Guidelines for Adding Congestion Notification to Protocols that Encapsulate IP
draft-ietf-tsvwg-ecn-encap-guidelines-15

Abstract

   The purpose of this document is to guide the design of congestion
   notification in any lower layer or tunnelling protocol that
   encapsulates IP. The aim is for explicit congestion signals to
   propagate consistently from lower layer protocols into IP. Then the
   IP internetwork layer can act as a portability layer to carry
   congestion notification from non-IP-aware congested nodes up to the
   transport layer (L4). Following these guidelines should assure
   interworking among IP layer and lower layer congestion notification
   mechanisms, whether specified by the IETF or other standards bodies.
   This document updates the advice to subnetwork designers about ECN in
   RFC 3819.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at https://datatracker.ietf.org/drafts/current/.

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

   This Internet-Draft will expire on September 9, 2021.

Copyright Notice

Table of Contents

1. Introduction

The benefits of Explicit Congestion Notification (ECN) described in
[RFC8087] and summarized below can only be fully realized if support
for ECN is added to the relevant subnetwork technology, as well as to
IP. When a lower layer buffer drops a packet obviously it does not
just drop at that layer; the packet disappears from all layers. In
contrast, when active queue management (AQM) at a lower layer marks a
packet with ECN, the marking needs to be explicitly propagated up the
layers. The same is true if AQM marks the outer header of a packet
that encapsulates inner tunnelled headers. Forwarding ECN is not as
straightforward as other headers because it has to be assumed ECN may
be only partially deployed. If a lower layer header that contains
ECN congestion indications is stripped off by a subnet egress that is
not ECN-aware, or if the ultimate receiver or sender is not ECN-
aware, congestion needs to be indicated by dropping a packet, not
marking it.

The purpose of this document is to guide the addition of congestion
notification to any subnet technology or tunnelling protocol, so that
lower layer AQM algorithms can signal congestion explicitly and it
will propagate consistently into encapsulated (higher layer) headers,
otherwise the signals will not reach their ultimate destination.

ECN is defined in the IP header (v4 and v6) [RFC3168] to allow a
resource to notify the onset of queue build-up without having to drop
packets, by explicitly marking a proportion of packets with the
congestion experienced (CE) codepoint.

Given a suitable marking scheme, ECN removes nearly all congestion
loss and it cuts delays for two main reasons:

o It avoids the delay when recovering from congestion losses, which
   particularly benefits small flows or real-time flows, making their
   delivery time predictably short [RFC2884];

o As ECN is used more widely by end-systems, it will gradually
   remove the need to configure a degree of delay into buffers before
   they start to notify congestion (the cause of bufferbloat). This
   is because drop involves a trade-off between sending a timely
   signal and trying to avoid impairment, whereas ECN is solely a
   signal not an impairment, so there is no harm triggering it
   earlier.

Some lower layer technologies (e.g. MPLS, Ethernet) are used to form
subnetworks with IP-aware nodes only at the edges. These networks
are often sized so that it is rare for interior queues to overflow.
However, until recently this was more due to the inability of TCP to

saturate the links. For many years, fixes such as window scaling [RFC7323] proved hard to deploy. And the Reno variant of TCP has remained in widespread use despite its inability to scale to high flow rates. However, now that modern operating systems are finally capable of saturating interior links, even the buffers of well-provisioned interior switches will need to signal episodes of queuing.

Propagation of ECN is defined for MPLS [RFC5129], and is being defined for TRILL [RFC7780], [I-D.ietf-trill-ecn-support], but it remains to be defined for a number of other subnetwork technologies.

Similarly, ECN propagation is yet to be defined for many tunnelling protocols. [RFC6040] defines how ECN should be propagated for IP-in-IPv4 [RFC2003], IP-in-IPv6 [RFC2473] and IPsec [RFC4301] tunnels, but there are numerous other tunnelling protocols with a shim and/or a layer 2 header between two IP headers (v4 or v6). Some address ECN propagation between the IP headers, but many do not. This document gives guidance on how to address ECN propagation for future tunnelling protocols, and a companion standards track specification [I-D.ietf-tsvwg-rfc6040update-shim] updates those existing IP-shim-(L2)-IP protocols that are under IETF change control and still widely used.

Incremental deployment is the most delicate aspect when adding support for ECN. The original ECN protocol in IP [RFC3168] was carefully designed so that a congested buffer would not mark a packet (rather than drop it) unless both source and destination hosts were ECN-capable. Otherwise its congestion markings would never be detected and congestion would just build up further. However, to support congestion marking below the IP layer or within tunnels, it is not sufficient to only check that the two layer 4 transport end-points support ECN; correct operation also depends on the decapsulator at each subnet or tunnel egress faithfully propagating congestion notifications to the higher layer. Otherwise, a legacy decapsulator might silently fail to propagate any ECN signals from the outer to the forwarded header. Then the lost signals would never be detected and again congestion would build up further. The guidelines given later require protocol designers to carefully consider incremental deployment, and suggest various safe approaches for different circumstances.

