PCN Working Group P. Eardley (Editor)
Internet-Draft BT
Intended status: Standards Track June 25, 2008
Expires: December 27, 2008

Marking behaviour of PCN-nodes
draft-eardley-pcn-marking-behaviour-01

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 first configured rate. Excess traffic marking marks a proportion of PCN-packets, such that the amount marked equals the traffic rate in excess of a second configured rate.

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

[I-D.ietf-pcn-architecture] describes a general architecture for flow admission and termination based on pre-congestion information in order to protect the quality of service of established inelastic flows within a single DiffServ domain. The pre-congestion information consists of specific markings of PCN-packets. The edge nodes of the DiffServ domain read these markings and convert them into flow admission and termination decisions. Overall the 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.

This document standardises the two marking behaviours of PCN-nodes. 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 some configured rate ("PCN-threshold-rate");

- excess traffic marking: whenever the rate of PCN-packets is greater than some 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 how the admission control mechanism limits the PCN-traffic on each link to *roughly* its PCN-threshold-rate and how the flow termination mechanism limits the PCN-traffic on each link to *roughly* its PCN-excess-rate.

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

- Packet classify and condition - decide whether an incoming packet belongs to a PCN-flow or not;

- Condition: drop or downgrade packets if the link is overloaded;

- Threshold meter - determine whether the rate of PCN-packets is greater than the configured PCN-threshold-rate;

- Excess traffic meter - measure by how much the rate of PCN-packets is greater than the configured PCN-excess-rate;

- Mark - actually mark the PCN-packets, if the meter functions indicate to do so;

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. [I-D.moncaster-pcn-baseline-encoding] defines two encoding states (PCN-marked and not PCN-marked), whilst [I-D.draft-moncaster-pcn-3-state-encoding] defines an extended scheme with three encoding states. So in a particular deployment the operator may have three encoding states available (so allowing both threshold marking and excess traffic marking) or may have only two encoding states (so allowing either threshold marking and excess traffic marking). As described in [I-D.ietf-pcn-architecture], flow termination is based on excess traffic marked packets, whilst admission control can be based on either threshold marked or excess traffic marked packets (the former is more accurate, [I-D.draft-charny-pcn-comparison]). This leads to the following four use cases:

1. an operator requires both admission control and flow termination, and has three encoding states available. Then admission control is triggered from PCN-packets that are threshold-marked, and flow termination from PCN-packets that are excess-traffic-marked.

2. an operator requires both admission control and flow termination, and has only two encoding states available. Then both admission control and flow termination are triggered from PCN-packets that are excess-traffic-marked.

3. an operator requires only admission control. Then admission control is triggered from PCN-packets that are threshold-marked and only two encoding states are needed. (Flow termination may be provided by a non PCN mechanism; this is out of scope.)

4. an operator requires only flow termination. Then flow termination is triggered from PCN-packets that are excess-traffic-marked and only two encoding states are needed. (Admission control may be provided by a non PCN mechanism; this is out of scope.)

```
                                       +---------+   Result
                                    +->|Threshold|-------+
                                    |  |  Meter  |       |
                                    |  +---------+       V
         +---------+   +---------+  |                 +------+
         |         |   |         |  |                 |      |    Marked
Packet =>|Classify |==>|Condition|==?================>|Marker|==> Packet
Stream   |         |   |         |  |                 |      |    Stream
         +---------+   +---------+  |                 +------+
                                    |  +---------+       ^
                                    |  | Excess  |       |
                                    +->| Traffic |-------+
                                       |  Meter  |   Result
                                       +---------+
```

Figure 1: Schematic of functions for PCN-marking

1.1. Terminology

In addition to the terminology defined in [I-D.ietf-pcn-architecture] and RFC 2474 [RFC2474], the following terms are defined:

- PCN-traffic: (defined in [draft-pcn-architecture] but need to clarify that PCN-BA is idenitfied by combination of DSCP & ECN fields)

- Other-traffic: traffic that uses the same DS codepoint as PCN-traffic, but a different value in the ECN field; has the same priority as PCN-traffic (in terms of scheduling at PCN-nodes); but is not subject to PCN-marking, nor PCN's admissin control and flow termination mechanisms. Thus PCN-traffic and other-traffic have different per-domain behaviours RFC 3086 [RFC3086]. Note: there may be no other-traffic in a PCN-domain. Note: the term PCN-BA does not include other-traffic (this is a clarification, as the definition of behaviour aggregate in RFC 2474 [RFC2474], RFC 2475 [RFC2475] is somewhat ambiguous in the context of PCN.

