                Pre-Congestion Notification Architecture
                      draft-ietf-pcn-architecture-02

Status of this Memo

Copyright Notice

Abstract

   The purpose of this document is to describe a general architecture
   for flow admission and termination based on aggregated pre-congestion
   information in order to protect the quality of service of established
   inelastic flows within a single DiffServ domain.

Status

Table of Contents

Eardley (Editor)   Expires May 22, 2008   [Page 3]

1. Introduction

   The purpose of this document is to describe a general architecture
   for flow admission and termination based on aggregated (pre-)
   congestion information in order to protect the quality of service of
   flows within a DiffServ domain [RFC2475]. This document defines an
   architecture for implementing two mechanisms to protect the quality
   of service of established inelastic flows within a single DiffServ
   domain, where all boundary and interior nodes are PCN-enabled and
   trust each other for correct PCN operation. Flow admission control
   determines whether a new flow should be admitted and protects the QoS
   of existing PCN-flows in normal circumstances, by avoiding congestion
   occurring. However, in abnormal circumstances, for instance a
   disaster affecting multiple nodes and causing traffic re-routes, then
   the QoS on existing PCN-flows may degrade even though care was
   exercised when admitting those flows before those circumstances.
   Therefore we also propose a mechanism for flow termination, which
   removes enough traffic in order to protect the QoS of the remaining
   PCN-flows.

   As a fundamental building block to enable these two mechanisms, PCN-
   interior-nodes generate, encode and transport pre-congestion
   information towards the PCN-egress-nodes. Two rates, a PCN-lower-
   rate and a PCN-upper-rate, can be associated with each link of the
   PCN-domain. Each rate is used by a marking behaviour (specified in
   another document) that determines how and when a number of PCN-
   packets are marked, and how the markings are encoded in packet
   headers. PCN-egress-nodes make measurements of the packet markings
   and send information as necessary to the nodes that make the decision

about which PCN-flows to accept/reject or terminate, based on this
information. Another document will describe the decision-making
behaviours. 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 the queue; the admission control mechanism
limits the PCN-traffic on each link to *roughly* its PCN-lower-rate
and the flow termination mechanism limits the PCN-traffic on each
link to *roughly* its PCN-upper-rate.

We believe that the key benefits of the PCN mechanisms described in
this document are that they are simple, scalable, and robust because:

o Per flow state is only required at the PCN-ingress-nodes
   ("stateless core"). This is required for policing purposes (to
   prevent non-admitted PCN traffic from entering the PCN-domain) and
   so on. It is not generally required that other network entities
   are aware of individual flows (although they may be in particular
   deployment scenarios).

   o Admission control is resilient: PCN's QoS is decoupled from the
      routing system; hence in general admitted flows can survive
      capacity, routing or topology changes without additional
      signalling, and they don't have to be told (or learn) about such
      changes. The PCN-lower-rates can be chosen small enough that
      admitted traffic can still be carried after a rerouting in most
      failure cases. This is an important feature as QoS violations in
      core networks due to link failures are more likely than QoS
      violations due to increased traffic volume [Iyer].

   o The PCN-marking behaviours only operate on the overall PCN-traffic
      on the link, not per flow.

   o The information of these measurements is signalled to the PCN-
      egress-nodes by the PCN-marks in the packet headers. No
      additional signalling protocol is required for transporting the
      PCN-marks. Therefore no secure binding is required between data
      packets and separate congestion messages.

   o The PCN-egress-nodes make separate measurements, operating on the
      overall PCN-traffic, for each PCN-ingress-node, ie not per flow.
      Similarly, signalling by the PCN-egress-node of PCN-feedback-

information (which is used for flow admission and termination
decisions) is at the granularity of the ingress-egress-aggregate.

o The admitted PCN-load is controlled dynamically. Therefore it
   adapts as the traffic matrix changes, and also if the network
   topology changes (eg after a link failure). Hence an operator can
   be less conservative when deploying network capacity, and less
   accurate in their prediction of the PCN-traffic matrix.

o The termination mechanism complements admission control. It
   allows the network to recover from sudden unexpected surges of
   PCN-traffic on some links, thus restoring QoS to the remaining
   flows. Such scenarios are expected to be rare but not impossible.
   They can be caused by large network failures that redirect lots of
   admitted PCN-traffic to other links, or by malfunction of the
   measurement-based admission control in the presence of admitted
   flows that send for a while with an atypically low rate and then
   increase their rates in a correlated way.

o The PCN-upper-rate may be set below the maximum rate that PCN-
   traffic can be transmitted on a link, in order to trigger
   termination of some PCN-flows before loss (or excessive delay) of
   PCN-packets occurs, or to keep the maximum PCN-load on a link
   below a level configured by the operator.

o Provisioning of the network is decoupled from the process of
   adding new customers. By contrast, with the DiffServ architecture
   [RFC2475] operators rely on subscription-time Service Level
   Agreements that statically define the parameters of the traffic
   that will be accepted from a customer, and so the operator has to
   run the provisioning process each time a new customer is added to
   check that the Service Level Agreement can be fulfilled. PCN does
   not use RFC2475-style traffic conditioning.

Operators of networks will want to use the PCN mechanisms in various
arrangements, for instance depending on how they are performing
admission control outside the PCN-domain (users after all are
concerned about QoS end-to-end), what their particular goals and
assumptions are, and so on. Several deployment models are possible:

o An operator may choose to deploy either admission control or flow
   termination or both (see [Section 4.3](#)).

o IntServ over DiffServ [[RFC2998](#)]. The DiffServ region is PCN-
   enabled, RSVP signalling is used end-to-end and the PCN-domain is
   a single RSVP hop, ie only the PCN-boundary-nodes process RSVP
   messages. Outside the PCN-domain RSVP messages are processed on
   each hop. This is described in
   [[I-D.briscoe-tsvwg-cl-architecture](#)]

o RSVP signalling is originated and/or terminated by proxies, with
   application-layer signalling between the end user and the proxy.
   For instance SIP signalling with a home hub.

o Similar to previous bullets but NSIS signalling is used instead of
   RSVP.

o NOTE: Consideration of signalling extensions for specific
   protocols is outside the scope of the PCN WG, however it will
   produce a "Requirements for signalling" document as potential
   input for the appropriate WGs.

o Depending on the deployment scenario, the decision-making
   functionality (about flow admission and termination) could reside
   at the PCN-ingress-nodes or PCN-egress-nodes or at some central
   control node in the PCN-domain. NOTE: The Charter restricts us:
   the decision-making functionality is at the PCN-boundary-nodes.

o If the operator runs both the access network and the core network,
   one deployment scenario is that only the core network uses PCN
   admission control but per microflow policing is done at the
   ingress to the access network and not at the PCN-ingress-node.
   Note: to aid readability, the rest of this draft assumes that

   policing is done by the PCN-ingress-nodes.

o There are several PCN-domains on the end-to-end path, each
   operating PCN mechanisms independently. NOTE: The Charter
   restricts us to considering a single PCN-domain. A possibility
   after re-chartering is to consider that the PCN-domain encompasses
   several autonomous systems that don't trust each other (ie weakens
   Assumption 1 about trust, see [Section 3.1](#))

o   The PCN-domain extends to the end users. NOTE: This isn't
    necessarily outside the Charter because it may not break
    Assumption 3 (aggregation see later) if it's known there's
    sufficient aggregation at any bottleneck, and it doesn't
    necessarily break Assumption 1 (trust), because in some
    environments, eg corporate, the end user may have a controlled
    configuration and so be trusted. The scenario is described in
    [I-D.babiarz-pcn-sip-cap]. A variant is that the PCN-domain
    extends out as far as the LAN edge switch.

o   Pseudowire: PCN may be used as a congestion avoidance mechanism
    for edge to edge pseudowire emulations
    [I-D.ietf-pwe3-congestion-frmwk]. NOTE: Specific consideration of
    pseudowires is not in the PCN WG Charter.

o   MPLS: [RFC3270] defines how to support the DiffServ architecture
    in MPLS networks. [I-D.ietf-tsvwg-ecn-mpls] describes how to add
    PCN for admission control of microflows into a set of MPLS
    aggregates (Multi-protocol label switching). PCN-marking is done
    in MPLS's EXP field.

o   Similarly, it may be possible to extend PCN into Ethernet
    networks, where PCN-marking is done in the Ethernet header. NOTE:
    Specific consideration of this extension is outside the IETF's
    remit.