Of course, the IETF does not have standards authority over every link layer protocol. So this document gives guidelines for designing propagation of congestion notification across the interface between IP and protocols that may encapsulate IP (i.e. that can be layered beneath IP). Each lower layer technology will exhibit different issues and compromises, so the IETF or the relevant standards body

must be free to define the specifics of each lower layer congestion
notification scheme. Nonetheless, if the guidelines are followed,
congestion notification should interwork between different
technologies, using IP in its role as a 'portability layer'.

Therefore, the capitalized terms 'SHOULD' or 'SHOULD NOT' are often
used in preference to 'MUST' or 'MUST NOT', because it is difficult
to know the compromises that will be necessary in each protocol
design. If a particular protocol design chooses not to follow a
'SHOULD (NOT)' given in the advice below, it MUST include a sound
justification.

It has not been possible to give common guidelines for all lower
layer technologies, because they do not all fit a common pattern.
Instead they have been divided into a few distinct modes of
operation: feed-forward-and-upward; feed-upward-and-forward; feed-
backward; and null mode. These modes are described in Section 3,
then in the subsequent sections separate guidelines are given for
each mode.

1.1. Update to RFC 3819

This document updates the brief advice to subnetwork designers about
ECN in [RFC3819], by replacing the last two paragraphs of Section 13
with the following sentence:

> By following the guidelines in [this document], subnetwork
> designers can enable a layer-2 protocol to participate in
> congestion control without dropping packets via propagation of
> explicit congestion notification (ECN [RFC3168]) to receivers.

and adding [this document] as an informative reference. {RFC Editor:
Please replace both instances of [this document] above with the
number of the present RFC when published.}

1.2. Scope

This document only concerns wire protocol processing of explicit
notification of congestion. It makes no changes or recommendations
concerning algorithms for congestion marking or for congestion
response, because algorithm issues should be independent of the layer
the algorithm operates in.

The default ECN semantics are described in [RFC3168] and updated by
[RFC8311]. Also the guidelines for AQM designers [RFC7567] clarify
the semantics of both drop and ECN signals from AQM algorithms.
[RFC4774] is the appropriate best current practice specification of
how algorithms with alternative semantics for the ECN field can be

partitioned from Internet traffic that uses the default ECN
semantics. There are two main examples for how alternative ECN
semantics have been defined in practice:

o RFC 4774 suggests using the ECN field in combination with a
   Diffserv codepoint such as in PCN [RFC6660], Voice over 3G [UTRAN]
   or Voice over LTE (VoLTE) [LTE-RA];

o RFC 8311 suggests using the ECT(1) codepoint of the ECN field to
   indicate alternative semantics such as for the experimental Low
   Latency Low Loss Scalable throughput (L4S) service
   [I-D.ietf-tsvwg-ecn-l4s-id]).

The aim is that the default rules for encapsulating and decapsulating
the ECN field are sufficiently generic that tunnels and subnets will
encapsulate and decapsulate packets without regard to how algorithms
elsewhere are setting or interpreting the semantics of the ECN field.
[RFC6040] updates RFC 4774 to allow alternative encapsulation and
decapsulation behaviours to be defined for alternative ECN semantics.
However it reinforces the same point - that it is far preferable to
try to fit within the common ECN encapsulation and decapsulation
behaviours, because expecting all lower layer technologies and
tunnels to be updated is likely to be completely impractical.

Alternative semantics for the ECN field can be defined to depend on
the traffic class indicated by the DSCP. Therefore correct
propagation of congestion signals could depend on correct propagation
of the DSCP between the layers and along the path. For instance, if
the meaning of the ECN field depends on the DSCP (as in PCN or VoLTE)
and if the outer DSCP is stripped on descapsulation, as in the pipe
model of [RFC2983], the special semantics of the ECN field would be
lost. Similarly, if the DSCP is changed at the boundary between
Diffserv domains, the special ECN semantics would also be lost. This
is an important implication of the localized scope of most Diffserv
arrangements. In this document, correct propagation of traffic class
information is assumed, while what 'correct' means and how it is
achieved is covered elsewhere (e.g. RFC 2983) and is outside the
scope of the present document.

The guidelines in this document do ensure that common encapsulation
and decapsulation rules are sufficiently generic to cover cases where
ECT(1) is used instead of ECT(0) to identify alternative ECN
semantics (as in L4S [I-D.ietf-tsvwg-ecn-l4s-id]) and where ECN
marking algorithms use ECT(1) to encode 3 severity levels into the
ECN field (e.g. PCN [RFC6660]) rather than the default of 2. All
these different semantics for the ECN field work because it has been
possible to define common default decapsulation rules that allow for
all cases.

Note that the guidelines in this document do not necessarily require
the subnet wire protocol to be changed to add support for congestion
notification. For instance, the Feed-Up-and-Forward Mode
(Section 3.2) and the Null Mode (Section 3.4) do not. Another way to
add congestion notification without consuming header space in the
subnet protocol might be to use a parallel control plane protocol.

This document focuses on the congestion notification interface
between IP and lower layer or tunnel protocols that can encapsulate
IP, where the term 'IP' includes v4 or v6, unicast, multicast or
anycast. However, it is likely that the guidelines will also be
useful when a lower layer protocol or tunnel encapsulates itself,
e.g. Ethernet MAC in MAC ([IEEE802.1Q]; previously 802.1ah) or when
it encapsulates other protocols. In the feed-backward mode,
propagation of congestion signals for multicast and anycast packets
is out-of-scope (because the complexity would make it unlikely to be
attempted).

