded, enough marked traffic reaches the PCN-egress-node to reach the configured CLE level. If this condition does not hold, then traffic may get dropped without ever triggering admission decision.

6.3. Effect of Reaction Timescale of Admission Mechanism

As mentioned earlier in this draft, there is a potential concern that slower reaction time of admissions mechanism presented in this draft compared to [I-D.briscoe-tsvwg-cl-architecture] may result in overshoot when the load grows rapidly, and undershoot when the load drops rapidly. While this is a valid concern theoretically, it should be noted that at least for the traffic and parameters used in the simulation study reported here, there was no indication that this was a problem.

6.4. Performance Implications and Tradeoffs

Replacement of a relatively well-studied queue-based measurement-based admission control approach by a cruder excess-rate measurement technique raises a number of algorithmic and performance concerns that need to be carefully evaluated. For example, a token-bucket excess rate measurement is expected to be substantially more sensitive to traffic burstiness and parameter setting, which may have a significant effect in the case of lower levels of traffic aggregation, especially for variable-rate traffic such as video. In addition, the appropriate timescale of rate measurement needs to be carefully evaluated, and in general it depends on the degree of expected traffic variability which is frequently unknown.

In view of that, an initial performance comparison of the token-bucket based measurement is presented in the following section. Within the constraints of this study, the performance tradeoffs observed between the queue-based technique suggested in [I-D.briscoe-tsvwg-cl-architecture] and a simpler token-bucket-based excess rate measurement do not appear to be a cause of substantial concern for cases when traffic aggregation is reasonably high at the bottleneck links as well as on a per ingress-egress pair basis. Details of the simulation study, as well as additional discussion of
its implications are presented in section 7.

Also, one mitigating consideration in favor of the simpler mechanism is that in a typical DiffServ environment, the real-time traffic is expected to be served at a higher priority and/or the target admission rate is expected to be substantially below the speed at which the real-time queue is actually served. If these assumptions hold, then there is some margin of safety for an admission control algorithm, making the requirements for admission control more forgiving to bounded errors — see additional discussion in section 7.

6.5. Effect on Proposed Anti-Cheating Mechanisms

Replacement of the queue-based admission control mechanism of [I-D.briscoe-tsvwg-cl-architecture] by an excess-rate based admission marking changing the semantics of the pre-congestion marking, and consequently interferes with mechanisms for cheating detection discussed in [I-D.briscoe-tsvwg-re-ecn-border-cheat]. Implications of excess-rate based marking on the anti-cheating mechanisms need to be considered.

6.6. ECMP Handling

An issue not directly addressed by neither the dual-marking approach described in [I-D.briscoe-tsvwg-cl-architecture] nor the single-marking approach described in this draft is that if ECMP is enabled in the PCN-domain, then the PCN-edge nodes do not have a way of knowing whether specific flows in the ingress-egress aggregate followed the same path or not. If multiple paths are followed, then some of those paths may be experiencing pre-congestion marking, and some are not. Hence, for example, an ingress node may choose to terminate a flow which takes an entirely un-congested path. This will not only unnecessarily terminate some flows, but also will not eliminate congestion on the actually congested path. While eventually, after several iterations, the correct number of flows might be terminated on the congestion path, this is clearly suboptimal, as the termination takes longer, and many flows are potentially terminated unnecessarily.

Two approaches for solving this problem were proposed in [I-D.babiarz-pcn-3sm], [I-D.westberg-pcn-load-control]. The former handles ECMP by terminating those flows that are termination-marked as soon as the termination marking is seen. The latter uses an additional DiffServ marking/codepoint to mark all packets of the flows passing through a congestion point, with the PCN-boundary-nodes terminating only those flows which are marked with this additional mark. Both of these approaches also differ in the termination-marking semantics, but we omit the discussion of these differences as
they can be considered largely independent of the ECMP issue.

It should be noted that although not proposed in this draft, either of these ideas can be used with dual- and single-marking approaches discussed here. Specifically, and when a PCN- ingress-node decides which flows to terminate, it can choose for termination only those flows that are termination-marked. Likewise, at the cost of an additional (DiffServ) codepoint, a PCN-internal-node can mark all packets of all flows using this additional marking, and then the PCN-boundary-nodes can use this additional marking to guide their flow termination decisions.

Either of these approaches appears to imply changes to the PCN architecture as proposed in [I-D.eardley-pcn-architecture]. Such changes have not been considered in this draft at this point.

6.7. Traffic Engineering Considerations

Dual-marking PCN can be viewed as a replacement for Resilient Network Provisioning (RNP). It is reasonable to expect that an operator currently using DiffServ provisioning for real-time traffic might consider a move to PCN. For such a move it is necessary to understand how to set the PCN rate thresholds to make sure that the move to PCN does not detrimentally affect the guarantees currently offered to the operator.

The key question addressed in this section is how to set PCN admission and termination thresholds in the dual marking approach or the single admission threshold and the scaling factor U reflecting the implicit termination threshold in the single-marking approach so that the result is "not worse" than provisioning in the amount of traffic that can be admitted. Even more specifically we will address what if any are the tradeoffs between the dual-marking and the single-approach arise when answering this question. This question was first raised in [Menth] and is further addressed below.

Typically, RNP would size the network (in this specific case traffic that is expected to use PCN) by making sure that capacity available for this (PCN) type of traffic is sufficient for PCN traffic under "normal" circumstances (that is, under no failure condition, for a given traffic matrix), and under a specific set of single failure scenarios (e.g. failure of each individual single link). Some of the obvious limitations of such provisioning is that

- the traffic matrix is often not known well, and at times, especially during flash-crowds, the actual traffic matrix can differ substantially from the one assumed by provisioning
unpredicted, non-planned failures can occur (e.g. multiple links, nodes, etc), causing overload.

It is specifically such unplanned cases that serve as the motivation for PCN. Yet, one may want to make sure that for cases that RNP can (and does today) plan for, PCN does no worse when an operator makes the decision to implement PCN on a currently provisioned network. This question directly relates to the choice of the PCN configured admission and termination thresholds.

For the dual-marking approach, where the termination and admission thresholds are set independently on any link, one can address this issue as follows [Menth]. If a provisioning tool is available, for a given traffic matrix, one can determine the utilisation of any link used by traffic expected to use PCN under the no-failure condition, and simply set the configured-admissible-rate to that "no-failure utilization". Then a network using PCN will be able to admit as much traffic as the RNP, and will reject any traffic that exceeds the expected traffic matrix. To address resiliency against a set of planned failures, one can use RNP to find the worst-case utilization of any link under the set of all provisioned failures, and then set the configured-termination-rate to that worst case utilisation.

Clearly, such setting of PCN thresholds with the dual-marking approach will achieve the following goals:

- PCN will admit the same traffic matrix as used by RNP and will protect it against all planned failures without terminating any traffic
- When traffic deviates from the planned traffic matrix, PCN will admit such traffic as long as the total usage of any link (without failure) does not exceed the configured-admission threshold, and all admitted traffic will be protected against all planned failures
- Additional traffic will not be admitted under the no-failure conditions, and traffic exceeding configure-termination threshold during non-planned failures will be terminated.
- Under non-planned failures, some of the planned traffic matrix may be terminated, but the remaining traffic will be able to receive its QoS treatment.