2. Terminology

o   PCN-domain: a PCN-capable domain; a contiguous set of PCN-enabled
    nodes that perform DiffServ scheduling; the compete set of PCN-
    nodes whose PCN-marking can in principle influence decisions about
    flow admission and termination for the PCN-domain, including the
    PCN-egress-nodes which measure these PCN-marks.

o   PCN-boundary-node: a PCN-node that connects one PCN-domain to a
    node either in another PCN-domain or in a non PCN-domain.

o   PCN-interior-node: a node in a PCN-domain that is not a PCN-

boundary-node.

o PCN-node: a PCN-boundary-node or a PCN-interior-node

o PCN-egress-node: a PCN-boundary-node in its role in handling
   traffic as it leaves a PCN-domain.

o PCN-ingress-node: a PCN-boundary-node in its role in handling
   traffic as it enters a PCN-domain.

o PCN-traffic: A PCN-domain carries traffic of different DiffServ
   classes [RFC4594]. Those using the PCN mechanisms are called PCN-
   classes (collectively called PCN-traffic) and the corresponding
   packets are PCN-packets. The same network may carry traffic using
   other DiffServ classes.

o Ingress-egress-aggregate: The collection of PCN-packets from all
   PCN-flows that travel in one direction between a specific pair of
   PCN-boundary-nodes.

o PCN-lower-rate: a reference rate configured for each link in the
   PCN-domain, which is lower than the PCN-upper-rate. It is used by
   a marking behaviour that determines whether a packet should be
   PCN-marked with a first encoding.

o PCN-upper-rate: a reference rate configured for each link in the
   PCN-domain, which is higher than the PCN-lower-rate. It is used
   by a marking behaviour that determines whether a packet should be
   PCN-marked with a second encoding.

o Threshold-marking: a PCN-marking behaviour such that all PCN-
   traffic is marked if the PCN-traffic exceeds a particular rate
   (either the PCN-lower-rate or PCN-upper-rate). NOTE: The
   definition reflects the overall intent rather than its
   instantaneous behaviour, since the rate measured at a particular
   moment depends on the behaviour, its implementation and the
   traffic's variance as well as its rate.

o Excess-rate-marking: a PCN-marking behaviour such that the amount
   of PCN-traffic that is PCN-marked is equal to the amount that
   exceeds a particular rate (either the PCN-lower-rate or PCN-upper-
   rate). NOTE: The definition reflects the overall intent rather
   than its instantaneous behaviour, since the rate measured at a
   particular moment depends on the behaviour, its implementation and
   the traffic's variance as well as its rate.

   o Pre-congestion: a condition of a link within a PCN-domain in which
      the PCN-node performs PCN-marking, in order to provide an "early
      warning" of potential congestion before there is any significant
      build-up of PCN-packets in the real queue.

   o PCN-marking: the process of setting the header in a PCN-packet
      based on defined rules, in reaction to pre-congestion.

   o PCN-feedback-information: information signalled by a PCN-egress-
      node to a PCN-ingress-node or central control node, which is
      needed for the flow admission and flow termination mechanisms.

3. Assumptions and constraints on scope

   The PCN WG's charter restricts the initial scope by a set of
   assumptions. Here we list those assumptions and explain them.

   1. these components are deployed in a single DiffServ domain, within
       which all PCN-nodes are PCN-enabled and trust each other for
       truthful PCN-marking and transport

   2. all flows handled by these mechanisms are inelastic and
       constrained to a known peak rate through policing or shaping

   3. the number of PCN-flows across any potential bottleneck link is
       sufficiently large that stateless, statistical mechanisms can be
       effective. To put it another way, the aggregate bit rate of PCN-
       traffic across any potential bottleneck link needs to be
       sufficiently large relative to the maximum additional bit rate
       added by one flow

   4. PCN-flows may have different precedence, but the applicability of
       the PCN mechanisms for emergency use (911, GETS, WPS, MLPP, etc.)
       is out of scope

   After completion of the initial phase, the PCN WG may re-charter to
   develop solutions for specific scenarios where some of these
   restrictions are not in place. It may also re-charter to consider
   applying the PCN mechanisms to additional deployment scenarios. One
   possible example is where a single PCN-domain encompasses several
   DiffServ domains that don't trust each other (perhaps by using a
   mechanism like re-ECN, [I-D.briscoe-re-pcn-border-cheat]. The WG may
   also re-charter to investigate additional response mechanisms that
   act on (pre-)congestion information. One example could be flow-rate
   adaptation by elastic applications (rather than flow admission or

termination). The details of these work items are outside the scope
of the initial phase, but the WG may consider their requirements in

order to design components that are sufficiently general to support
such extensions in the future. The working assumption is that the
standards developed in the initial phase should not need to be
modified to satisfy the solutions for when these restrictions are
removed.

3.1. Assumption 1: Trust - controlled environment

We assume that the PCN-domain is a controlled environment, i.e. all
the nodes in a PCN-domain run PCN and trust each other. There are
several reasons for proposing this assumption:

o The PCN-domain has to be encircled by a ring of PCN-boundary-
   nodes, otherwise PCN-packets could enter the PCN-domain without
   being subject to admission control, which would potentially
   destroy the QoS of existing flows.

o Similarly, a PCN-boundary-node has to trust that all the PCN-nodes
   are doing PCN-marking. A non PCN-node wouldn't be able to alert
   that it is suffering pre-congestion, which potentially would lead
   to too many PCN-flows being admitted (or too few being
   terminated). Worse, a rogue node could perform various attacks,
   as discussed in the Security Considerations section.

One way of assuring the above two points is that the entire PCN-
domain is run by a single operator. Another possibility is that
there are several operators but they trust each other to a sufficient
level, in their handling of PCN-traffic.

Note: All PCN-nodes need to be trustworthy. However if it's known
that an interface cannot become pre-congested then it's not strictly
necessary for it to be capable of PCN-marking. But this must be
known even in unusual circumstances, eg after the failure of some
links.

3.2. Assumption 2: Real-time applications

We assume that any variation of source bit rate is independent of the
level of pre-congestion. We assume that PCN-packets come from real

time applications generating inelastic traffic [Shenker] like voice
and video requiring low delay, jitter and packet loss, for example
the Controlled Load Service, [RFC2211], and the Telephony service
class, [RFC4594]. This assumption is to help focus the effort where
it looks like PCN would be most useful, ie the sorts of applications
where per flow QoS is a known requirement. For instance, the impact
of this assumption would be to guide simulations work.

3.3. Assumption 3: Many flows and additional load

   We assume that there are many flows on any bottleneck link in the
   PCN-domain (or, to put it another way, the aggregate bit rate of PCN-
   traffic across any potential bottleneck link is sufficiently large
   relative to the maximum additional bit rate added by one flow).
   Measurement-based admission control assumes that the present is a
   reasonable prediction of the future: the network conditions are
   measured at the time of a new flow request, however the actual
   network performance must be OK during the call some time later. One
   issue is that if there are only a few variable rate flows, then the
   aggregate traffic level may vary a lot, perhaps enough to cause some
   packets to get dropped. If there are many flows then the aggregate
   traffic level should be statistically smoothed. How many flows is
   enough depends on a number of things such as the variation in each
   flow's rate, the total rate of PCN-traffic, and the size of the
   "safety margin" between the traffic level at which we start
   admission-marking and at which packets are dropped or significantly
   delayed.

   We do not make explicit assumptions on how many PCN-flows are in each
   ingress-egress-aggregate. Performance evaluation work may clarify
   whether it is necessary to make any additional assumption on
   aggregation at the ingress-egress-aggregate level.

3.4. Assumption 4: Emergency use out of scope

   PCN-flows may have different precedence, but the applicability of the
   PCN mechanisms for emergency use (911, GETS, WPS, MLPP, etc) is out
   of scope for consideration by the PCN WG.

3.5. Other assumptions

As a consequence of Assumption 2 above, it is assumed that PCN-
marking is being applied to traffic scheduled with the expedited
forwarding per-hop behaviour, [RFC3246], or traffic with similar
characteristics.

The following two assumptions apply if the PCN WG decides to encode
PCN-marking in the ECN-field.

o It is assumed that PCN-nodes do not perform ECN, [RFC3168], on
   PCN-packets.

o If a packet that is part of a PCN-flow arrives at a PCN-ingress-
   node with its CE (Congestion experienced) codepoint set, then we
   assume that the PCN-ingress-node drops the packet. After its
   initial Charter is complete, the WG may decide to work on a

   mechanism (such as through a signalling extension) that enables
   ECN-marking to be carried transparently across the PCN-domain.

4. High-level functional architecture

   The high-level approach is to split functionality between:

   o PCN-interior-nodes 'inside' the PCN-domain, which monitor their
      own state of pre-congestion on each outgoing interface and mark
      PCN-packets if appropriate. They are not flow-aware, nor aware of
      ingress-egress-aggregates. The functionality is also done by PCN-
      ingress-nodes for their outgoing interfaces (ie those 'inside' the
      PCN-domain).

   o PCN-boundary-nodes at the edge of the PCN-domain, which control
      admission of new PCN-flows and termination of existing PCN-flows,
      based on information from PCN-interior-nodes. This information is
      in the form of the PCN-marked data packets (which are intercepted
      by the PCN-egress-nodes) and not signalling messages. Generally
      PCN-ingress-nodes are flow-aware and in several deployment
      scenarios PCN-egress-nodes will also be flow aware.

   The aim of this split is to keep the bulk of the network simple,
   scalable and robust, whilst confining policy, application-level and

security interactions to the edge of the PCN-domain. For example the
lack of flow awareness means that the PCN-interior-nodes don't care
about the flow information associated with the PCN-packets that they
carry, nor do the PCN-boundary-nodes care about which PCN-interior-
nodes its flows traverse.

4.1. Flow admission

At a high level, flow admission control works as follows. In order
to generate information about the current state of the PCN-domain,
each PCN-node PCN-marks packets if it is "pre-congested". Exactly
how a PCN-node decides if it is "pre-congested" (the algorithm) and
exactly how packets are "PCN-marked" (the encoding) will be defined
in a separate standards-track document, but at a high level it is
expected to be as follows:

o   the algorithm: a PCN-node meters the amount of PCN-traffic on each
    one of its outgoing links. The measurement is made as an
    aggregate of all PCN-packets, and not per flow. The algorithm has
    a configured parameter, PCN-lower-rate. As the amount of PCN-
    traffic exceeds the PCN-lower-rate, then PCN-packets are PCN-
    marked. See NOTE below for more explanation.

o   the encoding: a PCN-node PCN-marks a PCN-packet (with a first
    encoding) by setting fields in the header to specific values. It
    is expected that the ECN and/or DSCP fields will be used.

NOTE: Two main categories of algorithm have been proposed: if the
algorithm uses threshold-marking then all PCN-packets are marked if
the current rate exceeds the PCN-lower-rate, whereas if the algorithm
uses excess-rate-marking the amount marked is equal to the amount in
excess of the PCN-lower-rate. However, note that this description
reflects the overall intent of the algorithm rather than its
instantaneous behaviour, since the rate measured at a particular
moment depends on the detailed algorithm, its implementation (eg
virtual queue, token bucket...) and the traffic's variance as well as
its rate (eg marking may well continue after a recent overload even
after the instantaneous rate has dropped).

