PCN Working Group Philip. Eardley (Editor)
Internet-Draft BT
Intended status: Standards Track March 5, 2009
Expires: September 6, 2009

Marking behaviour of PCN-nodes
draft-ietf-pcn-marking-behaviour-02

Status of this Memo

Copyright Notice

Abstract

This document standardises the two marking behaviours of PCN-nodes: threshold marking and excess traffic marking. Threshold marking marks all PCN-packets if the PCN traffic rate is greater than a configured rate ("PCN-threshold-rate"). Excess traffic marking marks a proportion of PCN-packets, such that the amount marked equals the traffic rate in excess of a configured rate ("PCN-excess-rate"). Setting the configured rates below the physical link rates enables PCN-nodes to provide information to support admission control and flow termination in order to protect the quality of service of established inelastic flows.

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

This document standardises the two 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 marking: its objective is to mark all PCN-packets (with a "threshold-mark") whenever the rate of PCN-packets is greater than its configured rate ("PCN-threshold-rate");

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

[I-D.ietf-pcn-architecture] describes a general architecture for how, in a particular DiffServ domain, PCN-boundary-nodes convert these PCN-markings into decisions about flow admission and flow termination. Other documents describe the wider per-domain behaviour and how the PCN-markings are encoded in packet headers. PCN encoding uses a combination of the DSCP field and ECN field in the IP header to indicate that a packet is a PCN-packet and whether it is PCN-marked. The baseline encoding [I-D.ietf-pcn-baseline-encoding] standardises two encoding states (PCN-marked and not-marked), whilst other documents define extended schemes with three encoding states (eg [I-D.moncaster-pcn-3-state-encoding] defines PCN-threshold-marked, PCN-excess-traffic-marked and not-marked). [RFC3168] defines a broadly RED-like default congestion marking behaviour, but allows alternatives to be defined; this document defines such an alternative.

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

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

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

- Threshold meter: determine whether the rate of PCN-packets is greater than its configured PCN-threshold-rate. The meter operates on the aggregate of all PCN-packets on the link, and not on individual flows.

- Excess traffic meter: measure by how much the rate of PCN-packets is greater than its configured PCN-excess-rate. The meter operates on the aggregate of 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.

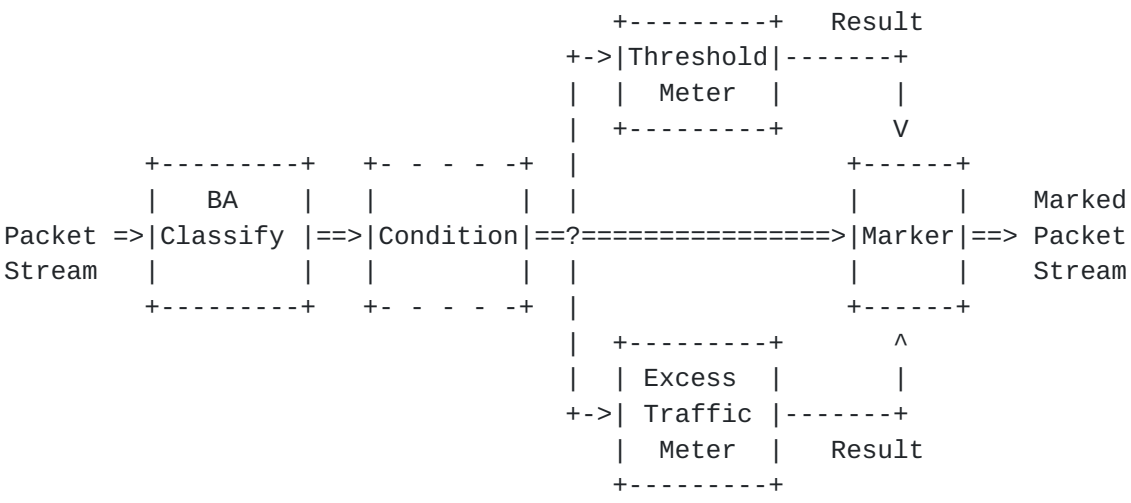

Figure 1: Schematic of functions for PCN-marking

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.

