Network Working Group
Request for Comments: 5670
Category: Standards Track

P. Eardley, Ed.
BT

November 2009

Metering and Marking Behaviour of PCN-Nodes

Abstract

The objective of Pre-Congestion Notification (PCN) is to protect the quality of service (QoS) of inelastic flows within a Diffserv domain in a simple, scalable, and robust fashion. This document defines the two metering and marking behaviours of PCN-nodes. Threshold-metering and -marking marks all PCN-packets if the rate of PCN-traffic is greater than a configured rate ("PCN-threshold-rate"). Excess-traffic-metering and -marking marks a proportion of PCN-packets, such that the amount marked equals the rate of PCN-traffic in excess of a configured rate ("PCN-excess-rate"). The level of marking allows PCN-boundary-nodes to make decisions about whether to admit or terminate PCN-flows.

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow

modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction

The objective of Pre-Congestion Notification (PCN) is to protect the quality of service (QoS) of inelastic flows within a Diffserv domain in a simple, scalable, and robust fashion. Two mechanisms are used: admission control to decide whether to admit or block a new flow request, and (in abnormal circumstances) flow termination to decide whether to terminate some of the existing flows. To achieve this, the overall rate of PCN-traffic is metered on every link in the domain, and PCN-packets are appropriately marked when certain configured rates are exceeded. These configured rates are below the rate of the link, thus providing notification to boundary nodes about

overloads before any congestion occurs (hence "Pre-Congestion Notification"). The level of marking allows boundary nodes to make decisions about whether to admit or terminate. Within the domain, PCN-traffic is forwarded in a prioritised Diffserv traffic class [RFC2475].

This document defines the two metering and marking behaviours of PCN-nodes. Their aim is to enable PCN-nodes to give an "early warning" of potential congestion before there is any significant build-up of PCN-packets in their queues. In summary, their objectives are:

- Threshold-metering and -marking: to mark all PCN-packets (with a "threshold-mark") when the bit rate of PCN-traffic is greater than its configured reference rate ("PCN-threshold-rate").

- Excess-traffic-metering and -marking: when the bit rate of PCN-packets is greater than its configured reference rate ("PCN-excess-rate"), to mark PCN-packets (with an "excess-traffic-mark") at a rate equal to the difference between the rate of PCN-traffic and the PCN-excess-rate.

Note that although [RFC3168] defines a broadly RED-like (Random Early Detection) default congestion marking behaviour, it allows alternatives to be defined; this document defines such an alternative.

Section 2 below describes the functions involved, which in outline (see Figure 1) are:

- Behaviour aggregate (BA) classification: decide whether or not an incoming packet is a PCN-packet.

- Dropping (optional): drop packets if the link is overloaded.

- Threshold-meter: determine whether the bit rate of PCN-traffic exceeds its configured reference rate (PCN-threshold-rate). The meter operates on all PCN-packets on the link, and not on individual flows.

- Excess-traffic-meter: measure by how much the bit rate of PCN-traffic exceeds its configured reference rate (PCN-excess-rate). The meter operates on all PCN-packets on the link, and not on individual flows.

- PCN-mark: actually mark the PCN-packets, if the meter functions indicate to do so.