The PCN-boundary-nodes monitor the PCN-marked packets in order to
extract information about the current state of the PCN-domain. Based

on this monitoring, a decision is made about whether to admit a
prospective new flow. Exactly how the admission control decision is
made will be defined separately (at the moment the intention is that
there will be one or more informational-track RFCs), but at a high
level two approaches have been proposed to date:

o   the PCN-egress-node measures (possibly as a moving average) the
    fraction of the PCN-traffic that is PCN-marked. The fraction is
    measured for a specific ingress-egress-aggregate. If the fraction
    is below a threshold value then the new flow is admitted.

o   if the PCN-egress-node receives one (or several) PCN-marked
    packets, then a new flow is blocked.

Note that the PCN-lower-rate is a parameter that can be configured by
the operator. It will be set lower than the traffic rate at which
the link becomes congested and the node drops packets. (Hence, by
analogy with ECN we call our mechanism Pre-Congestion Notification.)

Note also that the admission control decision is made for a
particular ingress-egress-aggregate. So it is quite possible for a
new flow to be admitted between one pair of PCN-boundary-nodes,
whilst at the same time another admission request is blocked between
a different pair of PCN-boundary-nodes.

4.2. Flow termination

At a high level, flow termination control works as follows. Each
PCN-node PCN-marks packets in a similar fashion to above. An obvious
approach is for the algorithm to use a second configured parameter,

PCN-upper-rate, and a second header encoding. However there is also
a proposal to use the same rate and the same encoding. Several
approaches have been proposed to date about how to convert this
information into a flow termination decision; at a high level these
are as follows:

o   One approach measures the rate of unmarked PCN-traffic (ie not
    PCN-upper-rate-marked) at the PCN-egress-node, which is the amount
    of PCN-traffic that can actually be supported; the PCN-ingress-
    node measures the rate of PCN-traffic that is destined for this
    specific PCN-egress-node, and hence can calculate the excess

amount that should be terminated.

o   Another approach instead measures the rate of PCN-upper-rate-
     marked traffic and calculates and selects the flows that should be
     terminated.

o   Another approach terminates any PCN-flow with a PCN-upper-rate-
     marked packet. Compared with the approaches above, PCN-marking
     needs to be done at a reduced rate otherwise far too much traffic
     would be terminated.

o   Another approach uses only one sort of marking, which is based on
     the PCN-lower-rate, to decide not only whether to admit more PCN-
     flows but also whether any PCN-flows need to be terminated. It
     assumes that the ratio of the (implicit) PCN-upper-rate and the
     PCN-lower-rate is the same on all links. This approach measures
     the rate of unmarked PCN-traffic at a PCN-egress-node. The PCN-
     ingress-node uses this measurement to compute the implicit PCN-
     upper-rate of the bottleneck link. It then measures the rate of
     PCN-traffic that is destined for this specific PCN-egress-node and
     hence can calculate the amount that should be terminated.

   Since flow termination is designed for "abnormal" circumstances, it
   is quite likely that some PCN-nodes are congested and hence packets
   are being dropped and/or significantly queued. The flow termination
   mechanism must bear this in mind.

   Note also that the termination control decision is made for a
   particular ingress-egress-aggregate. So it is quite possible for
   PCN-flows to be terminated between one pair of PCN-boundary-nodes,
   whilst at the same time none are terminated between a different pair
   of PCN-boundary-nodes.

4.3. Flow admission and flow termination

   Although designed to work together, flow admission and flow
   termination are independent mechanisms, and the use of one does not

---

   require or prevent the use of the other.

   For example, an operator could use just admission control, solving
   heavy congestion (caused by re-routing) by 'just waiting' - as

sessions end, existing microflows naturally depart from the system over time, and the admission control mechanism will prevent admission of new microflows that use the affected links. So the PCN-domain will naturally return to normal operation, but with reduced capacity. The drawback of this approach would be that until PCN-flows naturally depart to relieve the congestion, all PCN-flows as well as lower priority services will be adversely affected. On the other hand, an operator could just rely for admission control on statically provisioned capacity per PCN-ingress-node (regardless of the PCN-egress-node of a flow), as is typical in the hose model of the DiffServ architecture [RFC2475]. Such traffic conditioning agreements can lead to focused overload: many flows happen to focus on a particular link and then all flows through the congested link fail catastrophically. The flow termination mechanism could then be used to counteract such a problem.

A different possibility is to configure only the PCN-lower-rate and hence only do one type of PCN-marking, but generate admission and flow termination responses from different levels of marking. This is suggested in [I-D.charny-pcn-single-marking] which gives some of the pros and cons of this approach.

4.4. Information transport

The transport of pre-congestion information from a PCN-node to a PCN-egress-node is through PCN-markings in data packet headers, no signalling protocol messaging is needed. However, signalling is needed to transport PCN-feedback-information between the PCN-boundary-nodes, for example to convey the fraction of PCN-marked traffic from a PCN-egress-node to the relevant PCN-ingress-node. Exactly what information needs to be transported will be described in the future PCN WG document(s) about the boundary mechanisms. The signalling could be done by an extension of RSVP or NSIS, for instance; protocol work will be done by the relevant WG, but for example [I-D.lefaucheur-rsvp-ecn] describes the extensions needed for RSVP.

4.5. PCN-traffic

The following are some high-level points about how PCN works:

o There needs to be a way for a PCN-node to distinguish PCN-traffic from non PCN-traffic. They may be distinguished using the DSCP field and/or ECN field.

   o The PCN mechanisms may be applied to more than one traffic class
      (which are distinguished by DSCP).

   o There may be traffic that is more important than PCN, perhaps a
      particular application or an operator's control messages. A PCN-
      node may dedicate capacity to such traffic or priority schedule it
      over PCN. In the latter case its traffic needs to contribute to
      the PCN meters.

   o There will be traffic less important than PCN. For instance best
      effort or assured forwarding traffic. It will be scheduled at
      lower priority than PCN, and use a separate queue or queues.
      However, a PCN-node should dedicate some capacity to lower
      priority traffic so that it isn't starved.

   o There may be other traffic with the same priority as PCN-traffic.
      For instance, Expedited Forwarding sessions that are originated
      either without capacity admission or with traffic engineering. In
      [I-D.ietf-tsvwg-admitted-realtime-dscp] the two traffic classes
      are called EF and EF-ADMIT. A PCN-node could either use separate
      queues, or separate policers and a common queue; the draft
      provides some guidance when each is better, but for instance the
      latter is preferred when the two traffic classes are carrying the
      same type of application with the same jitter requirements.

5. Detailed Functional architecture

   This section is intended to provide a systematic summary of the new
   functional architecture in the PCN-domain. First it describes
   functions needed at the three specific types of PCN-node; these are
   data plane functions and are in addition to their normal router
   functions. Then it describes further functionality needed for both
   flow admission control and flow termination; these are signalling and
   decision-making functions, and there are various possibilities for
   where the functions are physically located. The section is split
   into:

   1. functions needed at PCN-interior-nodes

   2. functions needed at PCN-ingress-nodes

   3. functions needed at PCN-egress-nodes

   4. other functions needed for flow admission control

   5. other functions needed for flow termination control

Note: Probing is covered in [Section 7](#).

The section then discusses some other detailed topics:

1. addressing

2. tunnelling

3. fault handling

5.1. PCN-interior-node functions

Each interface of the PCN-domain is upgraded with the following
functionality:

o   Packet classify - decide whether an incoming packet is a PCN-
    packet or not. Another PCN WG document will specify encoding,
    using the DSCP and/or ECN fields.

o   PCN-meter - measure the 'amount of PCN-traffic'. The measurement
    is made as an aggregate of all PCN-packets, and not per flow.

o   PCN-mark - algorithms determine whether to PCN-mark PCN-packets
    and what packet encoding is used (as specified in another PCN WG
    document).

The same general approach of metering and PCN-marking is performed
for both flow admission control and flow termination, however the
algorithms and encoding may be different.

These functions are needed for each interface of the PCN-domain.
They are therefore needed on all interfaces of PCN-interior-nodes,
and on the interfaces of PCN-boundary-nodes that are internal to the
PCN-domain. There may be more than one PCN-meter and marker
installed at a given interface, eg one for admission and one for
termination.

5.2. PCN-ingress-node functions

Each ingress interface of the PCN-domain is upgraded with the

following functionality:

o Packet classify - decide whether an incoming packet is part of a
   previously admitted microflow, by using a filter spec (eg DSCP,
   source and destination addresses and port numbers)

o Police - police, by dropping or re-marking with a non-PCN DSCP,
   any packets received with a DSCP demanding PCN transport that do

   not belong to an admitted flow. Similarly, police packets that
   are part of a previously admitted microflow, to check that the
   microflow keeps to the agreed rate or flowspec (eg RFC1633
   [RFC1633] and NSIS equivalent).

o PCN-colour - set the DSCP field or DSCP and ECN fields to the
   appropriate value(s) for a PCN-packet. The draft about PCN-
   encoding will discuss further.

o PCN-meter - make "measurements of PCN-traffic". Some approaches
   to flow termination require the PCN-ingress-node to measure the
   (aggregate) rate of PCN-traffic towards a particular PCN-egress-
   node.

The first two are policing functions, needed to make sure that PCN-
packets let into the PCN-domain belong to a flow that's been admitted
and to ensure that the flow doesn't go at a faster rate than agreed.
The filter spec will for example come from the flow request message
(outside scope of PCN WG, see [I-D.briscoe-tsvwg-cl-architecture] for
an example using RSVP). PCN-colouring allows the rest of the PCN-
domain to recognise PCN-packets.