- Priority-traffic: traffic that is more important than PCN that shares the same capacity as PCN and is priority scheduled over PCN (perhaps an operator's control messages). Note: there may be no priority-traffic in a PCN-domain.

- Metered-traffic: the collective term for PCN-traffic and (if any) priority-traffic and other-traffic.

- Downgrade: Re-marking a packet, ie changing its DS codepoint, into a lower priority behaviour aggregate, such as best effort or assured forwarding; as a consequence perhaps dropping lower

priority packets.

- <these new terms are not great, but I couldn't find a way of writing the doc without them. I don't think they should be used outside this doc (so would be inclined to keep the terms here & not in [draft-pcn-architecture]).

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 Appendixes supplement it.

2.1. Scope

The functions defined in the following sub-sections SHOULD be implemented on all links in the PCN-domain.

There are three possibilities regarding encoding states:

- three encoding states are available,
  - one for threshold marks,
  - one for excess rate marks
  - one for "not PCN-marked";
- two encoding states are available,
  - one for threshold marks
  - one for "not PCN-marked";
- two encoding states are available,
  - one for excess rate marks
  - one for "not PCN-marked".

The same choice of encoding states MUST be used throughout a PCN-domain.

All metered-traffic MUST be metered by the metering functions specified in Sections 2.3, 2.4 and 2.5 (with the minor exception noted below in Section 2.5). Priority-traffic and other-traffic MUST NOT be PCN-marked (ie only PCN-packets can be PCN-marked).

2.2. Classify function

A packet MUST be classified as a PCN-packet if the value of its DSCP and ECN fields are as standardised in [I-D.moncaster-pcn-baseline-encoding] or [I-D.draft-moncaster-pcn-3-state-encoding], as applicable to the PCN-domain. Otherwise the packet MUST NOT be classified as a PCN-packet.

A packet MUST be classified as an other-traffic packet if it uses the same DSCP as PCN-traffic, but a different value in the ECN field.

A packet MUST be classified as a priority-traffic packet if it shares the same capacity as PCN-traffic and other-traffic and is priority scheduled over them.

2.3. Traffic conditioning function

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

o metering all metered-traffic to determine if the level of metered-traffic is sufficiently high to overload the PCN behaviour aggregate(s). (According to RFC 2475 [RFC2475] metering is "the process of measuring the temporal properties (eg rate) of a traffic stream".

o if the level of metered-traffic is sufficiently high, then do one or more of the following:

   * drop PCN-packets;

   * downgrade PCN-packets;

   * drop other-packets;

   * downgrade other-packets.

   * <you might argue that other-packets should get harsher treatment since they're not subject to PCN's adm & termination control, only subject to the weaker DiffServ style static TCAs at the PCN-ingress-node>

If PCN-packets are dropped (or downgraded) then:

o excess-traffic-marked PCN-packets SHOULD be preferentially dropped (downgraded);

o PCN-packets that are dropped (downgraded) SHOULD NOT be metered by the Excess traffic Meter.

In addition, PCN-ingress-nodes MUST police PCN-traffic by:

o metering PCN-packets that are part of a previously admitted PCN-flow, to check that it keeps to the agreed rate or flowspec (eg RFC 1633 [RFC1633] for a microflow, and its NSIS equivalent).

o checking that any packets received that demand PCN treatment do indeed belong to a previously admitted flow.

o dropping or downgrading packets that fail the above checks.