1.1. Terminology

In addition to the terminology defined in [I-D.ietf-pcn-architecture] 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 (ie only PCN-packets can be PCN-marked). Note: In general it is not advised to have any competing-non-PCN-traffic.

- Metered-packet: a packet that is metered by the metering functions specified below. 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.

- Note: "Excess-traffic-marked" means a packet that is marked as a result of the excess traffic meter function; "threshold-marked"

means a packet that is marked as a result of the threshold meter function. [I-D.ietf-pcn-baseline-encoding] defines only two PCN encoding states available (PCN-marked and not-marked); the deployment MUST choose whether PCN-marked is interpreted as excess-traffic-marked or threshold-marked; a consistent choice MUST be made throughout a PCN-domain.

2. Specified PCN-marking behaviour

This section specifies the PCN-marking behaviour. 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 DSCP and ECN fields correspond to a PCN-enabled codepoint, as defined in the encoding scheme applicable to the PCN-domain. 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. Traffic conditioning function

Note: if the PCN-node's queue overflows then naturally packets are dropped; traffic conditioning is action additional to this.

On all links in the PCN-domain, traffic conditioning MAY be done by:

- metering all metered-packets to determine if the rate of metered-traffic is greater than its scheduling rate (ie determine if any packets are out-of-profile. Metering is "the process of measuring the temporal properties (eg rate) of a traffic stream" [RFC2475].)

- 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 bit rate, termed PCN-threshold-rate. The amount of tokens in the token bucket is termed TBthreshold.fill. Tokens are added at the PCN-threshold-rate, to a maximum value TBthreshold.max. Tokens are removed equal to the size in bits of the metered-packet, to a minimum TBthreshold.fill=0.

The token bucket has a configured intermediate depth, termed TBthreshold.threshold. If TBthreshold.fill < TBthreshold.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 PCN-packet SHOULD NOT be metered (by this excess traffic meter function) in the following two cases:

- If the 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 that has behaviour functionally equivalent to the following.

The meter acts like a token bucket, which is sized in bits and has a configured bit rate, termed PCN-excess-rate. The amount of tokens in the token bucket is termed TBexcess.fill. Tokens are added at the PCN-excess-rate, to a maximum value TBexcess.max. Tokens are removed equal to the size in bits of the metered-packet, to a minimum TBexcess-fill=0. If the token bucket is empty (TBexcess.fill = 0), then the meter indicates to the Marking function that the packet is to be excess-traffic-marked. The PCN-excess-rate is greater than (or equal to) the PCN-threshold-rate.

In addition to the above, if the token bucket is within an MTU of being empty, then the meter SHOULD indicate to the Marking function that the packet is to be excess-traffic-marked; MTU means the maximum size of PCN-packets on the link ("packet size independent marking").

Otherwise the meter MUST NOT indicate marking.

2.5. Marking function

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.

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.

Note: In some deployment scenarios there may be only two PCN encoding states available (PCN-marked and not-marked). In such scenarios, the deployment MUST choose whether the Threshold meter function or the Excess traffic meter function can trigger a packet to be PCN-marked; a consistent choice MUST be made throughout a PCN-domain.

3. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

4. Security Considerations

See [I-D.ietf-pcn-architecture]

5. Acknowledgements

Michael Menth, Joe Babiarz, Anna Charny reviewed a preliminary version of the draft-eardley-pcn-marking-behaviour-00 draft.

Thanks to those who've made comments on this draft: Michael Menth, Joe Babiarz, Anna Charny, Ruediger Geib, Wei Gengyu, Fortune Huang, Bob Briscoe, Toby Moncaster, Christian Hublet, Ingemar Johansson, Ken Carlberg, Georgios Karagiannis.