5.3. PCN-egress-node functions

Each egress interface of the PCN-domain is upgraded with the
following functionality:

o Packet classify - determine which PCN-ingress-node a PCN-packet
   has come from.

o PCN-meter - make measurements of PCN-traffic. The measurement(s)
   is made as an aggregate (ie not per flow) of all PCN-packets from
   a particular PCN-ingress-node.

o PCN-colour - for PCN-packets, set the DSCP and ECN fields to the
   appropriate values for use outside the PCN-domain.

Another PCN WG document, about boundary mechanisms, will describe
what the "measurements of PCN-traffic" are. This depends on whether
the measurement is targeted at admission control or flow termination.
It also depends on what encoding and PCN-marking algorithms are
specified by the PCN WG.

5.4. Admission control functions

Specific admission control functions can be performed at a PCN-
boundary-node (PCN-ingress-node or PCN-egress-node) or at a
centralised node, but not at normal PCN-interior-nodes. The

functions are:

o Make decision about admission - compare the required "measurements
   of PCN-traffic" (output of the PCN-egress-node's PCN-meter
   function) with some reference level, and hence decide whether to
   admit the potential new PCN-flow. As well as the PCN
   measurements, the decision takes account of policy and application
   layer requirements.

o Communicate decision about admission - signal the decision to the
   node making the admission control request (which may be outside
   the PCN-domain), and to the policer (PCN-ingress-node function)

There are various possibilities for how the functionality can be
distributed (we assume the operator would configure which is used):

o The decision is made at the PCN-egress-node and signalled to the
   PCN-ingress-node

o The decision is made at the PCN-ingress-node, which requires that
   the PCN-egress-node signals to the PCN-ingress-node the fraction
   of PCN-traffic that is PCN-marked (or whatever the PCN WG agrees
   as the required "measurements of PCN-traffic").

o The decision is made at a centralised node, which requires that
   the PCN-egress-node signals its measurements to the centralised

node, and that the centralised node signals to the PCN-ingress-
node about the decision about admission control. It would be
possible for the centralised node to be one of the PCN-boundary-
nodes, when clearly the signalling would sometimes be replaced by
a message internal to the node.

5.5. Flow termination functions

Specific termination control functions can be performed at a PCN-
boundary-node (PCN-ingress-node or PCN-egress-node) or at a
centralised node, but not at normal PCN-interior-nodes. There are
various possibilities for how the functionality can be distributed,
similar to those discussed above in the Admission control section;
the flow termination decision could be made at the PCN-ingress-node,
the PCN-egress-node or at some centralised node. The functions are:

o PCN-meter at PCN-egress-node - make "measurements of PCN-traffic"
   from a particular PCN-ingress-node.

o (if required) PCN-meter at PCN-ingress-node - make "measurements
   of PCN-traffic" being sent towards a particular PCN-egress-node;
   again, this is done for the ingress-egress-aggregate and not per

flow.

o (if required) Communicate "measurements of PCN-traffic" to the
   node that makes the flow termination decision. For example, if
   the PCN-ingress-node makes the decision then communicate the PCN-
   egress-node's measurements to it (as in
   [I-D.briscoe-tsvwg-cl-architecture]).

o Make decision about flow termination - use the "measurements of
   PCN-traffic" to decide which PCN-flow or PCN-flows to terminate.
   The decision takes account of policy and application layer
   requirements.

o Communicate decision about flow termination - signal the decision
   to the node that is able to terminate the flow (which may be
   outside the PCN-domain), and to the policer (PCN-ingress-node
   function).

5.6. Addressing

PCN-nodes may need to know the address of other PCN-nodes:

o   Note: in all cases PCN-interior-nodes don't need to know the
    address of any other PCN-nodes (except as normal their next hop
    neighbours, for routing purposes)

o   in the cases of admission or termination decision by a PCN-
    boundary-node, the PCN-egress-node needs to know the address of
    the PCN-ingress-node associated with a flow, at a minimum so that
    the PCN-ingress-node can be informed to enforce the admission
    decision (and any flow termination decision) through policing.
    The addressing information can be gathered from signalling, for
    example as described for RSVP in [I-D.lefaucheur-rsvp-ecn].
    Another alternative is to use a probe packet that includes as
    payload the address of the PCN-ingress-node. Alternatively, if
    PCN-traffic is always tunnelled across the PCN-domain, then the
    PCN-ingress-node's address is simply the source address of the
    outer packet header; then the PCN-ingress-node needs to learn the
    address of the PCN-egress-node, either by manual configuration or
    by one of the automated tunnel endpoint discovery mechanisms (such
    as signalling or probing over the data route, interrogating
    routing or using a centralised broker).

o   in the cases of admission or termination decision by a central
    control node, the PCN-egress-node needs to be configured with the
    address of the centralised node. In addition, depending on the
    exact deployment scenario and its signalling, the centralised node
    may need to know the addresses of the PCN-ingress-node and PCN-

---

egress-node, the PCN-egress-node may need to know the address of
the PCN-ingress-node, and the PCN-ingress-node may need to know
the address of the centralised node and the PCN-egress-node.
NOTE: Consideration of the centralised case is out of scope of the
initial PCN WG Charter.

5.7. Tunnelling

Tunnels may originate and/or terminate within a PCN-domain. It is
important that the PCN-marking of any packet can potentially
influence PCN's flow admission control and termination - it shouldn't
matter whether the packet happens to be tunnelled at the PCN-node

that PCN-marks the packet, or indeed whether it's decapsulated or
encapsulated by a subsequent PCN-node. This suggests that the
"uniform conceptual model" described in [RFC2983] should be re-
applied in the PCN context. In line with this and the approach of
[RFC4303] and [I-D.briscoe-tsvwg-ecn-tunnel], the following rule is
applied if encapsulation is done within the PCN-domain:

o   any PCN-marking is copied into the outer header

Similarly, in line with the "uniform conceptual model" of [RFC2983]
and the "full-functionality option" of [RFC3168], the following rule
is applied if decapsulation is done within the PCN-domain:

o   if the outer header's marking state is more severe then it is
    copied onto the inner header

o   NB the order of increasing severity is: unmarked; PCN-marking with
    first encoding (ie associated with the PCN-lower-rate); PCN-
    marking with second encoding (ie associated with the PCN-upper-
    rate)

An operator may wish to tunnel PCN-traffic from PCN-ingress-nodes to
PCN-egress-nodes. The PCN-marks shouldn't be visible outside the
PCN-domain, which can be achieved by doing the PCN-colour function
(Section 5.3) after all the other (PCN and tunnelling) functions.
The potential reasons for doing such tunnelling are: the PCN-egress-
node then automatically knows the address of the relevant PCN-
ingress-node for a flow; even if ECMP is running, all PCN-packets on
a particular ingress-egress-aggregate follow the same path. But it
also has drawbacks, for example the additional overhead in terms of
bandwidth and processing.

Potential issues arise for a "partially PCN-capable tunnel", ie where
only one tunnel endpoint is in the PCN domain:

1. The tunnel starts outside a PCN-domain and finishes inside it.
    If the packet arrives at the tunnel ingress with the same
    encoding as used within the PCN-domain to indicate PCN-marking,
    then this could lead the PCN-egress-node to falsely measure pre-
    congestion.

2. The tunnel starts inside a PCN-domain and finishes outside it.
       If the packet arrives at the tunnel ingress already PCN-marked,
       then it will still have the same encoding when it's decapsulated
       which could potentially confuse nodes beyond the tunnel egress.

   In line with the solution for partially capable DiffServ tunnels in
   [2983], the following rules are applied:

   o For case (1), the tunnel egress node clears any PCN-marking on the
      inner header. This rule is applied before the 'copy on
      decapsulation' rule above.

   o For case (2), the tunnel ingress node clears any PCN-marking on
      the inner header. This rule is applied after the 'copy on
      encapsulation' rule above.

   Note that the above implies that one has to know, or figure out, the
   characteristics of the other end of the tunnel as part of setting it
   up.

5.8. Fault handling

   If a PCN-interior-node fails (or one of its links), then lower layer
   protection mechanisms or the regular IP routing protocol will
   eventually re-route round it. If the new route can carry all the
   admitted traffic, flows will gracefully continue. If instead this
   causes early warning of pre-congestion on the new route, then
   admission control based on pre-congestion notification will ensure
   new flows will not be admitted until enough existing flows have
   departed. Re-routing may result in heavy (pre-)congestion, when the
   flow termination mechanism will kick in.

   If a PCN-boundary-node fails then we would like the regular QoS
   signalling protocol to take care of things. As an example
   [I-D.briscoe-tsvwg-cl-architecture] considers what happens if RSVP is
   the QoS signalling protocol. The details for a specific signalling
   protocol are out of scope of the PCN WG, however there is a WG
   Milestone on generic "Requirements for signalling".

6. Design goals and challenges

   Prior work on PCN and similar mechanisms has thrown up a number of
   considerations about PCN's design goals (things PCN should be good
   at) and some issues that have been hard to solve in a fully
   satisfactory manner. Taken as a whole it represents a list of trade-
   offs (it's unlikely that they can all be 100% achieved) and perhaps
   as evaluation criteria to help an operator (or the IETF) decide
   between options.

   The following are key design goals for PCN (based on
   [I-D.chan-pcn-problem-statement]):

   o The PCN-enabled packet forwarding network should be simple,
      scalable and robust

   o Compatibility with other traffic (ie a proposed solution should
      work well when non-PCN traffic is also present in the network)

   o Support of different types of real-time traffic (eg should work
      well with CBR and VBR voice and video sources treated together)

   o Reaction time of the mechanisms should be commensurate with the
      desired application-level requirements (eg a termination mechanism
      needs to terminate flows before significant QoS issues are
      experienced by real-time traffic, and before most users hang up).

   o Compatibility with different precedence levels of real-time
      applications (e.g. preferential treatment of higher precedence
      calls over lower precedence calls, [ITU-MLPP].