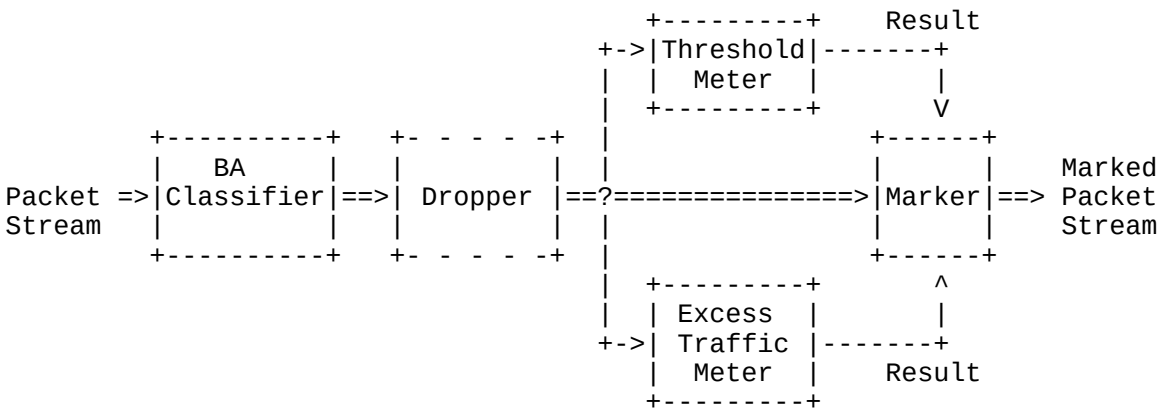

Figure 1: Schematic of PCN-interior-node functionality

Appendix A gives an example of algorithms that fulfil the specification of Section 2, and Appendix B provides some explanations of and comments on Section 2. Both the Appendices are informative.

The general architecture for PCN is described in [RFC5559], whilst [Menth10] is an overview of PCN.

1.1. Terminology

In addition to the terminology defined in [RFC5559] and [RFC2474], the following terms are defined:

- Competing-non-PCN-packet: a non-PCN-packet that shares a link with PCN-packets and competes with them for its forwarding bandwidth. Competing-non-PCN-packets MUST NOT be PCN-marked (only PCN-packets can be PCN-marked).

  Note: In general, it is not advised to have any competing-non-PCN-traffic.

  Note: There is likely to be traffic (such as best effort) that is forwarded at lower priority than PCN-traffic; although it shares the link with PCN-traffic, it doesn't compete for forwarding bandwidth, and hence it is not competing-non-PCN-traffic. See Appendix B.1 for further discussion about competing-non-PCN-traffic.

- Metered-packet: a packet that is metered by the metering functions specified in Sections 2.3 and 2.4. A PCN-packet MUST be treated as a metered-packet (with the minor exception noted below in Section 2.4). A competing-non-PCN-packet MAY be treated as a metered-packet.

1.1.1. 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].

2. Specified PCN-Metering and -Marking Behaviours

This section defines the two PCN-metering and -marking behaviours. The descriptions are functional and are not intended to restrict the implementation. The informative Appendices supplement this section.

2.1. Behaviour Aggregate Classification Function

A PCN-node MUST classify a packet as a PCN-packet if the value of its Differentiated Services Code Point (DSCP) and Explicit Congestion Notification (ECN) fields correspond to a PCN-enabled codepoint, as defined in the encoding scheme applicable to the PCN-domain (for example, [RFC5696] defines the baseline encoding). Otherwise, the packet MUST NOT be classified as a PCN-packet.

A PCN-node MUST classify a packet as a competing-non-PCN-packet if it is not a PCN-packet and it competes with PCN-packets for its forwarding bandwidth on a link.

2.2. Dropping Function

Note: If the PCN-node’s queue overflows, then naturally packets are dropped. This section describes additional action.

On all links in the PCN-domain, dropping MAY be done by first metering all metered-packets to determine if the rate of metered-traffic on the link is greater than the rate allowed for such traffic; if the rate of metered-traffic is too high, then drop metered-packets.

If the PCN-node drops PCN-packets, then:

- PCN-packets that arrive at the PCN-node already excess-traffic-marked SHOULD be preferentially dropped.

- the PCN-node's excess-traffic-meter SHOULD NOT meter the PCN-packets that it drops.

2.3. Threshold-Meter Function

A PCN-node MUST implement a threshold-meter that has behaviour functionally equivalent to the following.

The meter acts like a token bucket, which is sized in bits and has a configured reference rate (bits per second). The amount of tokens in the token bucket is termed F_tm. Tokens are added at the reference rate (PCN-threshold-rate), to a maximum value BS_tm. Tokens are removed equal to the size in bits of the metered-packet, to a minimum F_tm = 0. (Explanation of abbreviations: F is short for Fill of the token bucket, BS for bucket size, and tm for threshold-meter.)