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in [RFC2119] [RFC8174]
when, and only when, they appear in all capitals, as shown here.

Further terminology used within this document:

Protocol data unit (PDU): Information that is delivered as a unit
   among peer entities of a layered network consisting of protocol
   control information (typically a header) and possibly user data
   (payload) of that layer. The scope of this document includes
   layer 2 and layer 3 networks, where the PDU is respectively termed
   a frame or a packet (or a cell in ATM). PDU is a general term for
   any of these. This definition also includes a payload with a shim
   header lying somewhere between layer 2 and 3.

Transport: The end-to-end transmission control function,
   conventionally considered at layer-4 in the OSI reference model.
   Given the audience for this document will often use the word
   transport to mean low level bit carriage, whenever the term is
   used it will be qualified, e.g. 'L4 transport'.

Encapsulator: The link or tunnel endpoint function that adds an
   outer header to a PDU (also termed the 'link ingress', the 'subnet
   ingress', the 'ingress tunnel endpoint' or just the 'ingress'
   where the context is clear).

   Decapsulator: The link or tunnel endpoint function that removes an
      outer header from a PDU (also termed the 'link egress', the
      'subnet egress', the 'egress tunnel endpoint' or just the 'egress'
      where the context is clear).

   Incoming header: The header of an arriving PDU before encapsulation.

   Outer header: The header added to encapsulate a PDU.

   Inner header: The header encapsulated by the outer header.

   Outgoing header: The header forwarded by the decapsulator.

   CE: Congestion Experienced [RFC3168]

   ECT: ECN-Capable (L4) Transport [RFC3168]

   Not-ECT: Not ECN-Capable (L4) Transport [RFC3168]

   Load Regulator: For each flow of PDUs, the transport function that
      is capable of controlling the data rate. Typically located at the
      data source, but in-path nodes can regulate load in some
      congestion control arrangements (e.g. admission control, policing
      nodes or transport circuit-breakers [RFC8084]). Note the term "a
      function capable of controlling the load" deliberately includes a
      transport that does not actually control the load responsively but
      ideally it ought to (e.g. a sending application without congestion
      control that uses UDP).

   ECN-PDU: A PDU at the IP layer or below with a capacity to signal
      congestion that is part of a congestion control feedback loop
      within which all the nodes necessary to propagate the signal back
      to the Load Regulator are capable of doing that propagation. An
      IP packet with a non-zero ECN field implies that the endpoints are
      ECN-capable, so this would be an ECN-PDU. However, ECN-PDU is
      intended to be a general term for a PDU at lower layers, as well
      as at the IP layer.

   Not-ECN-PDU: A PDU at the IP layer or below that is part of a
      congestion control feedback-loop within which at least one node
      necessary to propagate any explicit congestion notification
      signals back to the Load Regulator is not capable of doing that
      propagation.

3. Modes of Operation

This section sets down the different modes by which congestion
information is passed between the lower layer and the higher one. It
acts as a reference framework for the following sections, which give
normative guidelines for designers of explicit congestion
notification protocols, taking each mode in turn:

Feed-Forward-and-Up: Nodes feed forward congestion notification
   towards the egress within the lower layer then up and along the
   layers towards the end-to-end destination at the transport layer.
   The following local optimisation is possible:

   Feed-Up-and-Forward: A lower layer switch feeds-up congestion
      notification directly into the higher layer (e.g. into the ECN
      field in the IP header), irrespective of whether the node is at
      the egress of a subnet.

Feed-Backward: Nodes feed back congestion signals towards the
   ingress of the lower layer and (optionally) attempt to control
   congestion within their own layer.

Null: Nodes cannot experience congestion at the lower layer except
   at ingress nodes (which are IP-aware or equivalently higher-layer-
   aware).

3.1. Feed-Forward-and-Up Mode

Like IP and MPLS, many subnet technologies are based on self-
contained protocol data units (PDUs) or frames sent unreliably. They
provide no feedback channel at the subnetwork layer, instead relying
on higher layers (e.g. TCP) to feed back loss signals.

In these cases, ECN may best be supported by standardising explicit
notification of congestion into the lower layer protocol that carries
the data forwards. Then a specification is needed for how the egress
of the lower layer subnet propagates this explicit signal into the
forwarded upper layer (IP) header. This signal continues forwards
until it finally reaches the destination transport (at L4). Then
typically the destination will feed this congestion notification back
to the source transport using an end-to-end protocol (e.g. TCP).
This is the arrangement that has already been used to add ECN to IP-
in-IP tunnels [RFC6040], IP-in-MPLS and MPLS-in-MPLS [RFC5129].