All the work by many people in the PCN WG.

6. Changes

6.1. Changes to -02 from -01

Updates as follows:

- added notes (end of S1.1 & 2.5) to clarify what "excess-traffic-marked" means when there is only one encoding for PCN-marking
- added explanations for in Section B.4 and B.6 about why various things are SHOULD or SHOULD NOT rather than MUST or MUST NOT.
- Deleted a couple of paragraphs about encoding states, as they are relevant to encoding documents rather than this document.

6.2. Changes to -01 from -00

Updates as follows:

- corrected the term 'not PCN-marked' to 'not-marked' (throughout)
- re-phrased the definition of competing-non-PCN-packets
- corrected the definition of metered-packet
- delete most of Section 2.5 (marking function). The material deleted belongs as part of [I-D.ietf-pcn-baseline-encoding]; other encoding schemes would need to include similar material.
- deleted Appendix C (it was only a temporary archive of material concerning per domain behaviour and PCN-boundary-node operation)
- clarifications throughout
- made all references Informative

6.3. Changes to -00

First version of WG draft, derived from draft-eardley-pcn-marking-behaviour-01, with the following changes:

- Removed material concerning per domain behaviour and PCN-boundary-node operation (temporarily archived to Appendix C)
- Removed mention of downgrading as an option for per-hop traffic conditioning. In fact, downgrading is no longer allowed because S 2.6 now says "A PCN-node MUST NOT ...change a PCN-packet into a non PCN-packet".

- Traffic conditioning is now a MAY. Since in general flow termination (not traffic conditioning) is PCN's method for handling problems of too much traffic.

- Metered-packets: competing-non-PCN-packets now MAY be metered. Since it is recommended that the operator doesn't allow any competing-non-PCN-traffic, and (if there is) there are potentially other ways of coping.

- No changes (outside traffic conditioning & metering of competing-non-PCN-traffic) to the Normative sections of the draft.

- Appendix B.1 added about competing-non-PCN-traffic. Recommended that there is no such traffic, but guidance given if there is.

7. Authors

Many people need to be added.

8. Informative References

[I-D.briscoe-tsvwg-byte-pkt-mark]
Briscoe, B., "Byte and Packet Congestion Notification", draft-briscoe-tsvwg-byte-pkt-mark-02 (work in progress), February 2008.

[I-D.briscoe-tsvwg-cl-architecture]
Briscoe, B., "An edge-to-edge Deployment Model for Pre-Congestion Notification: Admission Control over a DiffServ Region", draft-briscoe-tsvwg-cl-architecture-04 (work in progress), October 2006.

[I-D.charny-pcn-comparison]
Charny, A., "Comparison of Proposed PCN Approaches", draft-charny-pcn-comparison-00 (work in progress), November 2007.

[I-D.ietf-pcn-architecture]
Eardley, P., "Pre-Congestion Notification (PCN) Architecture", draft-ietf-pcn-architecture-07 (work in progress), September 2008.

[I-D.ietf-pcn-baseline-encoding]
Moncaster, T., Briscoe, B., and M. Menth, "Baseline Encoding and Transport of Pre-Congestion Information", draft-ietf-pcn-baseline-encoding-01 (work in progress),

October 2008.

[I-D.ietf-tsvwg-admitted-realtime-dscp]
Baker, F., Polk, J., and M. Dolly, "DSCPs for Capacity-Admitted Traffic", draft-ietf-tsvwg-admitted-realtime-dscp-04 (work in progress), February 2008.

[I-D.moncaster-pcn-3-state-encoding]
Moncaster, T., Briscoe, B., and M. Menth, "A three state extended PCN encoding scheme", draft-moncaster-pcn-3-state-encoding-00 (work in progress), June 2008.

[Menth] "Menth", 2008, <http://www3.informatik.uni-wuerzburg.de/staff/menth/Publications/Menth08-PCN-Comparison.pdf>.