   The following are open issues. They are mainly taken from
   [I-D.briscoe-tsvwg-cl-architecture] which also describes some
   possible solutions. Note that some may be considered unimportant in
   general or in specific deployment scenarios or by some operators.

   NOTE: Potential solutions are out of scope for this document.

   o ECMP (Equal Cost Multi-Path) Routing: The level of pre-congestion
      is measured on a specific ingress-egress-aggregate. However, if
      the PCN-domain runs ECMP, then traffic on this ingress-egress-
      aggregate may follow several different paths - some of the paths
      could be pre-congested whilst others are not. There are three
      potential problems:

      1. over-admission: a new flow is admitted (because the pre-
          congestion level measured by the PCN-egress-node is
          sufficiently diluted by unmarked packets from non-congested

paths that a new flow is admitted), but its packets travel
through a pre-congested PCN-node

2. under-admission: a new flow is blocked (because the pre-
   congestion level measured by the PCN-egress-node is
   sufficiently increased by PCN-marked packets from pre-
   congested paths that a new flow is blocked), but its packets
   travel along an uncongested path

3. ineffective termination: flows are terminated, however their
   path doesn't travel through the (pre-)congested router(s).
   Since flow termination is a 'last resort' that protects the
   network should over-admission occur, this problem is probably
   more important to solve than the other two.

o ECMP and signalling: It is possible that, in a PCN-domain running
   ECMP, the signalling packets (eg RSVP, NSIS) follow a different
   path than the data packets - it depends on which fields the ECMP
   algorithm uses. This could matter if the signalling packets are
   used as probes.

o Tunnelling: There are scenarios where tunnelling makes it hard to
   determine the path in the PCN-domain. The problem, its impact and
   the potential solutions are similar to those for ECMP.

o Scenarios with only one tunnel endpoint in the PCN domain may make
   it harder for the PCN-egress-node to gather from the signalling
   messages (eg RSVP, NSIS) the identity of the PCN-ingress-node.

o Bi-Directional Sessions: Many applications have bi-directional
   sessions - hence there are two flows that should be admitted (or
   terminated) as a pair - for instance a bi-directional voice call
   only makes sense if flows in both directions are admitted.
   However, PCN's mechanisms concern admission and termination of a
   single flow, and coordination of the decision for both flows is a
   matter for the signalling protocol and out of scope of PCN. One
   possible example would use SIP pre-conditions; there are others.

o Global Coordination: PCN makes its admission decision based on
   PCN-markings on a particular ingress-egress-aggregate. Decisions
   about flows through a different ingress-egress-aggregate are made
   independently. However, one can imagine network topologies and

traffic matrices where, from a global perspective, it would be
better to make a coordinated decision across all the ingress-
egress-aggregates for the whole PCN-domain. For example, to block
(or even terminate) flows on one ingress-egress-aggregate so that
more important flows through a different ingress-egress-aggregate
could be admitted. The problem may well be second order.

   o Aggregate Traffic Characteristics: Even when the number of flows
      is stable, the traffic level through the PCN-domain will vary
      because the sources vary their traffic rates. PCN works best when
      there's not too much variability in the total traffic level at a
      PCN-node's interface (ie in the aggregate traffic from all
      sources). Too much variation means that a node may (at one
      moment) not be doing any PCN-marking and then (at another moment)
      drop packets because it's overloaded. This makes it hard to tune
      the admission control scheme to stop admitting new flows at the
      right time. Therefore the problem is more likely with fewer,
      burstier flows.

   o Flash crowds and Speed of Reaction: PCN is a measurement-based
      mechanism and so there is an inherent delay between packet marking
      by PCN-interior-nodes and any admission control reaction at PCN-
      boundary-nodes. For example, potentially if a big burst of
      admission requests occurs in a very short space of time (eg
      prompted by a televote), they could all get admitted before enough
      PCN-marks are seen to block new flows. In other words, any
      additional load offered within the reaction time of the mechanism
      mustn't move the PCN-domain directly from no congestion to
      overload. This 'vulnerability period' may impact at the
      signalling level, for instance QoS requests should be rate limited
      to bound the number of requests able to arrive within the
      vulnerability period.

   o Silent at start: after a successful admission request the source
      may wait some time before sending data (eg waiting for the called
      party to answer). Then the risk is that, in some circumstances,
      PCN's measurements underestimate what the pre-congestion level
      will be when the source does start sending data.

   o Compatibility of PCN-encoding with ECN-encoding. This issue will
      be considered further in the PCN WG Milestone 'Survey of encoding
      choices'.

7. Probing

7.1. Introduction

Probing is an optional mechanism to assist admission control.

PCN's admission control, as described so far, is essentially a
reactive mechanism where the PCN-egress-node monitors the pre-
congestion level for traffic from each PCN-ingress-node; if the level
rises then it blocks new flows on that ingress-egress-aggregate.
However, it's possible that an ingress-egress-aggregate carries no

traffic, and so the PCN-egress-node can't make an admission decision
using the usual method described earlier.

One approach is to be "optimistic" and simply admit the new flow.
However it's possible to envisage a scenario where the traffic levels
on other ingress-egress-aggregates are already so high that they're
blocking new PCN-flows, and admitting a new flow onto this 'empty'
ingress-egress-aggregate adds extra traffic onto the link that's
already pre-congested - which may 'tip the balance' so that PCN's
flow termination mechanism is activated or some packets are dropped.
This risk could be lessened by configuring on each link sufficient
'safety margin' above the PCN-lower-rate.

An alternative approach is to make PCN a more proactive mechanism.
The PCN-ingress-node explicitly determines, before admitting the
prospective new flow, whether the ingress-egress-aggregate can
support it. This can be seen as a "pessimistic" approach, in
contrast to the "optimism" of the approach above. It involves
probing: a PCN-ingress-node generates and sends probe packets in
order to test the pre-congestion level that the flow would
experience.

One possibility is that a probe packet is just a dummy data packet,
generated by the PCN-ingress-node and addressed to the PCN-egress-
node. Another possibility is that a probe packet is a signalling
packet that is anyway travelling from the PCN-ingress-node to the
PCN-egress-node (eg an RSVP PATH message travelling from source to
destination).

. Probing functions

   The probing functions are:

   o Make decision that probing is needed. As described above, this is
      when the ingress-egress-aggregate or the ECMP path carries no PCN-
      traffic. An alternative is always to probe, ie probe before
      admitting every PCN-flow.

   o (if required) Communicate the request that probing is needed - the
      PCN-egress-node signals to the PCN-ingress-node that probing is
      needed

   o (if required) Generate probe traffic - the PCN-ingress-node
      generates the probe traffic. The appropriate number (or rate) of
      probe packets will depend on the PCN-marking algorithm; for
      example an excess-rate-marking algorithm generates fewer PCN-marks
      than a threshold-marking algorithm, and so will need more probe
      packets.

---

   o Forward probe packets - as far as PCN-interior-nodes are
      concerned, probe packets must be handled the same as (ordinary
      data) PCN-packets, in terms of routing, scheduling and PCN-
      marking.

   o Consume probe packets - the PCN-egress-node consumes probe packets
      to ensure that they don't travel beyond the PCN-domain.

7.3. Discussion of rationale for probing, its downsides and open issues

   It is an unresolved question whether probing is really needed, but
   three viewpoints have been put forward as to why it is useful. The
   first is perhaps the most obvious: there is no PCN-traffic on the
   ingress-egress-aggregate. The second assumes that multipath routing
   ECMP is running in the PCN-domain. The third viewpoint is that
   admission control is always done by probing. We now consider each in
   turn.

   The first viewpoint assumes the following:

   o There is no PCN-traffic on the ingress-egress-aggregate (so a

normal admission decision cannot be made).

o   Simply admitting the new flow has a significant risk of leading to
    overload: packets dropped or flows terminated.

On the former bullet, [PCN-email-traffic-empty-aggregates] suggests
that, during the future busy hour of a national network with about
100 PCN-boundary-nodes, there are likely to be significant numbers of
aggregates with very few flows under nearly all circumstances.

The latter bullet could occur if a new flow starts on many of the
empty ingress-egress-aggregates and causes overload on a link in the
PCN-domain. To be a problem this would probably have to happen in a
short time period (flash crowd) because, after the reaction time of
the system, other (non-empty) ingress-egress-aggregates that pass
through the link will measure pre-congestion and so block new flows,
and also flows naturally end anyway.

The downsides of probing for this viewpoint are:

o   Probing adds delay to the admission control process.

o   Sufficient probing traffic has to be generated to test the pre-
    congestion level of the ingress-egress-aggregate. But the probing
    traffic itself may cause pre-congestion, causing other PCN-flows
    to be blocked or even terminated – and in the flash crowd scenario
    there will be probing on many ingress-egress-aggregates.

The open issues associated with this viewpoint include:

o   What rate and pattern of probe packets does the PCN-ingress-node
    need to generate, so that there's enough traffic to make the
    admission decision?

o   What difficulty does the delay (whilst probing is done) cause
    applications, eg packets might be dropped?

o   Are there other ways of dealing with the flash crowd scenario?
    For instance limit the rate at which new flows are admitted; or
    perhaps for a PCN-egress-node to block new flows on its empty
    ingress-egress-aggregates when its non-empty ones are pre-
    congested.

The second viewpoint applies in the case where there is multipath
routing (ECMP) in the PCN-domain. Note that ECMP is often used on
core networks. There are two possibilities:

(1) If admission control is based on measurements of the ingress-
egress-aggregate, then the viewpoint that probing is useful assumes:

o   there's a significant chance that the traffic is unevenly balanced
    across the ECMP paths, and hence there's a significant risk of
    admitting a flow that should be blocked (because it follows an
    ECMP path that is pre-congested) or blocking a flow that should be
    admitted.

o   Note: [PCN-email-ECMP] suggests unbalanced traffic is quite
    possible, even with quite a large number of flows on a PCN-link
    (eg 1000) when Assumption 3 (aggregation) is likely to be
    satisfied.