The above argues that an operator moving from a purely provisioned network to a PCN network can find the settings of the PCN threshold with dual marking in such a way that all admitted traffic is protected against all planned failures.
It is easy to see that with the single-marking scheme, the above approach does not work directly [Menth]. Indeed, the ratio between the configured-termination thresholds and the configured-admissible-rate may not be constant on all links. Since the single-marking approach requires the (implicit) termination rate to be within a fixed factor of the configured admission rate, it can be argued (as was argued in [Menth].) that one needs to set the system-wide ratio $U$ between the (implicit) termination threshold and the configured admission threshold to correspond to the largest ratio between the worst case resilient utilization and the no-failure utilization of RNP, and set the admission threshold on each link to the worst case resilient utilization divided by that system wide ratio. Such approach would result in lower admission thresholds on some links than that of the dual-marking setting of the admission threshold proposed above. It can therefore be argued that PCN with single marking will be able to admit *less* traffic that can be fully protected under the planned set of failures than both RNP and the dual-marking approach.

However, the settings of the single-marking threshold proposed above are not the only one possible, and in fact we propose here that the settings are chosen differently. Such different settings (described below) will result in the following properties of the PCN network:

- PCN will admit the same traffic matrix as used by RNP *or more*
- The traffic matrix assumed by RNP will be fully protected against all planned failures without terminating any admitted traffic
- When traffic deviates from the planned traffic matrix, PCN will admit such traffic as long as the total usage of any link (without failure) does not exceed the configured-admission threshold. However, not all admitted traffic will be protected against all planned failures (i.e. even under planned failures, traffic exceeding the planned traffic matrix may be preempted)
- Under non-planned failures, some of the planned traffic matrix may be terminated, but the remaining traffic will be able to receive its QoS treatment.

It is easy to see that all of these properties can be achieved if instead of using the largest ratio between worst case resilient utilisation to the no-failure utilisation of RNP across all links for setting the system wide constant $U$ in the single-marking approach as proposed in [Menth], one uses the *smallest* ratio, and set the configured-admissible-rate to the worst case resilient utilization divided by that ratio. With such setting, the configured-admissions threshold on each link is at least as large as the non-failure RNP.
utilisation (and hence the planned traffic matrix is always admitted), and the implicit termination threshold is at the worst case planned resilient utilisation of RNP on each link (and hence the planned traffic matrix will be fully protected against the planned failures). Therefore, with such settings, the single-marking draft does as well as RNP or dual-marking with respect to the planned matrix and planned failures. In fact, unlike the dual marking approach, it can admit more traffic on some links than the planned traffic matrix would allow, but it is only guaranteed to protect up to the planned traffic matrix under planned failures.

In summary, we have argued that both the single-marking approach and the dual-marking approach can be configured to ensure that PCN "does no worse" than RNP for the planned matrix and the planned failure conditions, (and both can do better than RNP under non-planned conditions). The tradeoff between the two is that although the planned traffic matrix can be admitted with protection guarantees against planned failures with both approaches, the nature of the guarantee for the admitted traffic is different. Dual marking (with the settings proposed) would protect all admitted traffic but would not admit more than planned), while single marking (with the settings proposed) will admit more traffic than planned, but will not guarantee protection against planned failures for traffic exceeding planned utilization.

7. Performance Evaluation Comparison

7.1. Relationship to other drafts

Initial simulation results of admission and termination mechanisms of [I-D.briscoe-tswg-cl-architecture] were reported in [I-D.briscoe-tswg-cl-phb]. A follow-up study of these mechanisms is presented in a companion draft draft-zhang-cl-performance-evaluation-02.txt. The current draft concentrates on a performance comparison of the admission control mechanism of [I-D.briscoe-tswg-cl-phb] and the token-bucket-based admission control described in section 2 of this draft.

7.2. Limitations, Conclusions and Direction for Future Work

Due to time constraints, the study performed so far was limited to a small set of topologies, described in the Appendix. The key questions that have been investigated are the comparative sensitivity of the two schemes to parameter settings and the effect of traffic burstiness and of the degree of aggregation on a per ingress-egress pair on the performance of the admission control algorithms under study. The study is limited to the case where there is no packet
loss. While this is a reasonable initial assumption for an admission control algorithm that is supposed to maintain the traffic level significantly below the service capacity of the corresponding queue, nevertheless future study is necessary to evaluate the effect of packet loss.

7.2.1. High Level Conclusions

The results of this (preliminary) study indicate that there is a potential that a reasonable complexity/performance tradeoff may be viable for the choice of admission control algorithm. In turn, this suggests that using a single codepoint and metering technique for admission and Preemption may be a viable option.

The key high-level conclusions of the simulation study comparing the performance of queue-based and token-based admission control algorithms are summarized below:

1. At reasonable level of aggregation at the bottleneck and per ingress-egress pair traffic, both algorithms perform reasonably well for the range of traffic models considered (see section 4.3 for detail).

2. Both schemes are stressed for small levels of ingress-egress pair aggregation levels of bursty traffic (e.g. a single video-like bursty SVD flow per ingress-egress pair). However, while the queue-based scheme results in tolerable performance even at low levels of per ingress-egress aggregation, the token-bucket-based scheme is substantially more sensitive to parameter setting than the queue-based scheme, and its performance for the high rate bursty SVD traffic with low levels of ingress-egress aggregation is quite poor unless parameters are chosen carefully to curb the error. It should be noted that the SVD traffic model used in this study is expected to be substantially more challenging for both admission and Preemption mechanisms that the actual video traffic, as the latter is expected to be much smoother than the bursty on-off model with high peak-to-mean ratio we used. This expectation is confirmed by the fact that simulations with actual video traces reported in this version of the draft reveal that the performance of the video traces is much closer to that of VBR voice than of our crude SVD on-off model.

3. Even for small per ingress-egress pair aggregation, reasonable performance across a range of traffic models can be obtained for both algorithms (with a narrower range of parameter setting for the token-bucket based approach). However, at very low ingress-egress aggregation, the token bucket scheme is substantially more sensitive to parameter variations than the virtual-queue scheme.
In general, the token-bucket scheme performance is quite brittle at very low aggregations, and displays substantial performance degradation with BATCH traffic, as well synchronization effects resulting in substantial over-admission (see section 9.5.2).

4. The absolute value of round-trip time (RTT) or the RTT difference between different ingress-egress pair within the range of continental propagation delays does not appear to have a visible effect on the performance of both algorithms.

5. There is no substantial effect on the bottleneck utilisation of multi-bottleneck topologies for both schemes. Both schemes suffer substantial unfairness (and possibly complete starvation) of the long-haul aggregates traversing multiple bottlenecks compared to short-haul flows (a property shared by other MBAC algorithms as well). Token-bucket scheme displayed somewhat larger unfairness than the virtual-queue scheme.