In addition, PCN-ingress-nodes MUST police other-traffic by:

o metering other-traffic to check that it meeds its traffic conditioning agreement, which is the parameters of the traffic that will be accepted from a customer. Typically it is statically defined as part of the subscription-time service level agreement, as in the DiffServ architecture RFC 2475 [RFC2475]

o dropping or downgrading packets that fail the above check.

In addition, an operator MAY measure the amount of traffic entering (or leaving) its network for accounting reasons. Consideration is out of scope of this document.

2.4. Threshold meter function

The Threshold Meter MUST have behaviour that is 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 TB1.fill. Tokens are added at the PCN-threshold-rate, to a maximum value TB1.max. Tokens are removed equal to the size in bits of the metered-packet, to a minimum TB1.fill=0.

The token bucket has a configured token bucket depth (between 0 and TB1.max), termed TB1.threshold. If TB1.fill < TB1.threshold, then the meter indicates to the Marking function that the packet is to be threshold-marked; otherwise it does not.

2.5. Excess traffic meter function

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

o If the packet is already excess-traffic-marked;

o If this PCN-node drops (downgrades) the packet because the link is overloaded.

Otherwise it is metered by the Excess traffic Meter.

The Excess traffic Meter MUST have behaviour that is 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 TB2.fill. Tokens are added at the PCN-excess-rate, to a maximum value TB2.max. Tokens are removed equal to the size in bits of the metered-packet, to a minimum TB2-fill=0. The PCN-excess-rate is greater than (or equal to) the PCN-threshold-rate.

If the token bucket is empty (TB2.fill = 0), then the meter indicates to the Marking function that the packet is to be excess-traffic-marked. 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. Otherwise the meter does not indicate marking.

2.6. Marking function

If the packet is not a PCN-packet, then it MUST NOT be marked. A PCN-packet MUST be marked to reflect the metering results by setting its encoding state appropriately, as specified below. The encoding states are defined values of the DSCP and ECN fields, as specified in the appropriate encoding document, [I-D.moncaster-pcn-baseline-encoding] or [I-D.draft-moncaster-pcn-3-state-encoding].

There are three possibilities, depending on how many encoding states are available:

o if three encoding states are available (one for threshold-marked, one for excess-traffic-marked and one for "not PCN-marked") then:

   * the encoding state of a packet that has already been excess-traffic-marked is not altered, whatever the meters indicate;

   * Otherwise:

      + if both meters indicate marking, then the packet is excess-traffic-marked;

+ if the threshold meter indicates marking and the excess traffic meter doesn't, then threshold-marking is applied;

+ if the excess traffic meter indicates marking and the threshold traffic meter doesn't, then excess-traffic-marking is applied;

+ if neither meter indicates marking, then the packet's encoding state is not altered.

- if two encoding states are available (one for threshold-marked and one for "not PCN-marked") then:

  * if the Threshold Meter indicates marking, then the packet is threshold-marked;

  * otherwise the packet's encoding state is not altered.

- if two encoding states are available (one for excess-traffic-marked and one for "not PCN-marked") then:

  * if the Excess traffic Meter indicates marking, then the packet is excess-traffic-marked;

  * otherwise the packet's encoding state is not altered.

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 -00 draft.

Thanks to those who've made comments on this draft: Michael Menth, Joe Babiarz, Anna Charny, Ruediger Geib, Wei Gengyu, Fortune Huang.

All the work by many people in the PCN WG.

6. Changes

6.1. Changes from -00 to -01

- Traffic conditioning extensively re-written.
- New terms defined
- Changes resulting from split of encoding into two drafts, baseline [I-D.moncaster-pcn-baseline-encoding] and extension [I-D.draft-moncaster-pcn-3-state-encoding].
- Minor changes to improve clarity.

7. Authors

Many people need to be added.

8. References

8.1. Normative References

[RFC1633] Braden, B., Clark, D., and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994.

[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.

[RFC3086] Nichols, K. and B. Carpenter, "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification", RFC 3086, April 2001.

8.2. Informative References

[I-D.draft-briscoe-tsvwg-byte-pkt-mark]
"Baseline Encoding and Transport of Pre-Congestion Information", February 2008, <http://www.ietf.org/internet-drafts/draft-briscoe-tsvwg-byte-pkt-mark-02.txt>.