The token bucket has a configured intermediate depth, termed threshold. If F_tm < threshold, then the meter indicates to the marking function that the packet is to be threshold-marked; otherwise, it does not.

2.4. Excess-Traffic-Meter Function

A packet SHOULD NOT be metered (by this excess-traffic-meter function) in the following two cases:

- if the PCN-packet is already excess-traffic-marked on arrival at the PCN-node.

- if this PCN-node drops the packet.

Otherwise, the PCN-packet MUST be treated as a metered-packet -- that is, it is metered by the excess-traffic-meter.

A PCN-node MUST implement an excess-traffic-meter. The excess-traffic-meter SHOULD indicate packets to be excess-traffic-marked, independent of their size ("packet size independent marking"); if "packet size independent marking" is not implemented, then the excess-traffic-meter MUST use the "classic" metering behaviour.

For the "classic" metering behaviour, the excess-traffic-meter has behaviour functionally equivalent to the following.

The meter acts like a token bucket, which is sized in bits and has a configured reference rate (bits per second). The amount of tokens in the token bucket is termed F_etm. Tokens are added at the reference rate (PCN-excess-rate), to a maximum value BS_etm. Tokens are removed equal to the size in bits of the metered-packet, to a minimum

F_etm = 0. If the token bucket is empty (F_etm = 0), then the meter indicates to the marking function that the packet is to be excess-traffic-marked. (Explanation of abbreviations: F is short for Fill of the token bucket, BS for bucket size, and etm for excess-traffic-meter.)

For "packet size independent marking", the excess-traffic-meter has behaviour functionally equivalent to the following.

The meter acts like a token bucket, which is sized in bits and has a configured reference rate (bits per second). The amount of tokens in the token bucket is termed F_etm. Tokens are added at the reference rate (PCN-excess-rate), to a maximum value BS_etm. If the token bucket is not negative, then tokens are removed equal to the size in bits of the metered-packet (and the meter does not indicate to the marking function that the packet is to be excess-traffic-marked). If the token bucket is negative (F_etm < 0), then the meter indicates to the marking function that the packet is to be excess-traffic-marked (and no tokens are removed). (Explanation of abbreviations: F is short for Fill of the token bucket, BS for bucket size, and etm for excess-traffic-meter.)

Otherwise, the meter MUST NOT indicate marking.

2.5. Marking Function

A PCN-packet MUST be marked to reflect the metering results by setting its encoding state appropriately, as specified by the specific encoding scheme that applies in the PCN-domain. A consistent choice of encoding scheme MUST be made throughout a PCN-domain.

A PCN-node MUST NOT:

- PCN-mark a packet that is not a PCN-packet;
- change a non-PCN-packet into a PCN-packet;
- change a PCN-packet into a non-PCN-packet.

Note: Although competing-non-PCN-packets MAY be metered, they MUST NOT be PCN-marked.

3. Security Considerations

It is assumed that all PCN-nodes are PCN-enabled and are trusted for truthful PCN-metering and PCN-marking. If this isn’t the case, then there are numerous potential attacks. For instance, a rogue PCN-

interior-node could PCN-mark all packets so that no flows were admitted. Another possibility is that it doesn't PCN-mark any packets, even when it is pre-congested.

Note that PCN-interior-nodes are not flow-aware. This prevents some security attacks where an attacker targets specific flows in the data plane -- for instance, for Denial-of-Service (DoS) or eavesdropping.

As regards Security Operations and Management, PCN adds few specifics to the general good practice required in this field [RFC4778]. For example, it may be sensible for a PCN-node to raise an alarm if it is persistently PCN-marking.

Security considerations are further discussed in [RFC5559].

4. Acknowledgements

This document is the result of extensive collaboration within the PCN WG. Amongst the most active other contributors to the development of the ideas specified in this document have been Jozef Babiarz, Bob Briscoe, Kwok-Ho Chan, Anna Charny, Georgios Karagiannis, Michael Menth, Toby Moncaster, Daisuke Satoh, and Joy Zhang. Appendix A is based on text from Michael Menth.