6.

7.2.2. Future work

This study is but the first step in performance evaluation of the token-bucket based admission control. Further evaluation should include a range of investigation, including the following

- interactions between admission control and preemption
- effect of signaling delays/probing
- effect of loss of marked packets

8. Appendix A: Simulation Details

8.1. Network and Signaling Models

Network topologies used in this study are shown in the Figures below. The network is modeled as either Single Link (Fig. A.1), Multi Link Network with a single bottleneck (termed "RTT", Fig. A.2), or a range of multi-bottleneck topologies shown in Fig. A.3 (termed "Parking Lot").
Figure A.1: Simulated Single Link Network.

A --- B

Figure A.2: Simulated Multi Link Network.

A
\ / 
B - D - F
/ 
C

Figure A.3: Simulated Multiple-bottleneck (Parking Lot) Topologies.

A--B--C  A--B--C--D  A--B--C--D--E--F
| | | | | | | | | | | | | | |
D E F E F G H G H I J K L

(a) (b) (c)

Figure A.1 shows a single link between an ingress and an egress node, all flows enter at node A and depart at node B. This topology is used for the basic verification of the behavior of the algorithms with respect to a single ingress-egress aggregate in isolation.

In Figure A.2, a set of ingresses (A, B, C) are connected to an interior node in the network (D). This topology is used to study the behavior of the algorithm where many ingress-egress aggregates share a single bottleneck link. The number of ingresses varied in different simulation experiments in the range of 2-100. All links have generally different propagation delays, in the range 1ms - 100 ms (although in some experiments all propagation delays are set the same. This node D in turn is connected to the egress (F). In this topology, different sets of flows between each ingress and the egress converge on the single link D-F, where pre-congestion notification algorithm is enabled. The capacities of the ingress links are not limiting, and hence no PCN is enable on those. The bottleneck link D-F is modeled with a 10ms propagation delay in all simulations. Therefore the range of round-trip delays in the experiments is from 22ms to 220ms.

Another type of network of interest is multi-bottleneck (or Parking Lot, PLT for short) topology. The simplest PLT with 2 bottlenecks is
illustrated in Fig A.3(a). An example traffic matrix with this network on this topology is as follows:

- an aggregate of "2-hop" flows entering the network at A and leaving at C (via the two links A-B-C)
- an aggregate of "1-hop" flows entering the network at D and leaving at E (via A-B)
- an aggregate of "1-hop" flows entering the network at E and leaving at F (via B-C)

In the 2-hop PLT shown in Fig. A.3(a) the points of congestion are links A--B and B--C. Capacity of all other links is not limiting. We also experiment with larger PLT topologies with 3 bottlenecks (see Fig A.3(b)) and 5 bottlenecks (Fig A.3(c)). In all cases, we simulated one ingress-egress pair that carries the aggregate of "long" flows traversing all the N bottlenecks (where N is the number of bottleneck links in the PLT topology), and N ingress-egress pairs that carry flows traversing a single bottleneck link and exiting at the next "hop". In all cases, only the "horizontal" links in Fig. A.3 were the bottlenecks, with capacities of all "vertical" links non-limiting. Propagation delays for all links in all PLT topologies are set to 1ms.

Due to time limitations, other possible traffic matrices (e.g. some of the flows traversing a subset of several bottleneck links) have not yet been considered and remain the area for future investigation.

Our simulations concentrated primarily on the range of capacities of 'bottleneck' links with sufficient aggregation - above 10 Mbps for voice and 622 Mbps for SVD, up to 2.4 Gbps. But we also investigated slower 'bottleneck' links down to 512 Kbps in some experiments. Higher rate bottleneck speeds were not considered due to the simulation time limitations. It should generally be expected that the higher link speeds will result in higher levels of aggregation, and hence generally better performance of the measurement-based algorithms. Therefore it seems reasonable to believe that the link speeds studied do provide meaningful evaluation targets.

In the simulation model, a call requests arrives at the ingress and immediately sends a message to the egress. The message arrives at the egress after the propagation time plus link processing time (but no queuing delay). When the egress receives this message, it immediately responds to the ingress with the current Congestion-Level-Estimate. If the Congestion-Level-Estimate is below the specified CLE-threshold, the call is admitted, otherwise it is rejected. An admitted call sends packets according to one of the
chosen traffic models for the duration of the call (see next section). Propagation delay from source to the ingress and from destination to the egress is assumed negligible and is not modeled.

In the simulation model of admission control, a call request arrives at the ingress and immediately sends a message to the egress. The message arrives at the egress after the propagation time plus link processing time (but no queuing delay). When the egress receives this message, it immediately responds to the ingress with the current Congestion Level Estimate. If the Congestion Level Estimate is below the specified CLE- threshold, the call is admitted, otherwise it is rejected. For Flow Termination, once the ingress node of a PCN-domain decides to terminate a flow, that flow is preempted immediately and sends no more packets from that time on. The life of a flow outside the domain described above is not modelled.

Propagation delay from source to the ingress and from destination to the egress is assumed negligible and is not modelled.

8.2. Traffic Models

Four types of traffic were simulated (CBR voice, on-off traffic approximating voice with silence compression, and on-off traffic with higher peak and mean rates (we termed the latter "Synthetic Video" (SVD) as the chosen peak and mean rate was similar to that of an MPEG video stream. (but for SVD no attempt was made to match any other parameters of this traffic to those of a video stream), and finally real video traces from http://www.tkn.tu-berlin.de/research/trace/trace.html (courtesy Telecommunication Networks Group of Technical University of Berlin).

The distribution of flow duration was chosen to be exponentially distributed with mean 1min, regardless of the traffic type. In most of the experiments flows arrived according to a Poisson distribution with mean arrival rate chosen to achieve a desired amount of overload over the configured-admissible-rate in each experiment. Overloads in the range 1x to 5x and underload with 0.95x have been investigated. Note that the rationale for looking at the load and below is to see if any significant amount of "false rejects" would be seen (i.e. one would assume that all traffic should be accepted if the total demand is below the admission threshold). For on-off traffic, on and off periods were exponentially distributed with the specified mean. Traffic parameters for each type are summarized below:

8.2.1. Voice Traffic Models

Table A.1 below describes all voice codecs we modeled in our simulation results.
The first two rows correspond to our two basic models corresponding to the older G.711 encoding with and without silence compression. These two models are referred simply as "CBR" and "VBR" in the reported simulation results.

We also simulated several "mixes" of the different codecs reported in the table below. The primary mix consists of equal proportion of all voice codecs listed below. We have also simulated various other mix consist different proportion of the subset of all codecs. Though these result are not reported in this draft due to their similarities to the primary mix result.