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

[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.

[RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, January 2008.

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 [I-D.charny-pcn-comparison] and [Menth].

There is no attempt to optimise the algorithms. It implements the metering and marking functions 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.

A.1. Threshold metering and marking

A token bucket with the following parameters:

- TBthreshold.PCN-threshold-rate: token rate of token bucket (bits/second)
- TBthreshold.max: depth of token bucket (bits)
- TBthreshold.threshold: marking threshold of token bucket (bits)
- TBthreshold.lastUpdate: time the token bucket was last updated (seconds)
- TBthreshold.fill: 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 are the parameters:

- now: the current time (seconds)

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

- TBthreshold.fill = min(TBthreshold.max, TBthreshold.fill + (now - TBthreshold.lastUpdate) * TBthreshold.PCN-threshold-rate); // add tokens to token bucket
- TBthreshold.fill = max(0, TBthreshold.fill - packet.size); // remove tokens from token bucket
- if ((TBthreshold.fill < TBthreshold.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
- TBthreshold.lastUpdate = now

A.2. Excess traffic metering and marking

A token bucket with the following parameters:

- TBexcess.PCN-excess-rate: token rate of token bucket (bits/second)
- TBexcess.max: depth of TB in token bucket (bits)
- TBexcess.lastUpdate: time the token bucket was last updated (seconds)
- TBexcess.fill: 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 are the parameters:

- now: the current time (seconds)
- MTU: the maximum transfer unit of the link (or the known maximum size of PCN-packets on the link) (bits)

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

- TBexcess.fill = min(TBexcess.max, TBexcess.fill + (now - TBexcess.lastUpdate) * TBexcess.PCN-excess-rate); // add tokens to token bucket
- if (packet.mark != excess-traffic-marked) then TBexcess.fill = max(0, TBexcess.fill - packet.size); // remove tokens from token bucket, but do not meter packets that are already excess-traffic-marked
- if (TBexcess.fill < MTU) then packet.mark = excess-traffic-marked; // do (packet size independent) excess traffic marking
- TBthreshold.lastUpdate = now

Appendix B. Implementation notes

Note: This Appendix is informative, not normative. It comments on Section 2.

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 [I-D.ietf-pcn-architecture]. But if there is competing-non-PCN-traffic, then there needs to be:

1. a mechanism to limit it, for example:

   * limiting 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 police (traffic condition) the competing-non-PCN-traffic on each link, ie drop competing-non-PCN-packets in excess of some rate.

   * policing of competing-non-PCN-traffic at the PCN-ingress-nodes. For example, as in the DiffServ architecture - although its static traffic conditioning agreements risk a focused overload of traffic from several PCN-ingress-nodes on one link.

   * design: it is known by design 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 by:

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

   * each PCN-node reduces, on its links, the PCN-threshold-rate and PCN-excess-rate, in order to allow 'headroom' for the competing-non-PCN-traffic; also limiting 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.

It is left up to the operator to decide on appropriate action. Traffic conditioning is discussed further in the separate section below.

One specific example of competing-non-PCN-traffic occurs if the PCN-compatible Diffserv codepoint is the Voice-admit codepoint, and there is voice-admit traffic in the PCN-domain.

Another example would occur if there was more than one PCN-compatible Diffserv codepoint in a PCN-domain. For instance, suppose there were 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, eg by the design of the network topology, that some links can never be pre-congested (even in unusual circumstances, eg after the failure of some links). There is then no need to deploy PCN behaviour on those links.

The meter and marker can be implemented on the ingoing or outgoing interface of a PCN-node. It may be that existing hardware can support only one meter and marker per ingoing interface and one per outgoing interface. Then for instance threshold metering and marking could be run on all the ingoing interfaces and excess traffic metering and marking 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.

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 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 this scenario means that the Marking function ignores indications from one of the meters, 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 scope of PCN.

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.