This document is a development of [Briscoe06-2]. Its authors are therefore also contributors to this document: Jozef Babiarz, Attila Bader, Bob Briscoe, Kwok-Ho Chan, Anna Charny, Stephen Dudley, Philip Eardley, Georgios Karagiannis, Francois Le Faucheur, Vassilis Liatsos, Dave Songhurst, and Lars Westberg.

Thanks to those who've made comments on the document: Joe Babiarz, Fred Baker, David Black, Bob Briscoe, Ken Carlberg, Anna Charny, Ralph Droms, Mehmet Ersue, Adrian Farrel, Ruediger Geib, Wei Gengyu, Fortune Huang, Christian Hublet, Ingemar Johansson, Georgios Karagiannis, Alexey Melnikov, Michael Menth, Toby Moncaster, Dimitri Papadimitriou, Tim Polk, Daisuke Satoh, and Magnus Westerlund.

5. References

5.1. Normative Reference

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

5.2. Informative References

[Baker08] Baker, F., Polk, J., and M. Dolly, "DSCP for Capacity-Admitted Traffic", Work in Progress, November 2008.

[Briscoe06-1] Briscoe, B., Eardley, P., Songhurst, D., Le Faucheur, F., Charny, A., Babiarz, J., Chan, K., Dudley, S., Karagiannis, G., Bader, A., and L. Westberg, "An edge-to-edge Deployment Model for Pre-Congestion Notification: Admission Control over a DiffServ Region", Work in Progress, October 2006.

[Briscoe06-2] Briscoe, B., Eardley, P., Songhurst, D., Le Faucheur, F., Charny, A., Liatsos, V., Babiarz, J., Chan, K., Dudley, S., Karagiannis, G., Bader, A., and L. Westberg, "Pre-Congestion Notification marking", Work in Progress, October 2006.

[Briscoe08] Briscoe, B., "Byte and Packet Congestion Notification", Work in Progress, August 2008.

[Charny07] Charny, A., Babiarz, J., Menth, M., and X. Zhang, "Comparison of Proposed PCN Approaches", Work in Progress, November 2007.

[Menth10] Menth, M., Lehrieder, F., Briscoe, B., Eardley, P., Moncaster, T., Babiarz, J., Chan, K., Charny, A., Karagiannis, G., Zhang, X., Taylor, T., Satoh, D., and R. Geib, "A Survey of PCN-Based Admission Control and Flow Termination", IEEE Communications Surveys and Tutorials, 2010 (third issue), <http://www3.informatik.uni-wuerzburg.de/staff/menth/Publications/papers/Menth08-PCN-Overview.pdf>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC4778] Kaeo, M., "Operational Security Current Practices in Internet Service Provider Environments", RFC 4778, January 2007.

[RFC5127] Chan, K., Babiarz, J., and F. Baker, "Aggregation of DiffServ Service Classes", RFC 5127, February 2008.

[RFC5559] Eardley, P., "Pre-Congestion Notification (PCN) Architecture", RFC 5559, June 2009.

[RFC5696] Moncaster, T., Briscoe, B., and M. Menth, "Baseline Encoding and Transport of Pre-Congestion Information", RFC 5696, November 2009.

[Taylor09] Charny, A., Huang, F., Menth, M., and T. Taylor, "PCN Boundary Node Behaviour for the Controlled Load (CL) Mode of Operation", Work in Progress, March 2009.

Appendix A. Example Algorithms

Note: This Appendix is informative, not normative. It is an example of algorithms that implement Section 2 and is based on [Charny07] and [Menth10].

There is no attempt to optimise the algorithms. The metering and marking functions are implemented together. It is assumed that three encoding states are available (one for threshold-marked, one for excess-traffic-marked, and one for not-marked). It is assumed that all metered-packets are PCN-packets and that the link is never overloaded. For excess-traffic-marking, "packet size independent marking" applies.