(2) If admission control is based on measurements of pre-congestion
on specific ECMP paths, then the viewpoint that probing is useful
assumes:

o   There is no PCN-traffic on the ECMP path on which to base an
    admission decision.

o   Simply admitting the new flow has a significant risk of leading to
    overload.

o   The PCN-egress-node can match a packet to an ECMP path.

o   Note: This is similar to the first viewpoint and so similarly
    could occur in a flash crowd if a new flow starts more-or-less
    simultaneously on many of the empty ECMP paths. Because there are

    several (sometimes many) ECMP paths between each pair of PCN-
    boundary-nodes, it's presumably more likely that an ECMP path is
    'empty' than an ingress-egress-aggregate. To constrain the number
    of ECMP paths, a few tunnels could be set-up between each pair of
    PCN-boundary-nodes. Tunnelling also solves the third bullet
    (which is otherwise hard because an ECMP routing decision is made
    independently on each node).

The downsides of probing for this viewpoint are:

o Probing adds delay to the admission control process.

o Sufficient probing traffic has to be generated to test the pre-
   congestion level of the ECMP path. But there's the risk that the
   probing traffic itself may cause pre-congestion, causing other
   PCN-flows to be blocked or even terminated.

o The PCN-egress-node needs to consume the probe packets to ensure
   they don't travel beyond the PCN-domain (eg they might confuse the
   destination end node). Hence somehow the PCN-egress-node has to
   be able to disambiguate a probe packet from a data packet, via the
   characteristic setting of particular bit(s) in the packet's header
   or body - but these bit(s) mustn't be used by any PCN-interior-
   node's ECMP algorithm. In the general case this isn't possible,
   but it should be OK for a typical ECMP algorithm which examines:
   the source and destination IP addresses and port numbers, the
   protocol ID and the DSCP.

The third viewpoint assumes the following:

o Simply admitting the new flow has a significant risk of leading to
   overload, because the PCN-domain reaches out towards the end
   terminals where link capacity is low.

o Every admission control decision involves probing, using the
   signalling set-up message as the probe packet (eg RSVP PATH).

o The PCN-marking behaviour is such that every packet is PCN-marked
   if the flow should be blocked, hence only a single probing packet
   is needed.

The first point breaks Assumption 3 (aggregation) and hence means
that this viewpoint is out of scope of the initial Charter of the PCN
WG.

8. Operations and Management

   This Section considers operations and management issues, under the
   FCAPS headings: OAM of Faults, Configuration, Accounting, Performance
   and Security. Provisioning is discussed with performance.

8.1. Configuration OAM

   This architecture document predates the detailed standards actions of
   the PCN WG. Here we assume that only interoperable PCN-marking
   behaviours will be standardised, otherwise we would have to consider
   how to avoid interactions between non-interoperable marking
   behaviours. However, more diversity in edge-node behaviours is
   expected, in order to interface with diverse industry architectures.

   PCN configuration control variables fall into the following
   categories:

   o system options (enabling or disabling behaviours)

   o parameters (setting levels, addresses etc)

   All configurable variables will need to sit within an SNMP management
   framework [RFC3411], being structured within a defined management
   information base (MIB) on each node, and being remotely readable and
   settable via a suitably secure management protocol (SNMPv3).

   Some configuration options and parameters have to be set once to
   'globally' control the whole PCN-domain. Where possible, these are
   identified below. This may affect operational complexity and the
   chances of interoperability problems between kit from different
   vendors.

8.1.1. System options

   On PCN-interior-nodes there will be very few system options:

   o Whether two PCN-markings (based on the PCN-lower-rate and PCN-
      upper-rate) are enabled or only one (see Section 4.3). Typically
      all nodes throughout a PCN-domain will be configured the same in
      this respect. However, exceptions could be made. For example, if
      most PCN-nodes used both markings, but some legacy hardware was
      incapable of running two algorithms, an operator might be willing
      to configure these legacy nodes solely for PCN-marking based on
      the PCN-upper-rate to enable flow termination as a back-stop. It
      would be sensible to place such nodes where they could be
      provisioned with a greater leeway over expected traffic levels.

o   which marking algorithm to use, if an equipment vendor provides a
    choice

PCN-boundary-nodes (ingress and egress) will have more system
options:

o   Which of admission and flow termination are enabled. If any PCN-
    interior-node is configured to generate a marking, all PCN-
    boundary-nodes must be able to handle that marking. Therefore all
    PCN-boundary-nodes must be configured the same in this respect.

o   Where flow admission and termination decisions are made: at the
    PCN-ingress-node, PCN-egress-node or at a centralised node (see
    Sections 5.4 and 5.5). Theoretically, this configuration choice
    could be negotiated for each pair of PCN-boundary-nodes, but we
    cannot imagine why such complexity would be required, except
    perhaps in future inter-domain scenarios.

PCN-egress-nodes will have further system options:

o   How the mapping should be established between each packet and its
    aggregate, eg by MPLS label, by IP packet filterspec; and how to
    take account of ECMP.

o   If an equipment vendor provides a choice, there may be options to
    select which smoothing algorithm to use for measurements.

8.1.2. Parameters

Like any DiffServ domain, every node within a PCN-domain will need to
be configured with the DSCP(s) used to identify PCN-packets. On each
interior link the main configuration parameters are the PCN-lower-
rate and PCN-upper-rate. A larger PCN-lower-rate enables more PCN-
traffic to be admitted on a link, hence improving capacity
utilisation. A PCN-upper-rate set further above the PCN-lower-rate
allows greater increases in traffic (whether due to natural
fluctuations or some unexpected event) before any flows are
terminated, ie minimises the chances of unnecessarily triggering the
termination mechanism. For instance an operator may want to design
their network so that it can cope with a failure of any single PCN-
node without terminating any flows.

Setting these rates on first deployment of PCN will be very similar

to the traditional process for sizing an admission controlled
network, depending on: the operator's requirements for minimising
flow blocking (grade of service), the expected PCN traffic load on
each link and its statistical characteristics (the traffic matrix),
contingency for re-routing the PCN traffic matrix in the event of

single or multiple failures and the expected load from other classes
relative to link capacities. But once a domain is up and running, a
PCN design goal is to be able to determine growth in these configured
rates much more simply, by monitoring PCN-marking rates from actual
rather than expected traffic (see Section 8.2 on Performance &
Provisioning).

Operators may also wish to configure a rate greater than the PCN-
upper-rate that is the absolute maximum rate that a link allows for
PCN-traffic. This may simply be the physical link rate, but some
operators may wish to configure a logical limit to prevent starvation
of other traffic classes during any brief period after PCN-traffic
exceeds the PCN-upper-rate but before flow termination brings it back
below this rate.

Specific marking algorithms will also depend on further configuration
parameters. For instance, threshold-marking will require a threshold
queue depth and excess-rate-marking may require a scaling parameter.
It will be preferable for each marking algorithm to have rules to set
defaults for these parameters relative to the reference marking rate,
but then allow operators to change them, for instance if average
traffic characteristics change over time. The PCN-egress-node may
allow configuration of the following:

o   how it smoothes metering of PCN-markings (eg EWMA parameters)

Whichever node makes admission and flow termination decisions will
contain algorithms for converting PCN-marking levels into admission
or flow termination decisions. These will also require configurable
parameters, for instance:

o   Any admission control algorithm will at least require a marking
    threshold setting above which it denies admission to new flows;

o   flow termination algorithms will probably require a parameter to
    delay termination of any flows until it is more certain that an

anomalous event is not transient;

    o   a parameter to control the trade-off between how quickly excess
        flows are terminated and over-termination.

    One particular proposal, [I-D.charny-pcn-single-marking] would
    require a global parameter to be defined on all PCN-nodes, but only
    needs the PCN-lower-rate to be configured on each link. The global
    parameter is a scaling factor between admission and termination, for
    example the amount by which the PCN-upper-rate is implicitly assumed
    to be above the PCN-lower-rate. [I-D.charny-pcn-single-marking]
    discusses in full the impact of this particular proposal on the

    operation of PCN.

8.2. Performance & Provisioning OAM

    Monitoring of performance factors measurable from *outside* the PCN
    domain will be no different with PCN than with any other packet-based
    flow admission control system, both at the flow level (blocking
    probability etc) and the packet level (jitter [RFC3393], [Y.1541],
    loss rate [RFC4656], mean opinion score [P.800], etc). The
    difference is that PCN is intentionally designed to indicate
    internally which exact resource(s) are the cause of performance
    problems and by how much.

    Even better, PCN indicates which resources will probably cause
    problems if they are not upgraded soon. This can be achieved by the
    management system monitoring the total amount (in bytes) of PCN-
    marking generated by each queue over a period. Given possible long
    provisioning lead times, pre-congestion volume is the best metric to
    reveal whether sufficient persistent demand has mounted up to warrant
    an upgrade. Because, even before utilisation becomes problematic,
    the statistical variability of traffic will cause occasional bursts
    of pre-congestion. This 'early warning system' decouples the process
    of adding customers from the provisioning process. This should cut
    the time to add a customer when compared against admission control
    provided over native DiffServ [RFC2998], because it saves having to
    re-run the capacity planning process before adding each customer.

    Alternatively, before triggering an upgrade, the long term pre-
    congestion volume on each link can be used to balance traffic load

across the PCN-domain by adjusting the link weights of the routing
system. When an upgrade to a link's configured PCN-rates is
required, it may also be necessary to upgrade the physical capacity
available to other classes. But usually there will be sufficient
physical capacity for the upgrade to go ahead as a simple
configuration change. Alternatively, [Songhurst] has proposed an
adaptive rather than preconfigured system, where the configured PCN-
lower-rate is replaced with a high and low water mark and the marking
algorithm automatically optimises how physical capacity is shared
using the relative loads from PCN and other traffic classes.