[I-D.draft-charny-pcn-comparison]
"Pre-Congestion Notification Using Single Marking for Admission and Termination", November 2007, <http://www.watersprings.org/pub/id/draft-charny-pcn-comparison-00.txt>.

[I-D.draft-moncaster-pcn-3-state-encoding]
"Baseline Encoding and Transport of Pre-Congestion Information", June 2008, <http://www.ietf.org/internet-drafts/draft-moncaster-pcn-3-state-encoding-00.txt>.

[I-D.ietf-pcn-architecture]
"Pre-Congestion Notification Architecture", February 2008, <http://www.ietf.org/internet-drafts/draft-ietf-pcn-architecture-03.txt>.

[I-D.moncaster-pcn-baseline-encoding]
"Baseline Encoding and Transport of Pre-Congestion Information", June 2008, <http://www.ietf.org/internet-drafts/draft-moncaster-pcn-baseline-encoding-01.txt>.

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

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.draft-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 PCN-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:

- o TB1.PCN-threshold-rate: token rate of token bucket (bits/second)
- o TB1.max: depth of token bucket (bits)
- o TB1.threshold: marking threshold of token bucket (bits)
- o TB1.lastUpdate: time the token bucket was last updated (seconds)
- o TB1.fill: amount of tokens in token bucket (bits)

A PCN-packet has the following parameters:

- o packet.size: the size of the PCN-packet (bits)
- o packet.mark: the PCN encoding state of the packet

In addition there are the parameters:

- o now: the current time (seconds)

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

- o TB1.fill = min(TB1.max, TB1.fill + (now - TB1.lastUpdate) * TB1.PCN-threshold-rate); // add tokens to token bucket
- o TB1.fill = max(0, TB1.fill - packet.size); // remove tokens from token bucket
- o if ((TB1.fill < TB1.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
- o TB1.lastUpdate = now

A.2. Excess traffic metering and marking

A token bucket with the following parameters:

- o TB2.PCN-excess-rate: token rate of token bucket (bits/second)
- o TB2.max: depth of TB in token bucket (bits)

o TB2.lastUpdate: time the token bucket was last updated (seconds)

o TB2.fill: amount of tokens in token bucket (bits)

A PCN-packet has the following parameters:

o packet.size: the size of the PCN-packet (bits)

o packet.mark: the PCN encoding state of the packet

In addition there are the parameters:

o now: the current time (seconds)

o 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:

o TB2.fill = min(TB2.max, TB2.fill + (now - TB2.lastUpdate) * TB2.PCN-excess-rate); // add tokens to token bucket

o if (packet.mark != excess-traffic-marked) then TB2.fill = max(0, TB2.fill - packet.size); // remove tokens from token bucket, but do not meter packets that are already excess-traffic-marked

o if (TB2.fill < MTU) then packet.mark = excess-traffic-marked; // do (packet size independent) excess traffic marking

o TB1.lastUpdate = now

Appendix B. Implementation notes

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

B.1. 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 implement 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 both the meters are still implemented, in order to ease compatibility between equipment and remove a configuration option and associated complexity. Although this means that the Marking function ignores indications from one of the meters, they might 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.2. Classify

Traffic that has a higher DiffServ priority than PCN, but shares the same capacity, is metered as though it were PCN-traffic but cannot be PCN-marked. This means that 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.

Similar remarks can be made with respect to other-traffic.

B.3. Traffic conditioning

The objective of traffic conditioning is to minimise the queueing delay suffered by metered-traffic at a PCN-node, since PCN-traffic (and other-traffic) is expected to be inelastic traffic generated by real time applications. "Overload" therefore means breaking this objective. In practice it would be defined as exceeding a specific traffic profile, typically based on a token bucket. If both PCN-traffic and other-traffic is present then the details will depend on how the router's implementation handles the two sorts of traffic, for example it could have:

- a common traffic conditioner and a common queue for PCN-traffic and other-traffic;
- separate traffic conditioners but a common queue;
- separate traffic conditioners and separate queues.