This mode is illustrated in Figure 1. Along the middle of the
figure, layers 2, 3 and 4 of the protocol stack are shown, and one
packet is shown along the bottom as it progresses across the network
from source to destination, crossing two subnets connected by a

router, and crossing two switches on the path across each subnet.
Congestion at the output of the first switch (shown as *) leads to a
congestion marking in the L2 header (shown as C in the illustration
of the packet). The chevrons show the progress of the resulting
congestion indication. It is propagated from link to link across the
subnet in the L2 header, then when the router removes the marked L2
header, it propagates the marking up into the L3 (IP) header. The
router forwards the marked L3 header into subnet 2, and when it adds
a new L2 header it copies the L3 marking into the L2 header as well,
as shown by the 'C's in both layers (assuming the technology of
subnet 2 also supports explicit congestion marking).

Note that there is no implication that each 'C' marking is encoded
the same; a different encoding might be used for the 'C' marking in
each protocol.

Finally, for completeness, we show the L3 marking arriving at the
destination, where the host transport protocol (e.g. TCP) feeds it
back to the source in the L4 acknowledgement (the 'C' at L4 in the
packet at the top of the diagram).

```
                         _ _ _
              /_____  | | |C|  ACK Packet (V)
              \          |_|_|_|
 +---+        layer: 2 3 4 header                        +---+
 |   <|<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< Packet V <<<<<<<<<<<<<|<< |L4
 |   |                        +---+                        | ^ |
 |   | . . . . . . Packet U. . | >>|>>> Packet U >>>>>>>>>>>|>^ |L3
 |   |     +---+     +---+      | ^ |      +---+     +---+   |   |
 |   |     | *|>>>>>|>>>|>>>>>|>^ |      |   |     |   |     |   |L2
 |___|_____|___|_____|___|_____|___|_____|___|_____|___|_____|___|
  source          subnet A       router      subnet B       dest

     __ _ _ _   __ _ _ _   __ _ _   __ _ _ _
     | | | | |  | | | |C|  | | |C|    | | |C|C|  Data_____\
     |__|_|_|_|  |__|_|_|_|  |__|_|_|    |__|_|_|_|  Packet (U)  /
  layer: 4 3 2A    4 3 2A     4 3         4 3 2B
  header
```
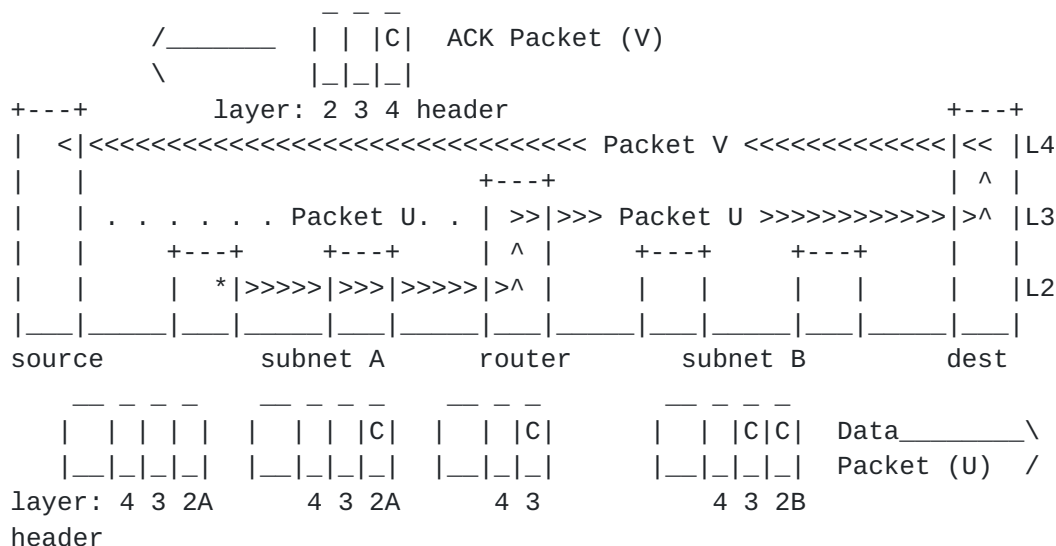
Figure 1: Feed-Forward-and-Up Mode

Of course, modern networks are rarely as simple as this text-book
example, often involving multiple nested layers. For example, a 3GPP
mobile network may have two IP-in-IP (GTP [GTPv1]) tunnels in series
and an MPLS backhaul between the base station and the first router.
Nonetheless, the example illustrates the general idea of feeding
congestion notification forward then upward whenever a header is
removed at the egress of a subnet.

Note that the FECN (forward ECN) bit in Frame Relay [Buck00] and the
explicit forward congestion indication (EFCI [ITU-T.I.371]) bit in
ATM user data cells follow a feed-forward pattern. However, in ATM,
this arrangement is only part of a feed-forward-and-backward pattern
at the lower layer, not feed-forward-and-up out of the lower layer--
the intention was never to interface to IP ECN at the subnet egress.
To our knowledge, Frame Relay FECN is solely used to detect where
more capacity should be provisioned.