All the above processes require just three extra counters associated
with each PCN queue: PCN-markings associated with the PCN-lower-rate
and PCN-upper-rate, and drop. Every time a PCN packet is marked or
dropped its size in bytes should be added to the appropriate counter.
Then the management system can read the counters at any time and
subtract a previous reading to establish the incremental volume of
each type of (pre-)congestion. Readings should be taken frequently,
so that anomalous events (eg re-routes) can be separated from regular

fluctuating demand if required.

8.3. Accounting OAM

Accounting is only done at trust boundaries so it is out of scope of
the initial Charter of the PCN WG which is confined to intra-domain
issues. Use of PCN internal to a domain makes no difference to the
flow signalling events crossing trust boundaries outside the PCN-
domain, which are typically used for accounting.

8.4. Fault OAM

Fault OAM is about preventing faults, telling the management system
(or manual operator) that the system has recovered (or not) from a
failure, and about maintaining information to aid fault diagnosis.

Admission blocking and particularly flow termination mechanisms
should rarely be needed in practice. It would be unfortunate if they
didn't work after an option had been accidentally disabled.
Therefore it will be necessary to regularly test that the live system
works as intended (devising a meaningful test is left as an exercise
for the operator).

Section 5.9 describes how the PCN architecture has been designed to
ensure admitted flows continue gracefully after recovering
automatically from link or node failures. The need to record and
monitor re-routing events affecting signalling is unchanged by the
addition of PCN to a DiffServ domain. Similarly, re-routing events
within the PCN-domain will be recorded and monitored just as they
would be without PCN.

PCN-marking does make it possible to record 'near-misses'. For
instance, at the PCN-egress-node a 'reporting threshold' could be set
to monitor how often the system comes close to triggering flow
blocking without actually doing so. Similarly, bursts of flow
termination marking could be recorded even if they are not
sufficiently sustained to trigger flow termination. Such statistics
could be correlated with per-queue counts of marking volume (Section
8.2) to upgrade resources in danger of causing service degradation,
or to trigger manual tracing of intermittent incipient errors that
would otherwise have gone unnoticed.

Finally, of course, many faults are caused by failings in the
management process ('human error'): a wrongly configured address in a
node, a wrong address given in a signalling protocol, a wrongly
configured parameter in a queueing algorithm, a node set into a
different mode from other nodes, and so on. Generally, a clean
design with few configurable options ensures this class of faults can

be traced more easily and prevented more often. Sound management
practice at run-time also helps. For instance: a management system
should be used that constrains configuration changes within system
rules (eg preventing an option setting inconsistent with other
nodes); configuration options should also be recorded in an offline
database; and regular automatic consistency checks between live
systems and the database. PCN adds nothing specific to this class of
problems. By the time standards are in place, we expect that the PCN
WG will have ruthlessly removed gratuitous configuration choices.
However, at the time of writing, the WG is yet to choose between
multiple competing proposals, so the range of possible options in
Section 8.1 does seem rather wide compared to the original near-zero
configuration intent of the architecture.

8.5. Security OAM

Security OAM is about using secure operational practices as well as
being able to track security breaches or near-misses at run-time.
PCN adds few specifics to the general good practice required in this
field [RFC4778], other than those below. The correct functions of
the system should be monitored (Section 8.2) in multiple independent
ways and correlated to detect possible security breaches. Persistent
(pre-)congestion marking should raise an alarm (both on the node
doing the marking and on the PCN-egress-node metering it).
Similarly, persistently poor external QoS metrics such as jitter or
MOS should raise an alarm. The following are examples of symptoms
that may be the result of innocent faults, rather than attacks, but
until diagnosed they should be logged and trigger a security alarm:

o Anomalous patterns of non-conforming incoming signals and packets
   rejected at the PCN-ingress-nodes (eg packets already marked PCN-
   capable, or traffic persistently starving token bucket policers).

o PCN-capable packets arriving at a PCN-egress-node with no
   associated state for mapping them to a valid ingress-egress-
   aggregate.

o A PCN-ingress-node receiving feedback signals about the pre-
   congestion level on a non-existent aggregate, or that are
   inconsistent with other signals (eg unexpected sequence numbers,
   inconsistent addressing, conflicting reports of the pre-congestion
   level, etc).

o Pre-congestion marking arriving at an PCN-egress-node with
   (pre-)congestion markings focused on particular flows, rather than
   randomly distributed throughout the aggregate.

9. IANA Considerations

   This memo includes no request to IANA.

10. Security considerations

   Security considerations essentially come from the Trust Assumption

(Section 3.1), ie that all PCN-nodes are PCN-enabled and trust each
other for truthful PCN-marking and transport. PCN splits
functionality between PCN-interior-nodes and PCN-boundary-nodes, and
the security considerations are somewhat different for each, mainly
because PCN-boundary-nodes are flow-aware and PCN-interior-nodes are
not.

o because the PCN-boundary-nodes are flow-aware, they are trusted to
   use that awareness correctly. The degree of trust required
   depends on the kinds of decisions they have to make and the kinds
   of information they need to make them. For example when the PCN-
   boundary-node needs to know the contents of the sessions for
   making the admission and termination decisions (perhaps based on
   the MLPP precedence), or when the contents are highly classified,
   then the security requirements for the PCN-boundary-nodes involved
   will also need to be high.

o the PCN-ingress-nodes police packets to ensure a flow sticks
   within its agreed limit, and to ensure that only flows which have
   been admitted contribute PCN-traffic into the PCN-domain. The
   policer must drop (or perhaps re-mark to a different DSCP) any
   PCN-packets received that are outside this remit. This is similar
   to the existing IntServ behaviour. Between them the PCN-boundary-
   nodes must encircle the PCN-domain, otherwise PCN-packets could
   enter the PCN-domain without being subject to admission control,
   which would potentially destroy the QoS of existing flows.

o PCN-interior-nodes aren't flow-aware. This prevents some security
   attacks where an attacker targets specific flows in the data plane
   - for instance for DoS or eavesdropping.

o PCN-marking by the PCN-interior-nodes along the packet forwarding
   path needs to be trusted, because the PCN-boundary-nodes rely on
   this information. 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's pre-congested. More subtly, the rogue PCN-interior-node
   could perform these attacks selectively on particular flows, or it
   could PCN-mark the correct fraction overall, but carefully choose
   which flows it marked.

o the PCN-boundary-nodes should be able to deal with DoS attacks and

state exhaustion attacks based on fast changes in per flow
signalling.

o the signalling between the PCN-boundary-nodes (and possibly a
   central control node) must be protected from attacks. For example
   the recipient needs to validate that the message is indeed from
   the node that claims to have sent it. Possible measures include
   digest authentication and protection against replay and man-in-
   the-middle attacks. For the specific protocol RSVP, hop-by-hop
   authentication is in [RFC2747], and
   [I-D.behringer-tsvwg-rsvp-security-groupkeying] may also be
   useful; for a generic signalling protocol the PCN WG document on
   "Requirements for signalling" will describe the requirements in
   more detail.

Operational security advice is given in Section 8.5.

11. Conclusions

The document describes a general architecture for flow admission and
termination based on aggregated pre-congestion information in order
to protect the quality of service of established inelastic flows
within a single DiffServ domain. The main topic is the functional
architecture (first covered at a high level and then at a greater
level of detail). It also mentions other topics like the assumptions
and open issues.

12. Acknowledgements

This document is a revised version of [I-D.eardley-pcn-architecture].
Its authors were: P. Eardley, J. Babiarz, K. Chan, A. Charny, R.
Geib, G. Karagiannis, M. Menth, T. Tsou. They are therefore
contributors to this document.

Thanks to those who've made comments on
[I-D.eardley-pcn-architecture] and on earlier versions of this draft:
Lachlan Andrew, Joe Babiarz, Fred Baker, David Black, Steven Blake,
Bob Briscoe, Ken Carlberg, Anna Charny, Joachim Charzinski, Andras
Csaszar, Lars Eggert, Ruediger Geib, Robert Hancock, Georgios
Karagiannis, Michael Menth, Tom Taylor, Tina Tsou, Delei Yu. Thanks
to Bob Briscoe who extensively revised the Operations and Management
section.

This document is the result of discussions in the PCN WG and
forerunner activity in the TSVWG. A number of previous drafts were

presented to TSVWG: [I-D.chan-pcn-problem-statement],
[I-D.briscoe-tsvwg-cl-architecture], [I-D.briscoe-tsvwg-cl-phb],
[I-D.charny-pcn-single-marking], [I-D.babiarz-pcn-sip-cap],
[I-D.lefaucheur-rsvp-ecn], [I-D.westberg-pcn-load-control]. The
authors of them were: B, Briscoe, P. Eardley, D. Songhurst, F. Le
Faucheur, A. Charny, J. Babiarz, K. Chan, S. Dudley, G. Karagiannis,
A. Bader, L. Westberg, J. Zhang, V. Liatsos, X-G. Liu, A. Bhargava.

13. Comments Solicited

   Comments and questions are encouraged and very welcome. They can be
   addressed to the IETF PCN working group mailing list <pcn@ietf.org>.