A.1. Threshold-Metering and -Marking

A token bucket with the following parameters:

* PCN-threshold-rate: token rate of token bucket (bits/second)

* BS_tm: depth of token bucket (bits)

* threshold: marking threshold of token bucket (bits)

* lastUpdate: time the token bucket was last updated (seconds)

* F_tm: amount of tokens in token bucket (bits)

A PCN-packet has the following parameters:

* packet_size: the size of the PCN-packet (bits)

* packet_mark: the PCN encoding state of the packet

In addition there is the parameter:

now: the current time (seconds)

The following steps are performed when a PCN-packet arrives on a link:

* F_tm = min(BS_tm, F_tm + (now - lastUpdate) * PCN-threshold-rate); // add tokens to token bucket

* F_tm = max(0, F_tm - packet_size); // remove tokens from token bucket

* if ((F_tm < threshold) AND (packet_mark != excess-traffic-marked)) then packet_mark = threshold-marked; // do threshold-marking, but don't re-mark packets that are already excess-traffic-marked

* lastUpdate = now // Note: 'now' has the same value as in step 1

A.2. Excess-Traffic-Metering and -Marking

A token bucket with the following parameters:

* PCN-excess-rate: token rate of token bucket (bits/second)

* BS_etm: depth of TB in token bucket (bits)

* lastUpdate: time the token bucket was last updated (seconds)

* F_etm: amount of tokens in token bucket (bits)

A PCN-packet has the following parameters:

* packet_size: the size of the PCN-packet (bits)

* packet_mark: the PCN encoding state of the packet

In addition there is the parameter:

* now: the current time (seconds)

The following steps are performed when a PCN-packet arrives on a link:

* F_etm = min(BS_etm, F_etm + (now - lastUpdate) * PCN-excess-rate); // add tokens to token bucket

* if (packet_mark != excess-traffic-marked) then // do not meter packets that are already excess-traffic-marked

  + if (F_etm < 0) then packet_mark = excess-traffic-marked; // do excess-traffic-marking. The algorithm ensures this is independent of packet size

  + else F_etm = F_etm - packet_size; // remove tokens from token bucket if don't mark packet

* lastUpdate = now // Note: 'now' has the same value as in step 1

Appendix B. Implementation Notes

Note: This Appendix is informative, not normative. It comments on Section 2, including reasoning about whether MUSTs or SHOULDs are required. For guidance on Operations and Management considerations, please see [RFC5559].

B.1. Competing-Non-PCN-Traffic

In general, it is not advised to have any competing-non-PCN-traffic, essentially because the unpredictable amount of competing-non-PCN-traffic makes the PCN mechanisms less accurate and so reduces PCN's ability to protect the QoS of admitted PCN-flows [RFC5559]. But if there is competing-non-PCN-traffic, then:

1. There should be a mechanism to limit it, for example:

   * limit the rate at which competing-non-PCN-traffic can be forwarded on each link in the PCN-domain. One method for achieving this is to queue competing-non-PCN-packets separately from PCN-packets and to limit the scheduling rate of the former. Another method is to drop competing-non-PCN-packets in excess of some rate.

   * police competing-non-PCN-traffic at the PCN-ingress-nodes, as in the Diffserv architecture, for example. However, Diffserv's static traffic conditioning agreements risk a focused overload of traffic from several PCN-ingress-nodes onto one link.

   * by design, it is known that the level of competing-non-PCN-traffic is always very small -- perhaps it consists of operator control messages only.

2. In general, PCN's mechanisms should take account of competing-non-PCN-traffic, in order to improve the accuracy of the decision about whether to admit (or terminate) a PCN-flow. For example:

   * competing-non-PCN-traffic contributes to the PCN-meters; competing-non-PCN-packets are treated as metered-packets.

   * each PCN-node, on its links: (1) reduces the reference rates (PCN-threshold-rate and PCN-excess-rate), in order to allow 'headroom' for the competing-non-PCN-traffic; (2) limits the maximum forwarding rate of competing-non-PCN-traffic to be less than the 'headroom'. In this case, competing-non-PCN-packets are not treated as metered-packets.