By conditioning traffic to a lower rate than the queue(s) can schedule traffic, the number of packets in the queue(s) can be minimised.

The choice of whether to drop or downgrade packets is left to the operator. For example, if the traffic is expected to be voice then dropping is simple and a small amount of dropping doesn't have much audible effect. But the dropping of a video I-frame will lead to a significant impact. Downgrading needs to be done carefully to avoid re-ordering traffic.

In [RFC2475] shaping is given as another possible action ("the process of delaying packets"). However, this is not suitable here as the traffic is expected to come from real time applications.

Preferential dropping of excess-traffic-marked packets: Section 2.2 specifies: "If the level of metered-traffic is sufficiently high, then ... if PCN-packets are dropped (or downgraded) then: excess-traffic-marked PCN-packets SHOULD be preferentially dropped (downgraded)". This avoids over-termination, with the CL/SM edge behaviour, in the event of multiple bottlenecks in the PCN-domain [I-D.draft-charny-pcn-comparison].

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.

<should we leave it this open or give some options, eg: definitely drop an excess-traffic-marked packet or drop with a higher probability; or, if there are no excess rate marked PCN-packets available, drop any PCN-packet, drop the next excess-traffic-marked PCN-packet>

Section 2.2 also specifies: "PCN-packets that are dropped (downgraded) SHOULD NOT be metered by the Excess traffic Meter." This avoids over-termination, with the CL/SM edge behaviour, in the event of multiple bottlenecks [I-D.draft-charny-pcn-comparison]. Effectively it means that traffic conditioning should be done before the meter functions - which is natural.

B.4. Threshold metering

The description is in terms of a 'token bucket with threshold', however the implementation is not standardised. For example, it could equally well be implemented as a virtual queue [I-D.ietf-pcn-architecture].

The behaviour must be functionally equivalent to the description above. "Functionally equivalent" is intended to allow implementation freedom over matters such as:

<is this list helpful? accurate? trying to clarify that there is some implementation freedom here>

- o whether tokens are added to the token bucket at regular time intervals or only when a packet is processed

- o 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.

- o when the token bucket is very nearly empty and a packet arrives larger than TB1.fill, then the precise change in TB1.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 TB1.fill = 0 and indicate threshold-mark to the Marking function.

  - check whether TB1.fill < TB1.threshold and if it is then indicate threshold-mark to the Marking function; then set TB1.fill = 0.

  - leave TB1.fill unaltered and indicate threshold-mark to the Marking function.

- o similarly, when the token bucket is very nearly full and a packet arrives large than (TB1.max - TB1.fill), then the precise change in TB1.fill is up to the implementation.

- o 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. The threshold meter

B.5. 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.

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.draft-briscoe-tsvwg-byte-pkt-mark] [Menth].

Section 2.4 specifies: "A packet SHOULD NOT be metered (by this excess traffic meter function) ... If the packet is already excess-traffic-marked". 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.draft-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.

Section 2.4 specifies: "A packet SHOULD NOT be metered (by this excess traffic meter function) ... If this PCN-node drops (downgrades) the packet because the link is overloaded." 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.)

Note that TB2.max is independent of TB1.max; TB2.fill is independent of TB1.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.6. Marking

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

[I-D.moncaster-pcn-baseline-encoding] specifies two encoding states and [I-D.draft-moncaster-pcn-3-state-encoding] specifies three encoding states. In some environments encoding states may be scarce, for example MPLS, and then only two encoding states may be preferable.

Section 2.6 states: "if three encoding states are available ... if the threshold meter indicates marking and the excess traffic meter doesn't, then threshold-marking is applied; if the excess traffic

meter indicates marking and the threshold traffic meter doesn't, then excess-traffic-marking is applied". Normally this means that the Threshold Meter indicates marking and the Excess traffic Meter doesn't. However, the reverse is possible for a short time - because the meters react at different speeds when the traffic rate changes.

Author's Address

Philip Eardley +++
BT
Adastral Park, Martlesham Heath
Ipswich IP5 3RE
UK

Email: philip.eardley@bt.com

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).