<table>
<thead>
<tr>
<th>Name/Codecs</th>
<th>Packet Size (Bytes)</th>
<th>Inter-Arrival Time (ms)</th>
<th>On/Off Period Ratio</th>
<th>Average Rate (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;CBR&quot;</td>
<td>160</td>
<td>20</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>&quot;VBR&quot;</td>
<td>160</td>
<td>20</td>
<td>0.34</td>
<td>21.75</td>
</tr>
<tr>
<td>G.711 CBR</td>
<td>200</td>
<td>20</td>
<td>1</td>
<td>80</td>
</tr>
<tr>
<td>G.711 VBR</td>
<td>200</td>
<td>20</td>
<td>0.4</td>
<td>32</td>
</tr>
<tr>
<td>G.711 CBR</td>
<td>120</td>
<td>10</td>
<td>1</td>
<td>96</td>
</tr>
<tr>
<td>G.711 VBR</td>
<td>120</td>
<td>10</td>
<td>0.4</td>
<td>38.4</td>
</tr>
<tr>
<td>G.729 CBR</td>
<td>60</td>
<td>20</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>G.729 VBR</td>
<td>60</td>
<td>20</td>
<td>0.4</td>
<td>9.6</td>
</tr>
</tbody>
</table>

Table A.1 Simulated Voice Codices.

8.2.2. "Synthetic Video": High Rate ON-OFF traffic with Video-like Mean and Peak Rates ("SVD")

This model is on-off traffic with video-like mean-to-peak ratio and mean rate approximating that of an MPEG-2 video stream. No attempt is made to simulate any other aspects of a real video stream, and this model is merely that of on-off traffic. Although there is no claim that this model represents the performance of video traffic under the algorithms in question adequately, intuitively, this model should be more challenging for a measurement-based algorithm than the actual MPEG video, and as a result, ‘good’ or "reasonable" performance on this traffic model indicates that MPEG traffic should perform at least as well. We term this type of traffic SVD for "Synthetic Video".
o Long term average rate 4 Mbps

o On Period mean duration 340ms; during the on-period the packets are sent at 12 Mbps (1500 byte packets, packet inter-arrival: 1ms)

o Off Period mean duration 660ms

8.2.3. Real Video Traces (VTR)

We used a publicly available library of frame size traces of long MPEG-4 and H.263 encoded video obtained from http://www.tkn.tu-berlin.de/research/trace/trace.html. Each trace in that repository is roughly 60 minutes in length, consisting of a list of records in the format of <FrameArrivalTime, FrameSize>. Among the 160 available traces, we picked the two with the highest average rate (averaged over the trace length, in this case, 60 minutes. In addition, the two also have a similar average rate). The trace file used in the simulation is the concatenation of the two.

Since the duration of the flow in our simulation is much smaller than the length of the trace, we checked whether the expected rate of flow corresponds to the trace’s long term average. To do so, we simulated a number of flows starting from random locations in the trace with duration chosen to be exponentially distributed with the mean of 1min. The results show that the expected rate of flow is roughly the same as the trace’s average.

In summary, our simulations use a set of segments of the 120 min trace chosen at random offset from the beginning and with mean duration of 1 min.

Since the traces provide only the frame size, we also simulated packetization of the frame as a CBR segment with packet size and inter-arrival time corresponding to those of our SVD model. Since the frame size is not always a multiple of the chosen packet size, the last packet in a frame may be shorter than 1500 bytes chosen for the SVD encoding.

Traffic characteristics for our VTR models are summarized below:

o Average rate 769 Kbps

o Each frame is sent with packet length 1500 bytes and packet inter-arrival time 1ms

o No traffic is sent between frames.
8.3. Randomization of Base Traffic Models

To emulate some degree of disruption of the arrival models we used by the queuing encountered by the traffic stream before its arrival to the bottleneck link (e.g. prior to its arrival in the PCN-domain), we implemented limited randomization of the base models by randomly moving the packet by a small amount of time around its transmission time in the corresponding base traffic model. More specifically, for each packet we chose a random number picked from uniform distribution of interval [0, p], and delayed the packet by p times the relevant CBR packet inter-arrival time compared to its ideal CBR departure time.

To simulate a range of queueing delays, we varied the degree of randomization in different experiments by choosing p from 0.0001 to 0.1. While we do not assume that this is necessarily an adequate model for network-introduced jitter, we chose it for the simplicity of implementation as a means to eliminate any simulation artifacts of strictly CBR traffic generation.

We implemented randomized versions of all 4 traffic streams (CBR, VBR, SVD and VTR) by randomizing the CBR portion of each model.

8.4. Parameter Settings

8.4.1. Queue-based settings

All the queue-based simulations were run with the following Virtual Queue thresholds:

- virtual-queue-rate: configured-admissible-rate, 1/2 link speed
- min-marking-threshold: 5ms at virtual-queue-rate
- max-marking-threshold: 15ms at virtual-queue-rate
- virtual-queue-upper-limit: 20ms at virtual-queue-rate

At the egress, the CLE is computed as an exponential weighted moving average (EWMA) on an interval basis, with 100ms measurement interval chosen in all simulations. We simulated the EWMA weight ranging 0.1 to 0.9. The CLE threshold is chosen to be 0.05, 0.15, 0.25, and 0.5.

8.4.2. Token Bucket Settings

The token bucket rate is set to the configured-admissible-rate, which is half of the link speed in all experiments. Token bucket depth ranges from 64 to 512 packets. Our simulation results indicate that
depth of token bucket has no significant impact on the performance of the algorithms and hence, in the rest of the section, we only present the result with 256 packets bucket depth.

The CLE is calculated using EWMA just as in the case of virtual-queue settings, with weights from 0.1 to 0.9. The CLE thresholds are chosen to be 0.0001, 0.001, 0.01, 0.05 in this case. Note that the since meaning of the CLE is different for the Token bucket and queue-based algorithms, so there is no direct correspondence between the choice of the CLE thresholds in the two cases.

8.5. Simulation Details

To evaluate the performance of the algorithms, we recorded the actual admitted load at a granularity of 50ms, from which the mean admitted load over the duration of the simulation run can be computed. We verified that the actual admitted load at any time does not deviate much from the mean admitted load in each experiment by computing the coefficient of variation (CV is consistently 0.07 for CBR, 0.15 for VBR, 0.17 for VTR and 0.51 for SVD for all experiments). Finally, the performance of the algorithms is evaluated using a metric called over-admission-percentage, which is calculated as a percentage difference between the mean admitted load (with the mean taken over the duration of the experiment) and the configured admission rate. Given reasonably small deviation of the admitted rate from the mean admitted in the experiments, this seems reasonable.

8.5.1. Sensitivity to EWMA weight and CLE

Table A.2 summarized the comparison result of over-admission-percentage values from 15 experiments with different [weight, CLE threshold] settings for each type of traffic and each topology. The Ratio of the demand on the bottleneck link to the configured admission threshold is set to 5x. (In the results for 0.95x can be found in previous draft). For parking lot topologies we report the worst case result across all bottlenecks. We present here only the extreme value over the range of resulting over-admission-percentage values.

We found that the virtual-queue admission control algorithm works reliably with the range of parameters we simulated, for all five types of traffic. In addition, except for SVD, the performance is insensitive to the parameters change under all tested topologies. For SVD, the algorithms does show certain sensitivity to the tested parameters. The high level conclusion that can be drawn is that (predictably) high peak-to-mean ratio SVD traffic is substantially more stressful to the queue-based admission control algorithm, but a set of parameters exists that keeps the over-admission within about
-4% - +7% of the expected load even for the bursty SVD traffic.