3.2. Feed-Up-and-Forward Mode

Ethernet is particularly difficult to extend incrementally to support
explicit congestion notification. One way to support ECN in such
cases has been to use so called 'layer-3 switches'. These are
Ethernet switches that dig into the Ethernet payload to find an IP
header and manipulate or act on certain IP fields (specifically
Diffserv & ECN). For instance, in Data Center TCP [RFC8257], layer-3
switches are configured to mark the ECN field of the IP header within
the Ethernet payload when their output buffer becomes congested.
With respect to switching, a layer-3 switch acts solely on the
addresses in the Ethernet header; it does not use IP addresses, and
it does not decrement the TTL field in the IP header.

```
                     _ _ _
           /_____  | | |C|  ACK packet (V)
           \         |_|_|_|
 +---+         layer: 2 3 4 header                         +---+
 |   |  <|<<<<<<<<<<<<<<<<<<<<<<<<<<<<< Packet V <<<<<<<<<<<<<|<< |L4
 |   |   |                         +---+                     | ^ |
 |   |   | . . .  >>>> Packet U >>>|>>>|>>> Packet U >>>>>>>>>>>|>^ |L3
 |   |      +--^+      +---+       |   |      +---+      +---+      |   |
 |   |      | *|      |   |       |   |      |   |      |   |      |   |L2
 |___|_____|___|_____|___|_____|___|_____|___|_____|___|_____|___|
   source          subnet E         router          subnet F          dest

      __ _ _ _      __ _ _ _      __ _ _      __ _ _ _
     | | | | | |    | | |C| |    | | |C|      | | |C|C|  data_____\
     |__|_|_|_|    |__|_|_|_|    |__|_|_|      |__|_|_|_|  packet (U)  /
   layer: 4 3 2      4 3 2        4 3          4 3 2
   header
```
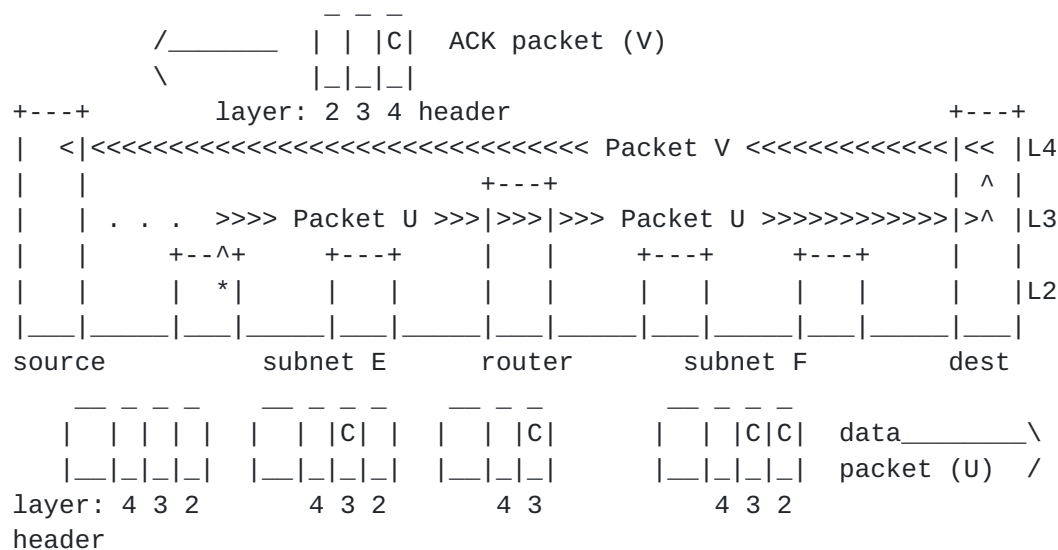
Figure 2: Feed-Up-and-Forward Mode

By comparing Figure 2 with Figure 1, it can be seen that subnet E
(perhaps a subnet of layer-3 Ethernet switches) works in feed-up-and-
forward mode by notifying congestion directly into L3 at the point of
congestion, even though the congested switch does not otherwise act
at L3. In this example, the technology in subnet F (e.g. MPLS) does

support ECN natively, so when the router adds the layer-2 header it
copies the ECN marking from L3 to L2 as well.

3.3. Feed-Backward Mode

In some layer 2 technologies, explicit congestion notification has
been defined for use internally within the subnet with its own
feedback and load regulation, but typically the interface with IP for
ECN has not been defined.