3. The operator should decide on appropriate action. Dropping is discussed further in Appendix B.4.

One specific example of competing-non-PCN-traffic occurs if the PCN-compatible Diffserv codepoint is one of those that [Baker08] defines as suitable for use with admission control and there is such non-PCN-traffic in the PCN-domain. A similar example could occur for Diffserv codepoints of the Real-Time Treatment Aggregate [RFC5127]. In such cases, PCN-traffic and competing-non-PCN-traffic are distinguished by different values of the ECN field [RFC5696].

Another example would occur if there is more than one PCN-compatible Diffserv codepoint in a PCN-domain. For instance, suppose there are two PCN-BAs treated at different priorities. Then as far as the lower priority PCN-BA is concerned, the higher priority PCN-traffic needs to be treated as competing-non-PCN-traffic.

B.2. Scope

It may be known, for instance by the design of the network topology, that some links can never be pre-congested (even in unusual circumstances, such as after the failure of some links). There is then no need to deploy the PCN-metering and -marking behaviour on those links.

The meters can be implemented on the ingoing or outgoing interface of a PCN-node. It may be that existing hardware can support only one meter per ingoing interface and one per outgoing interface. Then, for instance, threshold-metering could be run on all the ingoing interfaces and excess-traffic-metering on all the outgoing interfaces; note that the same choice must be made for all the links in a PCN-domain to ensure that the two metering behaviours are applied exactly once for all the links.

The baseline encoding [RFC5696] specifies only two encoding states (PCN-marked and not-marked). In this case, "excess-traffic-marked" means a packet that is PCN-marked as a result of the excess-traffic-meter function, and "threshold-marked" means a packet that is PCN-marked as a result of the threshold-meter function. As far as terminology is concerned, this interpretation is consistent with that defined in [RFC5559]. Note that a deployment needs to make a consistent choice throughout the PCN-domain whether PCN-marked is interpreted as excess-traffic-marked or threshold-marked.

Note that even if there are only two encoding states, it is still required that both the meters are implemented, in order to ease compatibility between equipment and to remove a configuration option and associated complexity. Hardware with limited availability of

token buckets could be configured to run only one of the meters, but it must be possible to enable either meter. Although, in the scenario with two encoding states, indications from one of the meters are ignored by the marking function, they may be logged or acted upon in some other way, for example, by the management system or an explicit signalling protocol; such considerations are out of the scope of this document.

B.3. Behaviour Aggregate Classification

Configuration of PCN-nodes will define what values of the DSCP and ECN fields indicate a PCN-packet in a particular PCN-domain. For instance, [RFC5696] defines the baseline encoding.

Configuration will also define what values of the DSCP and ECN fields indicate a competing-non-PCN-packet in a particular PCN-domain.

B.4. Dropping

The objective of the dropping function is to minimise the queueing delay suffered by metered-traffic at a PCN-node, since PCN-traffic (and perhaps competing-non-PCN-traffic) is expected to be inelastic traffic generated by real-time applications. In practice, it would be defined as exceeding a specific traffic profile, typically based on a token bucket.

If there is no competing-non-PCN-traffic, then it is not expected that the dropping function is needed, since PCN's flow admission and termination mechanisms limit the amount of PCN-traffic. Even so, it still might be implemented as a back stop against misconfiguration of the PCN-domain, for instance.

If there is competing-non-PCN-traffic, then the details of the dropping function will depend on how the router's implementation handles the two sorts of traffic:

1. a common queue for PCN-traffic and competing-non-PCN-traffic, with a traffic conditioner for the competing-non-PCN-traffic; or

2. separate queues, in which case the amount of competing-non-PCN-traffic can be limited by limiting the rate at which the scheduler (for the competing-non-PCN-traffic) forwards packets.

(The discussion here is based on that in [Baker08].)