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. Traffic conditioning

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

The objective of traffic conditioning 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. The details will depend on how the router's implementation handles the two sorts of traffic [I-D.ietf-tsvwg-admitted-realtime-dscp]:

- a common queue for PCN-traffic and competing-non-PCN-traffic, and a traffic conditioner for the competing-non-PCN-traffic;

- separate queues. In this 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 traffic conditioning action is to drop packets (which is often called "policing"). Downgrading of packets to a lower priority BA is not allowed (see B.7), since it wold lead to packet mis-ordering. Shaping ("the process of delaying packets" [RFC2475]) is not suitable if the traffic comes from real time applications. In general it is reasonable for competing-non-PCN-traffic to get harsher treatment than PCN-traffic (ie 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 traffic conditioner should not be needed for the PCN-traffic.

Preferential dropping of excess-traffic-marked packets: Section 2.3 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 this avoids over-termination, with the CL/SM edge behaviour, in the event of multiple bottlenecks in the PCN-domain [I-D.charny-pcn-comparison]. A fuller explanation is as follows. The optimal dropping behaviour depends on the particular edge behaviour [Menth]. 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 CL/SM edge behaviour, in the event of multiple bottlenecks in the PCN-domain [I-D.charny-pcn-comparison]). 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; (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 [Menth]. Effectively it means that traffic conditioning 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 [I-D.briscoe-tsvwg-cl-architecture] views as a virtual queue). However the implementation is not standardised.

Section 2.3 defines: "If TBthreshold.fill < TBthreshold.threshold, then the meter indicates to the Marking function that the packet is to be threshold-marked; otherwise it does not." Note that the PCN-packet (that causes the token bucket to cross TBthreshold.threshold) 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. 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 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 TBthreshold.fill, then the precise change in TBthreshold.fill is up to the implementation. A behaviour is functionally equivalent if either precisely the same set of packets is marked, or if the set is shifted by one packet. For

instance, the following should all be considered as "functionally equivalent":

* set TBthreshold.fill = 0 and indicate threshold-mark to the Marking function.

* check whether TBthreshold.fill < TBthreshold.threshold and if it is then indicate threshold-mark to the Marking function; then set TBthreshold.fill = 0.

* leave TBthreshold.fill unaltered and indicate threshold-mark to the Marking function.

- similarly, when the token bucket is very nearly full and a packet arrives large than (TBthreshold.max - TBthreshold.fill), then the precise change in TBthreshold.fill is up to the implementation.

- Note that all packets, even if already marked, are metered by the threshold meter function (unlike the excess traffic meter function - see below) - 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.

As in Section B.3, "functionally equivalent" allows some implementation flexibility 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 [I-D.charny-pcn-comparison]. Note that an implementation could determine whether the packet is already excess-traffic-marked as an integral part of its 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 [Menth]. (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 PCN-marking all packets.) It seems natural to do traffic conditioning 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 is specified as a SHOULD in Section 2.4 ("if the token bucket is within an MTU of being empty, then the meter SHOULD indicate to the Marking function that the packet is to be excess-traffic-marked; MTU means the maximum size of PCN-packets on the link".) Without it, large packets are more likely to be excess-traffic-marked than small packets and this means that, with some edge behaviours, flows with large packets are more likely to be terminated than flows with small packets [I-D.briscoe-tsvwg-byte-pkt-mark] [Menth]. The behaviour is a 'SHOULD', rather than a 'MUST', because packet size independent marking may be slightly harder for some equipment to implement, and the impact of not doing it is moderate (sufficient traffic is terminated, but flows with large packets are more likely to be terminated).

Note that TBexcess.max is independent of TBthreshold.max; TBexcess.fill is independent of TBthreshold.fill (except in that a packet changes 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 MUST NOT traffic condition by downgrading a PCN-packet into a lower priority DiffServ BA.

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 then 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
BT
Adastral Park, Martlesham Heath
Ipswich IP5 3RE
UK

Email: philip.eardley@bt.com