14. Changes

   Changes from -01 to -02:

   o S1: Benefits: provisioning bullet extended to stress that PCN does
      not use RFC2475-style traffic conditioning.

   o S1: Deployment models: mentioned, as variant of PCN-domain
      extending to end nodes, that may extend to LAN edge switch.

   o S3.1: Trust Assumption: added note about not needing PCN-marking
      capability if known that an interface cannot become pre-congested.

   o S4: now divided into sub-sections

   o S4.1: Admission control: added second proposed method for how to
      decide to block new flows (PCN-egress-node receives one (or
      several) PCN-marked packets).

   o S5: Probing sub-section removed. Material now in new S7.

   o S5.6: Addressing: clarified how PCN-ingress-node can discover
      address of PCN-egress-node

   o S5.6: Addressing: centralised node case, added that PCN-ingress-
      node may need to know address of PCN-egress-node

   o S5.8: Tunnelling: added case of "partially PCN-capable tunnel" and
      degraded bullet on this in S6 (Open Issues)

   o S7: Probing: new section. Much more comprehensive than old S5.5.

   o S8: Operations and Management: substantially revised.

   o other minor changes not affecting semantics

   Changes from -00 to -01:

   In addition to clarifications and nit squashing, the main changes
   are:

   o S1: Benefits: added one about provisioning (and contrast with
      DiffServ SLAs)

   o S1: Benefits: clarified that the objective is also to stop PCN-
      packets being significantly delayed (previously only mentioned not
      dropping packets)

   o S1: Deployment models: added one where policing is done at ingress
      of access network and not at ingress of PCN-domain (assume trust
      between networks)

   o S1: Deployment models: corrected MPLS-TE to MPLS

   o S2: Terminology: adjusted definition of PCN-domain

   o S3.5: Other assumptions: corrected, so that two assumptions (PCN-
      nodes not performing ECN and PCN-ingress-node discarding arriving
      CE packet) only apply if the PCN WG decides to encode PCN-marking
      in the ECN-field.

   o S4 & S5: changed PCN-marking algorithm to marking behaviour

   o S4: clarified that PCN-interior-node functionality applies for
      each outgoing interface, and added clarification: "The
      functionality is also done by PCN-ingress-nodes for their outgoing
      interfaces (ie those 'inside' the PCN-domain)."

   o S4 (near end): altered to say that a PCN-node "should" dedicate
      some capacity to lower priority traffic so that it isn't starved
      (was "may")

o S5: clarified to say that PCN functionality is done on an
      'interface' (rather than on a 'link')

   o S5.2: deleted erroneous mention of service level agreement

   o S5.5: Probing: re-written, especially to distinguish probing to
      test the ingress-egress-aggregate from probing to test a
      particular ECMP path.

   o S5.7: Addressing: added mention of probing; added that in the case
      where traffic is always tunnelled across the PCN-domain, add a
      note that he PCN-ingress-node needs to know the address of the
      PCN-egress-node.

   o S5.8: Tunnelling: re-written, especially to provide a clearer
      description of copying on tunnel entry/exit, by adding explanation
      (keeping tunnel encaps/decaps and PCN-marking orthogonal),
      deleting one bullet ("if the inner header's marking state is more
      sever then it is preserved" - shouldn't happen), and better
      referencing of other IETF documents.

   o S6: Open issues: stressed that "NOTE: Potential solutions are out
      of scope for this document" and edited a couple of sentences that
      were close to solution space.

   o S6: Open issues: added one about scenarios with only one tunnel
      endpoint in the PCN domain .

   o S6: Open issues: ECMP: added under-admission as another potential
      risk

   o S6: Open issues: added one about "Silent at start"

   o S10: Conclusions: a small conclusions section added.

15. Informative References

   [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.briscoe-tsvwg-cl-phb]
             Briscoe, B., "Pre-Congestion Notification marking",
             draft-briscoe-tsvwg-cl-phb-03 (work in progress),
             October 2006.

   [I-D.babiarz-pcn-sip-cap]
             Babiarz, J., "SIP Controlled Admission and Preemption",
             draft-babiarz-pcn-sip-cap-00 (work in progress),
             October 2006.

   [I-D.lefaucheur-rsvp-ecn]
             Faucheur, F., "RSVP Extensions for Admission Control over
             Diffserv using Pre-congestion Notification (PCN)",

             draft-lefaucheur-rsvp-ecn-01 (work in progress),
             June 2006.

   [I-D.chan-pcn-problem-statement]
             Chan, K., "Pre-Congestion Notification Problem Statement",
             draft-chan-pcn-problem-statement-01 (work in progress),
             October 2006.

   [I-D.ietf-pwe3-congestion-frmwk]
             Bryant, S., "Pseudowire Congestion Control Framework",
             draft-ietf-pwe3-congestion-frmwk-00 (work in progress),
             February 2007.

   [I-D.ietf-tsvwg-admitted-realtime-dscp]
             "DSCPs for Capacity-Admitted Traffic", November 2006, <htt
             p://www.ietf.org/internet-drafts/
             ietf-tsvwg-admitted-realtime-dscp-02.txt>.

   [I-D.briscoe-tsvwg-ecn-tunnel]
             "Layered Encapsulation of Congestion Notification",
             June 2007, <http://www.ietf.org/internet-drafts/
             briscoe-tsvwg-ecn-tunnel-00.txt>.

   [I-D.ietf-tsvwg-ecn-mpls]
             "Explicit Congestion Marking in MPLS", October 2007, <http

                 ://www.ietf.org/internet-drafts/
                 draft-ietf-tsvwg-ecn-mpls-02.txt>.

    [I-D.charny-pcn-single-marking]
                 "Pre-Congestion Notification Using Single Marking for
                 Admission and Termination", November 2007, <http://
                 www.ietf.org/internet-drafts/
                 draft-charny-pcn-single-marking-03.txt>.

    [I-D.eardley-pcn-architecture]
                 "Pre-Congestion Notification Architecture", June 2007, <ht
                 tp://www.ietf.org/internet-drafts/
                 draft-eardley-pcn-architecture-00.txt>.

    [I-D.westberg-pcn-load-control]
                 "LC-PCN: The Load Control PCN Solution", November 2007, <h
                 ttp://www.ietf.org/internet-drafts/
                 draft-westberg-pcn-load-control-02.txt>.

    [I-D.behringer-tsvwg-rsvp-security-groupkeying]
                 "A Framework for RSVP Security Using Dynamic Group
                 Keying", June 2007, <http://www.watersprings.org/pub/id/
                 draft-behringer-tsvwg-rsvp-security-groupkeying-00.txt>.

    [I-D.briscoe-re-pcn-border-cheat]
                 "Emulating Border Flow Policing using Re-ECN on Bulk
                 Data", June 2006, <http://www.watersprings.org/pub/id/
                 briscoe-re-pcn-border-cheat-01.txt>.

    [RFC4303]   Kent, S., "IP Encapsulating Security Payload (ESP)",
                 RFC 4303, December 2005.

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

    [RFC3246]   Davie, B., Charny, A., Bennet, J., Benson, K., Le Boudec,
                 J., Courtney, W., Davari, S., Firoiu, V., and D.
                 Stiliadis, "An Expedited Forwarding PHB (Per-Hop
                 Behavior)", RFC 3246, March 2002.

    [RFC4594]   Babiarz, J., Chan, K., and F. Baker, "Configuration

                Guidelines for DiffServ Service Classes", RFC 4594,
                August 2006.

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

    [RFC2211]   Wroclawski, J., "Specification of the Controlled-Load
                Network Element Service", RFC 2211, September 1997.

    [RFC2998]   Bernet, Y., Ford, P., Yavatkar, R., Baker, F., Zhang, L.,
                Speer, M., Braden, R., Davie, B., Wroclawski, J., and E.
                Felstaine, "A Framework for Integrated Services Operation
                over Diffserv Networks", RFC 2998, November 2000.

    [RFC3270]   Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen,
                P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-
                Protocol Label Switching (MPLS) Support of Differentiated
                Services", RFC 3270, May 2002.

    [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.

    [RFC2983]   Black, D., "Differentiated Services and Tunnels",
                RFC 2983, October 2000.

    [RFC2747]   Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic
                Authentication", RFC 2747, January 2000.

    [RFC3411]   Harrington, D., Presuhn, R., and B. Wijnen, "An
                Architecture for Describing Simple Network Management
                Protocol (SNMP) Management Frameworks", STD 62, RFC 3411,
                December 2002.

    [RFC3393]   Demichelis, C. and P. Chimento, "IP Packet Delay Variation
                Metric for IP Performance Metrics (IPPM)", RFC 3393,
                November 2002.

[RFC4656]   Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M.
            Zekauskas, "A One-way Active Measurement Protocol
            (OWAMP)", RFC 4656, September 2006.

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

[ITU-MLPP]
            "Multilevel Precedence and Pre-emption Service (MLPP)",
            ITU-T Recommendation I.255.3, 1990.

[Iyer]   "An approach to alleviate link overload as observed on an
            IP backbone", IEEE INFOCOM , 2003,
            <http://www.ieee-infocom.org/2003/papers/10_04.pdf>.

[Shenker]   "Fundamental design issues for the future Internet", IEEE
            Journal on selected areas in communications pp 1176 -
            1188, Vol 13 (7), 1995.

[Y.1541]   "Network Performance Objectives for IP-based Services",
            ITU-T Recommendation Y.1541, February 2006.

[P.800]   "Methods for subjective determination of transmission
            quality", ITU-T Recommendation P.800, August 1996.

[Songhurst]
            "Guaranteed QoS Synthesis for Admission Control with
            Shared Capacity", BT Technical Report TR-CXR9-2006-001,
            Feburary 2006, <http://www.cs.ucl.ac.uk/staff/B.Briscoe/
            projects/ipe2eqos/gqs/papers/GQS_shared_tr.pdf>.

[PCN-email-ECMP]
            "Email to PCN WG mailing list", November 2007, <http://
            www1.ietf.org/mail-archive/web/pcn/current/msg00871.html>.

[PCN-email-traffic-empty-aggregates]
            "Email to PCN WG mailing list", October 2007, <http://
            www1.ietf.org/mail-archive/web/pcn/current/msg00831.html>.

Author's Address

   Philip Eardley
   BT
   B54/77, Sirius House Adastral Park Martlesham Heath
   Ipswich, Suffolk IP5 3RE
   United Kingdom

   Email: philip.eardley@bt.com