The token bucket-based admission control algorithm shows higher sensitivity to the parameter settings compared to the virtual queue based algorithm. It is important to note here that for the token bucket-based admission control no traffic will be marked until the rate of traffic exceeds the configured admission rate by the chosen CLE. As a consequence, even with the ideal performance of the algorithms, the over-admission-percentage will not be 0, rather it is expected to equal to CLE threshold if the algorithm performs as expected. Therefore, a more meaningful metric for the token-based results is actually the over-admission-percentage (listed below) minus the corresponding (CLE threshold * 100). For example, for CLE = 0.01, one would expect that 1% over-admission is inherently embedded in the algorithm, with the algorithm by design reacting to 0.5% overload (or more) only. Hence, with CLE = 0.01 a 10% over-admission in the token-bucket case should be compared to a 1% over-admission in the queue-based algorithm. When comparing the performance of token bucket (with the adjusted over-admission-percentage) to its corresponding virtual queue result, we found that token bucket performs only slightly worse for voice-like CBR VBR, and MIX traffic.

The results for SVD traffic require some additional commentary. Note from the results in Table A.2. in the Single Link topology the performance of the token-based solution is comparable to the performance of the queue-based scheme. However, for the RTT topology, the worse case performance for SVD traffic becomes very bad, with up to 23% over-admission in a high overload. We investigated two potential causes of this drastic degradation of performance by concentrating on two key differences between the Single Link and the RTT topologies: the difference in the round-trip times and the degree of aggregation in a per ingress-egress pair aggregate.

To investigate the effect of the difference in round-trip times, we also conducted a subset of the experiments described above using the RTT topology that has the same RTT across all ingress-egress pairs rather than the range of RTTs in one experiment. We found out that neither the absolute nor the relative difference in RTT between different ingress-egress pairs appear to have any visible effect on the over-load performance or the fairness of both algorithms (we do not present these results here as their are essentially identical to those in Table A.2). In view of that and noting that in the RTT topology we used for these experiments for the SVD traffic, there is only 1 highly bursty flow per ingress, we believe that the severe degradation of performance in this topology is directly attributable to the lack of traffic aggregation on the ingress-egress pair basis.
We also note that even for this highly challenging scenario, it is possible to find a range of parameters that limit the over-admission case for SVD traffic to quite a reasonable range of -3% + 10% (adjusted by the CLE). Luckily, these are the same parameter settings that work quite well for the other types of traffic tested.

<table>
<thead>
<tr>
<th>Type</th>
<th>Topo</th>
<th>Over Admission Perc Stats</th>
<th>Queue-based</th>
<th>Bucket-Based</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>---------------------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>S.Link</td>
<td>RTT</td>
<td>0.224</td>
<td>1.105</td>
<td>-0.99</td>
</tr>
<tr>
<td>CBR</td>
<td>RTT</td>
<td>0.200</td>
<td>1.192</td>
<td>6.495</td>
</tr>
<tr>
<td></td>
<td>PLT</td>
<td>-0.93</td>
<td>0.990</td>
<td>-2.24</td>
</tr>
<tr>
<td>VBR</td>
<td>RTT</td>
<td>-0.07</td>
<td>1.646</td>
<td>-2.94</td>
</tr>
<tr>
<td></td>
<td>PLT</td>
<td>-0.11</td>
<td>1.830</td>
<td>-1.92</td>
</tr>
<tr>
<td>MIX</td>
<td>RTT</td>
<td>-0.14</td>
<td>1.961</td>
<td>-2.85</td>
</tr>
<tr>
<td></td>
<td>PLT</td>
<td>-0.46</td>
<td>1.803</td>
<td>-3.18</td>
</tr>
<tr>
<td>VTR</td>
<td>RTT</td>
<td>-0.05</td>
<td>1.581</td>
<td>-2.36</td>
</tr>
<tr>
<td></td>
<td>PLT</td>
<td>-0.57</td>
<td>1.313</td>
<td>-1.44</td>
</tr>
<tr>
<td>SVD</td>
<td>RTT</td>
<td>-2.73</td>
<td>6.525</td>
<td>-11.25</td>
</tr>
<tr>
<td></td>
<td>PLT</td>
<td>-4.84</td>
<td>4.294</td>
<td>-11.40</td>
</tr>
</tbody>
</table>

Table A.2 Parameter sensitivity: Queue-based v.s. Token Bucket-based. For the single bottleneck topologies (S. Link and RTT) the overload column represents the ratio of the mean demand on the bottleneck link to the configured admission threshold. For parking lot topologies we report the worst case result across all bottlenecks. We present here only the worst case value over the range of resulting over-admission-percentage values.

Effect of Ingress-Egress Aggregation

To investigate the effect of Ingress-Egress Aggregation, we fix a particular EWMA weight and CLE setting (in this case, weight=0.3, for virtual queue scheme CLE=0.05, and for the token bucket scheme CLE=0.0001), vary the level of ingress-egress aggregation by using RTT topologies with different number of ingresses.
Table A.3 shows the change of over-admission-percentage with respect to the increase in the number of ingress for both virtual queue and token bucket. For all traffic, the leftmost column in the represents the case with the largest aggregation (only two ingresses), while the right most column represents the lowest level of aggregation (expected number calls per ingress is just 1 in this case). In all experiments the aggregate load on the bottleneck is the same across each traffic type (with the aggregate load being evenly divided between all ingresses).

As seen from Table A.3, the virtual queue based approach is relatively insensitive to the level of ingress-egress aggregation. On the other hand, the Token Bucket based approach is performing significantly worse at lower levels of ingress-egress aggregation. For example for CBR (with expect 1-call per ingress), the over-admission-percentage can be as bad as 45%.
Our investigation reveals that the cause of the poor performance of the token bucket scheme in our experiments is attributed directly to the same "synchronisation" effect as was earlier described in the Termination (preemption) results in draft-zhang-pcn-performance-evaluation, and to which we refer the reader for a more detailed description of this effect. In short however, for CBR traffic, a periodic pattern arises where packets of a given flow see roughly the same state of the token bucket at the
bottleneck, and hence either all get marked, or all do not get marked. As a result, at low levels of aggregation a subset of ingresses always get their packets marked, while some other ingresses do not.

As reported in draft-zhang-pcn-performance-evaluation, in the case of Termination this synchronization effect is beneficial to the algorithm. In contrast, for Admission, this synchronization is detrimental to the algorithm performance at low aggregations. This can be easily explained by noting that ingresses which packets do not get marked continue admitting new traffic even if the aggregate bottleneck load has been reached or exceeded.

Since most of the other traffic patterns contain large CBR segments, this effect is seen with other traffic types as well, although to a different extent.

A natural initial reaction can be to write-off this effect as purely a simulation artifact. In fact, one can expect that if some jitter is introduced into the strict CBR traffic pattern so that the packet transmission is longer strictly periodic, then the "synchronization" effect might be easily broken.