For instance, for the available bit-rate (ABR) service in ATM, the
relative rate mechanism was one of the more popular mechanisms for
managing traffic, tending to supersede earlier designs. In this
approach ATM switches send special resource management (RM) cells in
both the forward and backward directions to control the ingress rate
of user data into a virtual circuit. If a switch buffer is
approaching congestion or is congested it sends an RM cell back
towards the ingress with respectively the No Increase (NI) or
Congestion Indication (CI) bit set in its message type field
[ATM-TM-ABR]. The ingress then holds or decreases its sending bit-
rate accordingly.

```
                     _ _ _
              /_____  | | |C|   ACK packet (X)
              \         |_|_|_|
   +---+          layer: 2 3 4 header                        +---+
   |   <|<<<<<<<<<<<<<<<<<<<<<<<<<<<< Packet X <<<<<<<<<<<<<|<< |L4
   |   |                      +---+                          | ^ |
   |   |                      | *|>>> Packet W >>>>>>>>>>>>|>^ |L3
   |   |     +---+      +---+  |  |      +---+      +---+     |   |
   |   |     |   |      |   |  |  <|<<<<<|<<<|<(V)<|<<<|      |   |L2
   |   | . . | . |Packet U | . . | . | . . | . | . . | .*| . . |   |L2
   |___|_____|___|_____|___|_____|___|_____|___|_____|___|_____|___|
    source          subnet G      router        subnet H       dest
     __ _ _ _    __ _ _ _    __ _ _    __ _ _ _   later
    | | | | | |  | | | | | |  | | | | |    | | |C| |  data_____\
    |__|_|_|_|  |__|_|_|_|  |__|_|_|    |__|_|_|_|  packet (W)  /
        4 3 2       4 3 2       4 3         4 3 2
                                        _
                                     /__  |C|  Feedback control
                                     \    |_|  cell/frame (V)
                                           2
     __ _ _ _    __ _ _ _    __ _ _    __ _ _ _   earlier
    | | | | | |  | | | | | |  | | | | |    | | | | |  data_____\
    |__|_|_|_|  |__|_|_|_|  |__|_|_|    |__|_|_|_|  packet (U)  /
 layer:  4 3 2       4 3 2       4 3         4 3 2
 header
```
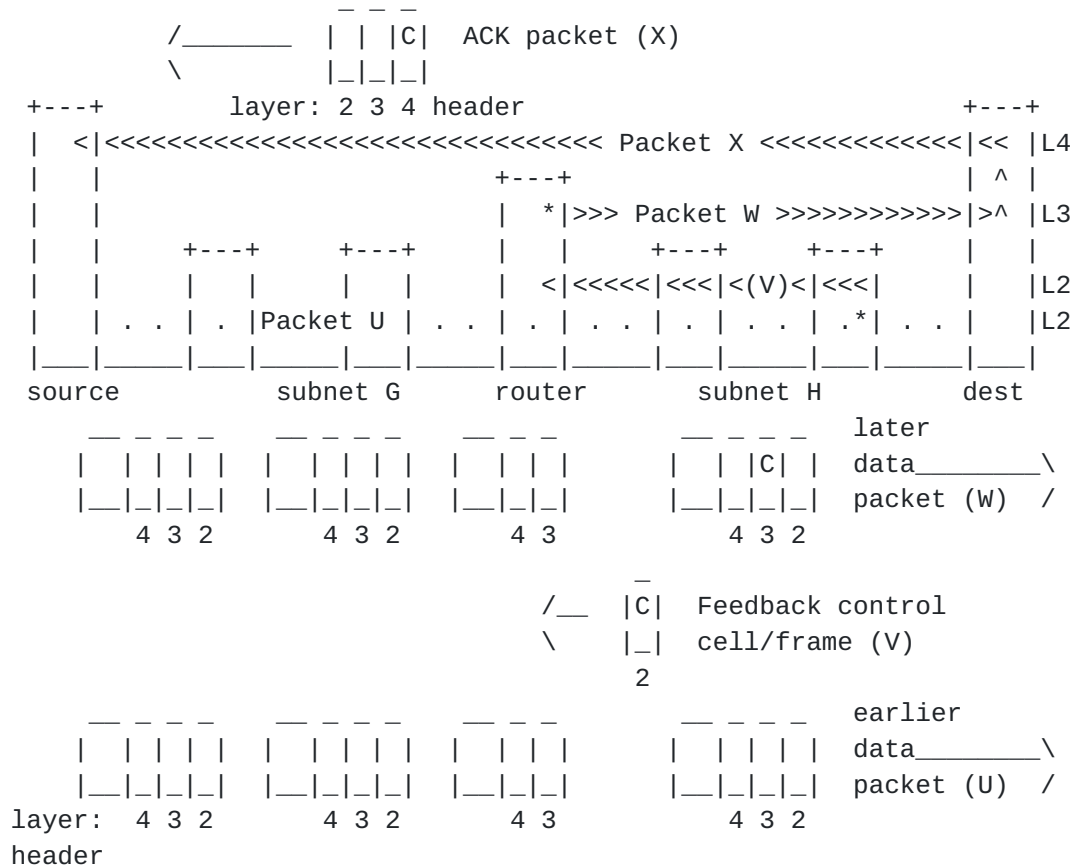
Figure 3: Feed-Backward Mode

ATM's feed-backward approach does not fit well when layered beneath
IP's feed-forward approach--unless the initial data source is the
same node as the ATM ingress. Figure 3 shows the feed-backward
approach being used in subnet H. If the final switch on the path is
congested (*), it does not feed-forward any congestion indications on
packet (U). Instead it sends a control cell (V) back to the router
at the ATM ingress.

However, the backward feedback does not reach the original data
source directly because IP does not support backward feedback (and
subnet G is independent of subnet H). Instead, the router in the
middle throttles down its sending rate but the original data sources
don't reduce their rates. The resulting rate mismatch causes the
middle router's buffer at layer 3 to back up until it becomes
congested, which it signals forwards on later data packets at layer 3
(e.g. packet W). Note that the forward signal from the middle router
is not triggered directly by the backward signal. Rather, it is
triggered by congestion resulting from the middle router's mismatched
rate response to the backward signal.

In response to this later forward signalling, end-to-end feedback at
layer-4 finally completes the tortuous path of congestion indications
back to the origin data source, as before.