Note that only dropping of packets is allowed. Downgrading of packets to a lower priority BA is not allowed (see Appendix B.7), since it would lead to packet mis-ordering. Shaping ("the process of delaying packets" [RFC2475]) is not suitable if the traffic comes from real-time applications.

Preferential dropping of competing-non-PCN-traffic:
In general, it is reasonable for competing-non-PCN-traffic to get harsher treatment than PCN-traffic (that is, competing-non-PCN-packets are preferentially dropped) because PCN's flow admission and termination mechanisms are stronger than the mechanisms that are likely to be applied to the competing-non-PCN-traffic. The PCN mechanisms also mean that a dropper should not be needed for the PCN-traffic.

Preferential dropping of excess-traffic-marked packets:
Section 2.2 specifies, "If the PCN-node drops PCN-packets, then ... PCN-packets that arrive at the PCN-node already excess-traffic-marked SHOULD be preferentially dropped". In brief, the reason is that, with the "controlled load" edge behaviour [Taylor09], this avoids over-termination in the event of multiple bottlenecks in the PCN-domain [Charny07]. A fuller explanation is as follows. The optimal dropping behaviour depends on the particular edge behaviour [Menth10]. A single dropping behaviour is defined, as it is simpler to standardise, implement, and operate. The standardised dropping behaviour is at least adequate for all edge behaviours (and good for some), whereas others are not (for example, with tail dropping, far too much traffic may be terminated with the "controlled load" edge behaviour, in the event of multiple bottlenecks in the PCN-domain [Charny07]). The dropping behaviour is defined as a 'SHOULD', rather than a 'MUST', in recognition that other dropping behaviour may be preferred in particular circumstances, for example: (1) with the "marked flow" termination edge behaviour, preferential dropping of unmarked packets may be better [Menth10]; (2) tail dropping may make PCN-marking behaviour easier to implement on current routers.

Exactly what "preferentially dropped" means is left to the implementation. It is also left to the implementation what to do if there are no excess-traffic-marked PCN-packets available at a particular instant.

Section 2.2 also specifies, "the PCN-node's excess-traffic-meter SHOULD NOT meter the PCN-packets that it drops." This avoids over-termination [Menth10]. Effectively, it means that the dropping function (if present) should be done before the meter functions -- which is natural.

B.5. Threshold-Metering

The description is in terms of a 'token bucket with threshold' (which [Briscoe06-1] views as a virtual queue). However, the description is not intended to standardise implementation.

The reference rate of the threshold-meter (PCN-threshold-rate) is configured at less than the rate allocated to the PCN-traffic class. Also, the PCN-threshold-rate is less than, or possibly equal to, the PCN-excess-rate.

Section 2.3 specifies, "If F_tm < threshold, then the meter indicates to the marking function that the packet is to be threshold-marked; otherwise, it does not." Note that a PCN-packet is marked without explicit additional bias for the packet's size.

The behaviour must be functionally equivalent to the description in Section 2.3. "Functionally equivalent" means the observable 'black box' behaviour is the same or very similar, for example, if either precisely the same set of packets is marked or if the set is shifted by one packet. It is intended to allow implementation freedom over matters such as:

- whether tokens are added to the token bucket at regular time intervals or only when a packet is processed.
- whether the new token bucket depth is calculated before or after it is decided whether to PCN-mark the packet. The effect of this is simply to shift the sequence of marks by one packet.
- when the token bucket is very nearly empty and a packet arrives larger than F_tm, then the precise change in F_tm is up to the implementation. For instance:
  - set F_tm = 0 and indicate threshold-mark to the marking function.
  - check whether F_tm < threshold and if it is, then indicate threshold-mark to the marking function; then set F_tm = 0.
  - leave F_tm unaltered and indicate threshold-mark to the marking function.
- similarly, when the token bucket is very nearly full and a packet arrives larger than (BS_tm - F_tm), then the precise change in F_tm is up to the implementation.

Note that all PCN-packets, even if already marked, are metered by the threshold-meter function (unlike the excess-traffic-meter function), because all packets should contribute to the decision whether there is room for a new flow.