To verify whether this is indeed the case, we ran the experiment with same topologies and parameter settings, but with randomized version of the base traffic types. As described earlier, our randomized traffic types simulate some amount of network-introduced jitter by offsetting every packet’s transmission time by a random amount chosen uniformly in the "deviation interval" (where the "deviation interval" is calculated as a percentage of the original, non-randomized inter-packet-arrival-time for the given traffic type). The larger the deviation interval, the larger the expected jitter is.

The results are summarized in Table A.4 (note that the column of "No-Rand" actually correspond to the token bucket results in Table A.3). It turns out that indeed introducing enough jitter does break the synchronization effect and the performance of the algorithm much improves. However, it takes sufficient amount of the randomization before it is noticed (in our simulations one would need to introduce about 1ms jitter to our CBR traffic before the synchronisation effect is broken. While 1 ms per-hop jitter for voice traffic is not unreasonable to expect for voice traffic, in well provisioned networks with a relatively small amount of voice traffic in the priority queue one might in fact find lower jitter levels. In any case, the fact that jitter smaller than 1ms does not substantially help the performance indicates the "synchronization" effect can not be completely written off as a simulation artifact.
The good news, however, that this effect is visible only at very low ingress-egress aggregation levels, and as the ingress-egress aggregation increases, the effect quickly disappears.

We observed the synchronisation effect consistently across all types of traffic we tested with the exception of VTR. VTR also exhibits some aggregation effect - however randomization of its CBR portion has almost have no effect on performance. We suspect this is because the randomization we perform is at packet level, while the synchronization that seems to be causing the performance degradation at low ingress-egress aggregation for VTR traffic occurs at frame-level. Although our investigation of this issue is not completed yet, our preliminary results show that if we calculating random deviation for our artificially induced jitter using frame inter-arrival time instead of packet-interarrival time, we can reduce the over-admission percentage for VTR to roughly 3%. It is unclear however, whether such randomisation at the frame level meaningfully reflects network-introduced jitter.
### Table A.4 Ingress-Egress Aggregation: Token-based results for Randomized traffic

<table>
<thead>
<tr>
<th></th>
<th>0.0001</th>
<th>0.001</th>
<th>0.005</th>
<th>0.01</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ingr No-Rand</strong></td>
<td><strong>Deviation Interval</strong></td>
<td><strong>0.0001</strong></td>
<td><strong>0.001</strong></td>
<td><strong>0.005</strong></td>
<td><strong>0.01</strong></td>
</tr>
<tr>
<td>2</td>
<td>0.725</td>
<td>0.683</td>
<td>0.784</td>
<td>0.725</td>
<td>0.772</td>
</tr>
<tr>
<td>10</td>
<td>0.753</td>
<td>0.725</td>
<td>0.543</td>
<td>0.645</td>
<td>0.733</td>
</tr>
<tr>
<td>100</td>
<td>7.666</td>
<td>5.593</td>
<td>2.706</td>
<td>1.454</td>
<td>1.226</td>
</tr>
<tr>
<td>CBR</td>
<td>300</td>
<td>21.16</td>
<td>15.52</td>
<td>6.699</td>
<td>3.105</td>
</tr>
<tr>
<td>600</td>
<td>33.69</td>
<td>25.51</td>
<td>11.41</td>
<td>6.021</td>
<td>4.676</td>
</tr>
<tr>
<td>1000</td>
<td>44.58</td>
<td>36.20</td>
<td>17.03</td>
<td>7.094</td>
<td>5.371</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.736</td>
<td>0.753</td>
<td>0.627</td>
<td>0.751</td>
<td>0.850</td>
</tr>
<tr>
<td>10</td>
<td>0.649</td>
<td>0.737</td>
<td>0.780</td>
<td>0.824</td>
<td>0.867</td>
</tr>
<tr>
<td>100</td>
<td>1.960</td>
<td>1.705</td>
<td>1.428</td>
<td>1.160</td>
<td>1.149</td>
</tr>
<tr>
<td>VBR</td>
<td>300</td>
<td>4.652</td>
<td>4.724</td>
<td>3.760</td>
<td>2.692</td>
</tr>
<tr>
<td>600</td>
<td>10.31</td>
<td>9.629</td>
<td>7.289</td>
<td>5.520</td>
<td>4.958</td>
</tr>
<tr>
<td>1800</td>
<td>14.80</td>
<td>12.59</td>
<td>8.039</td>
<td>5.687</td>
<td>5.694</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.758</td>
<td>0.756</td>
<td>0.872</td>
<td>0.894</td>
<td>0.825</td>
</tr>
<tr>
<td>10</td>
<td>0.649</td>
<td>0.737</td>
<td>0.780</td>
<td>0.824</td>
<td>0.867</td>
</tr>
<tr>
<td>70</td>
<td>1.335</td>
<td>1.101</td>
<td>1.066</td>
<td>1.181</td>
<td>0.978</td>
</tr>
<tr>
<td>VTR</td>
<td>140</td>
<td>1.694</td>
<td>1.162</td>
<td>1.979</td>
<td>1.791</td>
</tr>
<tr>
<td>600</td>
<td>13.28</td>
<td>13.76</td>
<td>13.81</td>
<td>13.18</td>
<td>12.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-1.64</td>
<td>-2.30</td>
<td>-2.14</td>
<td>-1.61</td>
<td>-1.01</td>
</tr>
<tr>
<td>10</td>
<td>-0.93</td>
<td>-1.65</td>
<td>-2.41</td>
<td>-2.98</td>
<td>-2.58</td>
</tr>
<tr>
<td>35</td>
<td>0.237</td>
<td>-0.31</td>
<td>-0.35</td>
<td>-1.02</td>
<td>-0.96</td>
</tr>
<tr>
<td>SVD</td>
<td>140</td>
<td>4.732</td>
<td>4.640</td>
<td>4.152</td>
<td>2.287</td>
</tr>
</tbody>
</table>

Finally, we investigated the impact of call arrival assumptions at different levels of ingress-egress aggregation by comparing the results with Poisson and BATCH arrivals. We reported in draft-zhang-pcn-performance-evaluation that virtual queue-based
admission is relatively insensitive to the BATCH vs Poisson arrivals, even at lower aggregation levels. In contrast, the call arrival assumption does affect the performance of token bucket-based algorithm, and causes substantial degradation of performance at low ingress–egress aggregation level. An example result with CBR traffic is presented in table A.5. Here we use batch arrival with mean = 5. The results show that with the lowest aggregation, the batch arrival gives worse result than the normal Poisson arrival, however, as the level of aggregation become sufficient (e.g. 100 ingress, 10 call/ingress), the difference becomes insignificant. This behavior is consistent across all types of traffic.