Quantized congestion notification (QCN [IEEE802.1Q]) would suffer
from similar problems if extended to multiple subnets. However, from
the start QCN was clearly characterized as solely applicable to a
single subnet (see Section 6).

3.4. Null Mode

Often link and physical layer resources are 'non-blocking' by design.
In these cases congestion notification may be implemented but it does
not need to be deployed at the lower layer; ECN in IP would be
sufficient.

A degenerate example is a point-to-point Ethernet link. Excess
loading of the link merely causes the queue from the higher layer to
back up, while the lower layer remains immune to congestion. Even a
whole meshed subnetwork can be made immune to interior congestion by
limiting ingress capacity and sufficient sizing of interior links,
e.g. a non-blocking fat-tree network [Leiserson85]. An alternative
to fat links near the root is numerous thin links with multi-path
routing to ensure even worst-case patterns of load cannot congest any
link, e.g. a Clos network [Clos53].

4. Feed-Forward-and-Up Mode: Guidelines for Adding Congestion Notification

Feed-forward-and-up is the mode already used for signalling ECN up
the layers through MPLS into IP [RFC5129] and through IP-in-IP
tunnels [RFC6040], whether encapsulating with IPv4 [RFC2003], IPv6
[RFC2473] or IPsec [RFC4301]. These RFCs take a consistent approach
and the following guidelines are designed to ensure this consistency
continues as ECN support is added to other protocols that encapsulate
IP. The guidelines are also designed to ensure compliance with the
more general best current practice for the design of alternate ECN
schemes given in [RFC4774] and extended by [RFC8311].

The rest of this section is structured as follows:

o Section 4.1 addresses the most straightforward cases, where
   [RFC6040] can be applied directly to add ECN to tunnels that are
   effectively IP-in-IP tunnels, but with shim header(s) between the
   IP headers.

o The subsequent sections give guidelines for adding ECN to a subnet
   technology that uses feed-forward-and-up mode like IP, but it is

not so similar to IP that [RFC6040] rules can be applied directly.
Specifically:

* Sections 4.2, 4.3 and 4.4 respectively address how to add ECN
  support to the wire protocol and to the encapsulators and
  decapsulators at the ingress and egress of the subnet.

* Section 4.5 deals with the special, but common, case of
  sequences of tunnels or subnets that all use the same
  technology

* Section 4.6 deals with the question of reframing when IP
  packets do not map 1:1 into lower layer frames.

4.1. IP-in-IP Tunnels with Shim Headers

A common pattern for many tunnelling protocols is to encapsulate an
inner IP header with shim header(s) then an outer IP header. A shim
header is defined as one that is not sufficient alone to forward the
packet as an outer header. Another common pattern is for a shim to
encapsulate a layer 2 (L2) header, which in turn encapsulates (or
might encapsulate) an IP header. [I-D.ietf-tsvwg-rfc6040update-shim]
clarifies that RFC 6040 is just as applicable when there are shim(s)
and possibly a L2 header between two IP headers.

However, it is not always feasible or necessary to propagate ECN
between IP headers when separated by a shim. For instance, it might
be too costly to dig to arbitrary depths to find an inner IP header,
there may be little or no congestion within the tunnel by design (see
null mode in Section 3.4 above), or a legacy implementation might not
support ECN. In cases where a tunnel does not support ECN, it is
important that the ingress does not copy the ECN field from an inner
IP header to an outer. Therefore section 4 of
[I-D.ietf-tsvwg-rfc6040update-shim] requires network operators to
configure the ingress of a tunnel that does not support ECN so that
it zeros the ECN field in the outer IP header.

Nonetheless, in many cases it is feasible to propagate the ECN field
between IP headers separated by shim header(s) and/or a L2 header.
Particularly in the typical case when the outer IP header and the
shim(s) are added (or removed) as part of the same procedure. Even
if the shim(s) encapsulate a L2 header, it is often possible to find
an inner IP header within the L2 PDU and propagate ECN between that
and the outer IP header. This can be thought of as a special case of
the feed-up-and-forward mode (Section 3.2), so the guidelines for
this mode apply (Section 5).

Numerous shim protocols have been defined for IP tunnelling. More
recent ones e.g. Generic UDP Encapsulation (GUE)
[I-D.ietf-intarea-gue] and Geneve [I-D.ietf-nvo3-geneve] cite and
follow RFC 6040. And some earlier ones, e.g. CAPWAP [RFC5415] and
LISP [RFC6830], cite RFC 3168, which is compatible with RFC 6040.

However, as Section 9.3 of RFC 3168 pointed out, ECN support needs to
be defined for many earlier shim-based tunnelling protocols, e.g.
L2TPv2 [RFC2661], L2TPv3 [RFC3931], GRE [RFC2784], PPTP [RFC2637],
GTP [GTPv1], [GTPv1-U], [GTPv2-C] and Teredo [RFC4380] as well as
some recent ones, e.g. VXLAN [RFC7348], NVGRE [RFC7637] and NSH
[RFC8300].

All these IP-based encapsulations can be updated in one shot by
simple reference to RFC 6040. However, it would not be appropriate
to update all these protocols from within the present guidance
document. Instead a companion specification
[I-D.ietf-tsvwg-rfc6040update-shim] has been prepared that has the
appropriate standards track status to update standards track
protocols. For those that are not under IETF change control
[I-D.ietf-tsvwg-rfc6040update-shim] can only recommend that the
relevant body updates them.