B.6. Excess-Traffic-Metering

The description is in terms of a token bucket, however the implementation is not standardised.

The reference rate of the excess-traffic-meter (PCN-excess-rate) is configured at less than (or possibly equal to) the rate allocated to the PCN-traffic class. Also, the PCN-excess-rate is greater than, or possibly equal to, the PCN-threshold-rate.

As in Section B.5, "functionally equivalent" allows some implementation flexibility, for example, the exact algorithm when the token bucket is very nearly empty or very nearly full.

Section 2.4 specifies, "A packet SHOULD NOT be metered (by this excess-traffic-meter function) ... if the packet is already excess-traffic-marked on arrival at the PCN-node". This avoids over-termination (with some edge behaviours) in the event that the PCN-traffic passes through multiple bottlenecks in the PCN-domain [Charny07]. Note that an implementation could determine whether the packet is already excess-traffic-marked as an integral part of its BA classification function. The behaviour is defined as a 'SHOULD NOT', rather than a 'MUST NOT', because it may be slightly harder to implement than a metering function that is blind to previous packet markings.

Section 2.4 specifies, "A packet SHOULD NOT be metered (by this excess-traffic-meter function) ... if this PCN-node drops the packet." This avoids over-termination [Menth10]. (A similar statement could also be made for the threshold-meter function but is irrelevant, as a link that is overloaded will already be substantially pre-congested and hence threshold-marking all packets.) It seems natural to perform the dropping function before the metering functions, although for some equipment it may be harder to implement; hence, the behaviour is defined as a 'SHOULD NOT', rather than a 'MUST NOT'.

"Packet size independent marking" -- excess-traffic-marking that is independent of packet size -- is specified as a 'SHOULD' rather than a 'MUST' in Section 2.4 because it may be slightly harder for some equipment to implement, and the impact of not doing so is undesirable but moderate (sufficient traffic is terminated, but flows with large packets are more likely to be terminated). With the "classic"

excess-traffic-meter behaviour, large packets are more likely to be excess-traffic-marked than small packets (because packets are marked if the number of tokens in the token bucket is smaller than the packet size). This means that, with some edge behaviours, flows with large packets are more likely to be terminated than flows with small packets ([Briscoe08], [Menth10]). "Packet size independent marking" can be achieved by a small modification of the "classic" excess-traffic-meter. The number of tokens in the bucket can become negative; if this number is negative at a packet's arrival, the packet is marked; otherwise, the amount of tokens equal to the packet size is removed from the bucket. Note that with "packet size independent marking", either the packet is marked or tokens are removed -- never both. Hence, the token bucket cannot become more negative than the maximum packet size on the link. The algorithm described in Appendix A implements this behaviour.

Note that BS_etm is independent of BS_tm, F_etm is independent of F_tm (except in that a packet can change both), and the two configured rates (PCN-excess-rate and PCN-threshold-rate) are independent (except that PCN-excess-rate >= PCN-threshold-rate).

B.7. Marking

Section 2.5 defines, "A PCN-node MUST NOT ...change a PCN-packet into a non-PCN-packet". This means that a PCN-node is not allowed to downgrade a PCN-packet into a lower priority Diffserv BA (hence, downgrading is not allowed as an alternative to dropping).

Section 2.5 defines, "A PCN-node MUST NOT ...PCN-mark a packet that is not a PCN-packet". This means that in the scenario where competing-non-PCN-packets are treated as metered-packets, a meter may indicate a packet is to be PCN-marked, but the marking function knows it cannot be marked. It is left open to the implementation exactly what to do in this case; one simple possibility is to mark the next PCN-packet. Note that unless the PCN-packets are a large fraction of all the metered-packets, the PCN mechanisms may not work well.

Although the metering functions are described separately from the marking function, they can be implemented in an integrated fashion.

Author’s Address

Philip Eardley (editor)
BT
Adastral Park, Martlesham Heath
Ipswich IP5 3RE
UK

EMail: philip.eardley@bt.com