<table>
<thead>
<tr>
<th>No. Ingr</th>
<th>Deviation Interval No-Rand</th>
<th>0.0001</th>
<th>0.001</th>
<th>0.005</th>
<th>0.01</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.918</td>
<td>1.007</td>
<td>0.836</td>
<td>0.933</td>
<td>1.014</td>
<td>0.971</td>
</tr>
<tr>
<td>10</td>
<td>1.221</td>
<td>0.936</td>
<td>0.767</td>
<td>0.906</td>
<td>0.920</td>
<td>0.857</td>
</tr>
<tr>
<td>100</td>
<td>8.857</td>
<td>7.092</td>
<td>3.265</td>
<td>1.821</td>
<td>1.463</td>
<td>1.036</td>
</tr>
<tr>
<td>CBR</td>
<td>300</td>
<td>22.59</td>
<td>8.596</td>
<td>4.979</td>
<td>4.550</td>
<td>2.165</td>
</tr>
<tr>
<td>600</td>
<td>43.36</td>
<td>37.12</td>
<td>17.37</td>
<td>10.02</td>
<td>8.005</td>
<td>4.223</td>
</tr>
<tr>
<td>1000</td>
<td>63.60</td>
<td>50.36</td>
<td>25.48</td>
<td>12.82</td>
<td>9.339</td>
<td>6.219</td>
</tr>
</tbody>
</table>

(Table A.5 In/Egress Aggregation with batch traffic: Token-based results)

8.5.3. Effect of Multiple Bottlenecks

The results in Table A.2 (Section 9.5.1, parameter sensitivity study) implied that from the bottleneck point of view, the performance on the multiple-bottleneck topology, for all types of traffic, is comparable to the ones on the SingleLink, for both queue-based and token bucket-based algorithms. However, the results in Table A.2 only show the worst case values over all bottleneck links. In this section we consider two other aspects of the Multiple Bottleneck effects: relative performance at individual bottlenecks and fairness of bandwidth usage between the short- and the long- haul ingress-egress aggregates.

8.5.3.1. Relative performance of different bottlenecks

In Table A.5, we show a snapshot of the behavior with 5 bottleneck topology, with the goal of studying the performance of different bottlenecks more closely. Here, the over-admission-percentage displayed is an average across all 15 experiments with different [weight, CLE] setting. (We do observe the same behavior in each of the individual experiment, hence providing a summarized statistics is
meaningful).

One differences in token-bucket case vs the queue-based admissions in the PLT topology case revealed in Table A.6 is that there appears to be a consistent relationship between the position of the bottleneck link (how far downstream it is) and its over-admission-percentage. The data shows the further downstream the bottleneck is, the more it tends to over-admit, regardless the type of the traffic. The exact cause of this phenomenon is yet to be explained, but the effect of it seems to be insignificant in magnitude, at least in the experiments we ran.

<table>
<thead>
<tr>
<th>Traffic Type</th>
<th>Traffic Type</th>
<th>Bottleneck LinkId</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CBR</td>
<td>0.288</td>
<td>0.286</td>
</tr>
<tr>
<td>VBR</td>
<td>0.319</td>
<td>0.420</td>
</tr>
<tr>
<td>MIX</td>
<td>0.363</td>
<td>0.394</td>
</tr>
<tr>
<td>VTR</td>
<td>0.466</td>
<td>0.309</td>
</tr>
<tr>
<td>SVD</td>
<td>0.319</td>
<td>0.420</td>
</tr>
<tr>
<td>CBR</td>
<td>0.121</td>
<td>0.300</td>
</tr>
<tr>
<td>VBR</td>
<td>-0.07</td>
<td>0.251</td>
</tr>
<tr>
<td>MIX</td>
<td>0.042</td>
<td>0.350</td>
</tr>
<tr>
<td>VTR</td>
<td>0.277</td>
<td>0.488</td>
</tr>
<tr>
<td>SVD</td>
<td>-2.64</td>
<td>-2.50</td>
</tr>
</tbody>
</table>

(Table A.6 Bottleneck Performance: queue-based v.s. token bucket-based)
8.5.3.2. (Un)Fairness Between Different Ingress-Egress pairs

It was reported in draft-zhang-pcn-performance-evaluation that virtual-queue-based admission control favors significantly short-haul connection over long-haul connections. As was discussed there, this property is in fact common for measurement-based admission control algorithms (see for example [Jamin] for a discussion). It is common knowledge that in the limit of large demands, long-haul connections can be completely starved. We show in draft-zhang-performance-evaluation that in fact starvation of long-haul connections can occur even with relatively small (but constant) overloads. We identify there that the primary reason for it is a desynchronization of the "congestion periods" at different bottlenecks, resulting in the long-haul connections almost always seeing at least one bottleneck and hence almost never being allowed to admit new flows. We refer the reader to that draft for more detail.

Here we investigate the comparative behavior of the token-bucket based scheme and virtual queue based scheme with respect to fairness.

The fairness is illustrated using the ratio between bandwidth of the long-haul aggregates and the short-haul aggregates. Several potential factors that can effect the level of unfairness are the levels of demand overload, the EWMA weight and CLE and the number of bottleneck links traversed by the long-haul aggregate.

As is intuitively expected, (and also confirmed experimentally), the unfairness is the larger the higher the demand, and the more bottlenecks traversed by the long-haul aggregate Therefore, we report here the "worst case" results across our experiments corresponding to the 5x demand overload and the 5-PLT topology.

Table A.7 summaries, at 5x overload, with CLE=0.05 (for virtual queue), 0.0001(for token bucket), the fairness results to different weight and topology. We display the ratio as function of time, in 10 sec increments, (the reported ratios are averaged over the corresponding 10 simulation-second interval). The result presented in this section uses the aggregates that traverse the first bottleneck. The results on all other bottlenecks are extremely similar.
<table>
<thead>
<tr>
<th>Topo</th>
<th>Weight</th>
<th>Simulation Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>PLT5</td>
<td>0.1</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>1.03</td>
</tr>
<tr>
<td>Virtual Queue</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>Based</td>
<td>PLT3</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>1.02</td>
</tr>
<tr>
<td>Token Bucket</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>Based</td>
<td>PLT2</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>1.02</td>
</tr>
</tbody>
</table>

(Table A.7 Fairness performance: Virtual Queue v.s. Token Bucket. The numbers in the cells represent the ratio between the bandwidth of the long- and short-haul aggregates. Each row represents the time series of these results in 10 simulation second increments.)

9. Appendix B. Controlling The Single Marking Configuration with a Single Parameter

9.1. Details of the Proposed Enhancements to PCN Architecture
9.1.1. PCN-Internal-Node

No substantive change is required for the PCN framework (as defined in [I-D.eardley-pcn-architecture]) to enable Single Marking Operation in the PCN Internal Node. The architecture already allows the implementation of only one marking and metering algorithm at the PCN-internal-node.

However, we propose to rename the terms "configured-admissible-rate" and "configured-termination-rate" to "Type Q threshold" and "Type R" threshold. The architecture should allow configuring either one of these thresholds or both at the PCN-ingress node. The type of the threshold determines the type of the marking semantics/algorithm associated with the threshold.

9.1.2. PCN-Egress-Node

The only proposed change at the PCN-egress-node is the addition of a single (globally defined) configuration constant U. The setting of this constant defines the type of marking CLE is measured against. If U=1, the system defaults to the dual-marking behavior and the CLE is measured against Type Q marked packets. If U>1, the CLE is measured against Type R marked traffic. No other change is required.

In more detail,

- If U=1, a PCN-egress-node expects to receive either Type Q marking only (the network implements virtual-queue-based admission only), or Type R marking only (the system implements excess-rate-based flow termination only), or both (the system implements dual-marking admission and termination).

- If U>1, a PCN egress node expects to receive only type-R marking (the network implements single-marking approach).

- If U=1 and Type-Q marking is received (as indicated by the encoding in the PCN packets), then the PCN-egress-node always measures the CLE (fraction of traffic carrying Type-Q marks) on a per-ingress basis against Type Q marking. This represents no change (other than renaming "admission-marked-packets" to "type Q-marked" packets) compared to the current architecture. The PCN-egress-node then signals the (type Q-based) CLE to the PCN-ingress-node - again as already enabled by the current PCN architecture.

- If U=1 and a PCN-egress-node receives "Type R" marking (as indicated in the encoding of the PCN packets), it measures sustainable rate with respect to Type-R marked traffic, (i.e. it
measures the amount of traffic without the "Type-R" marks). This also is just a renaming change (with termination-marking renamed to "Type R" marking) and is fully compatible with the current PCN architecture.

- If U > 1, the PCN-egress node computes both the CLE and the Sustainable rate with respect to Type-R marking.
- Once computed, the CLE and/or the Sustainable rate are communicated to the PCN-ingress-node as described in [I-D.eardley-pcn-architecture].

### 9.1.3. PCN-Ingress-Node

The only proposed change at the PCN-ingress-node is the addition of a single (globally defined) configuration constant U (in fact, this is the same constant as defined for the PCN-egress-node, so U in fact is a per PCN-boundary-node constant; its value however is assumed to be global for all PCN-boundary nodes in the PCN-domain (or at least a subset of nodes communicating with each other only)). The value of this constant is used to multiply the sustainable rate received from a given PCN-egress-node to compute the rate threshold used for flow termination decisions. The value U=1 corresponds to the dual-marking approach, and results in using the sustainable rate received from the PCN-egress-node directly. The value U>1 corresponds to the single marking approach and its (globally defined) value signifies the desired system-wide implicit ratio between flow termination and flow admission thresholds as described in Section 2.

Note that constant U is assumed to be defined per PCN-boundary node (i.e. the ingress and the egress functions of the PCN-boundary-node use the same configuration constant to guide their behavior.

In more detail:

- A PCN-ingress-node receives CLE and/or Sustainable Rate from each PCN-egress-node it has traffic to. This is fully compatible with PCN architecture as described in [I-D.eardley-pcn-architecture].
- A PCN-ingress-node bases its admission decisions on the value of CLE. Specifically, once the value of CLE exceeds a configured threshold, the PCN-ingress-node stops admitting new flows. It restarts admitting when the CLE value goes down below the specified threshold. This is fully compatible with PCN architecture as described in draft-earley-pcn-architecture-00.
- A PCN-ingress node receiving a Sustainable Rate from a particular PCN-egress node measures its traffic to that egress node. This
again is fully compatible with PCN architecture as described in draft-earley-pcn-architecture-00.

- The PCN-ingress-node computes the desired Termination Rate to a particular PCN-egress-node by multiplying the sustainable rate from a given PCN-egress-node by the value of the configuration parameter U. This computation step represents a proposed change to the current version of [I-D.eardley-pcn-architecture].

- Once the Termination Rate is computed, it is used for the flow termination decision in a manner fully compatible with [I-D.eardley-pcn-architecture]. Namely the PCN-ingress-node compares the measured traffic rate destined to the given PCN-egress-node with the computed Termination rate for that egress node, and terminates a set of traffic flows to reduce the rate exceeding that Termination rate. This is fully compatible with [I-D.eardley-pcn-architecture].

### 9.2. Impact on PCN-Egress-Node

The only proposed change at the PCN-egress-node is the addition of a single (globally defined) configuration constant U. The setting of this constant defines the type of marking CLE is measured against. If U=1, the system defaults to the dual-marking behavior and the CLE is measured against Type Q marked packets. If U>1, the CLE is measured against Type R marked traffic. No other change is required.

In more detail,

- If U=1, a PCN-egress-node expects to receive either Type Q marking only (the network implements virtual-queue-based admission only), or Type R marking only (the system implements excess-rate-based flow termination only), or both (the system implements dual-marking admission and termination).

- If U>1, a PCN egress node expects to receive only type-R marking (the network implements single-marking approach).

- If U=1 and Type-Q marking is received (as indicated by the encoding in the PCN packets), then the PCN-egress-node always measures the CLE (fraction of traffic carrying Type-Q marks) on a per-ingress basis against Type Q marking. This represents no change (other than renaming "admission-marked-packets" to "type Q-marked" packets) compared to the current architecture. The PCN-egress-node then signals the (type Q-based) CLE to the PCN-ingress-node - again as already enabled by the current PCN architecture.
o If U=1 and a PCN-egress-node receives "Type R" marking (as indicated in the encoding of the PCN packets), it measures sustainable rate with respect to Type-R marked traffic, (i.e. it measures the amount of traffic without the "Type-R" marks). This also is just a renaming change (with termination-marking renamed to "Type R" marking) and is fully compatible with the current PCN architecture.

o If U > 1, the PCN-egress node computes both the CLE and the Sustainable rate with respect to Type-R marking.

o Once computed, the CLE and/or the Sustainable rate are communicated to the PCN-ingress-node as described in [I-D.eardley-pcn-architecture].

10. Security Considerations

TBD

11. References

11.1. Normative References


11.2. Informative References


(work in progress), June 2006.

[I-D.briscoe-tsvwg-re-ecn-tcp]
Briscoe, B., "Re-ECN: Adding Accountability for Causing Congestion to TCP/IP", draft-briscoe-tsvwg-re-ecn-tcp-03 (work in progress), October 2006.

[I-D.davie-ecn-mpls]

[I-D.eardley-pcn-architecture]

[I-D.lefaucheur-emergency-rsvp]

[I-D.westberg-pcn-load-control]

11.3.  References


Authors’ Addresses

Anna Charny
Cisco Systems, Inc.
1414 Mass. Ave.
Boxborough, MA 01719
USA

Email: acharny@cisco.com
Xinyang (Joy) Zhang
Cisco Systems, Inc. and Cornell University
1414 Mass. Ave.
Boxborough, MA  01719
USA

Email: joyzhang@cisco.com

Francois Le Faucheur
Cisco Systems, Inc.
Village d’Entreprise Green Side – Batiment T3 ,
400 Avenue de Roumanille, 06410 Biot Sophia-Antipolis,
France

Email: flefauch@cisco.com

Vassilis Liatsos
Cisco Systems, Inc.
1414 Mass. Ave.
Boxborough, MA  01719
USA

Email: vliatsos@cisco.com