4.2. Wire Protocol Design: Indication of ECN Support

This section is intended to guide the redesign of any lower layer
protocol that encapsulate IP to add native ECN support at the lower
layer. It reflects the approaches used in [RFC6040] and in
[RFC5129]. Therefore IP-in-IP tunnels or IP-in-MPLS or MPLS-in-MPLS
encapsulations that already comply with [RFC6040] or [RFC5129] will
already satisfy this guidance.

A lower layer (or subnet) congestion notification system:

1. SHOULD NOT apply explicit congestion notifications to PDUs that
    are destined for legacy layer-4 transport implementations that
    will not understand ECN, and

2. SHOULD NOT apply explicit congestion notifications to PDUs if the
    egress of the subnet might not propagate congestion notifications
    onward into the higher layer.

    We use the term ECN-PDUs for a PDU on a feedback loop that will
    propagate congestion notification properly because it meets both
    the above criteria. And a Not-ECN-PDU is a PDU on a feedback
    loop that does not meet at least one of the criteria, and will
    therefore not propagate congestion notification properly. A

corollary of the above is that a lower layer congestion
notification protocol:

3. SHOULD be able to distinguish ECN-PDUs from Not-ECN-PDUs.

Note that there is no need for all interior nodes within a subnet to
be able to mark congestion explicitly. A mix of ECN and drop signals
from different nodes is fine. However, if _any_ interior nodes might
generate ECN markings, guideline 2 above says that all relevant
egress node(s) SHOULD be able to propagate those markings up to the
higher layer.

In IP, if the ECN field in each PDU is cleared to the Not-ECT (not
ECN-capable transport) codepoint, it indicates that the L4 transport
will not understand congestion markings. A congested buffer must not
mark these Not-ECT PDUs, and therefore drops them instead.

The mechanism a lower layer uses to distinguish the ECN-capability of
PDUs need not mimic that of IP. The above guidelines merely say that
the lower layer system, as a whole, should achieve the same outcome.
For instance, ECN-capable feedback loops might use PDUs that are
identified by a particular set of labels or tags. Alternatively,
logical link protocols that use flow state might determine whether a
PDU can be congestion marked by checking for ECN-support in the flow
state. Other protocols might depend on out-of-band control signals.

The per-domain checking of ECN support in MPLS [RFC5129] is a good
example of a way to avoid sending congestion markings to L4
transports that will not understand them, without using any header
space in the subnet protocol.

In MPLS, header space is extremely limited, therefore RFC5129 does
not provide a field in the MPLS header to indicate whether the PDU is
an ECN-PDU or a Not-ECN-PDU. Instead, interior nodes in a domain are
allowed to set explicit congestion indications without checking
whether the PDU is destined for a L4 transport that will understand
them. Nonetheless, this is made safe by requiring that the network
operator upgrades all decapsulating edges of a whole domain at once,
as soon as even one switch within the domain is configured to mark
rather than drop during congestion. Therefore, any edge node that
might decapsulate a packet will be capable of checking whether the
higher layer transport is ECN-capable. When decapsulating a CE-
marked packet, if the decapsulator discovers that the higher layer
(inner header) indicates the transport is not ECN-capable, it drops
the packet--effectively on behalf of the earlier congested node (see
Decapsulation Guideline 1 in Section 4.4).

It was only appropriate to define such an incremental deployment
strategy because MPLS is targeted solely at professional operators,
who can be expected to ensure that a whole subnetwork is consistently
configured. This strategy might not be appropriate for other link
technologies targeted at zero-configuration deployment or deployment
by the general public (e.g. Ethernet). For such 'plug-and-play'
environments it will be necessary to invent a failsafe approach that
ensures congestion markings will never fall into black holes, no
matter how inconsistently a system is put together. Alternatively,
congestion notification relying on correct system configuration could
be confined to flavours of Ethernet intended only for professional
network operators, such as Provider Backbone Bridges (PBB
[IEEE802.1Q]; previously 802.1ah).

ECN support in TRILL [I-D.ietf-trill-ecn-support] provides a good
example of how to add ECN to a lower layer protocol without relying
on careful and consistent operator configuration. TRILL provides an
extension header word with space for flags of different categories
depending on whether logic to understand the extension is critical.
The congestion experienced marking has been defined as a 'critical
ingress-to-egress' flag. So if a transit RBridge sets this flag and
an egress RBridge does not have any logic to process it, it will drop
it; which is the desired default action anyway. Therefore TRILL
RBridges can be updated with support for ECN in no particular order
and, at the egress of the TRILL campus, congestion notification will
be propagated to IP as ECN whenever ECN logic has been implemented,
or as drop otherwise.

QCN [IEEE802.1Q] is not intended to extend beyond a single subnet, or
to interoperate with ECN. Nonetheless, the way QCN indicates to
lower layer devices that the end-points will not understand QCN
provides another example that a lower layer protocol designer might
be able to mimic for their scenario. An operator can define certain
Priority Code Points (PCPs [IEEE802.1Q]; previously 802.1p) to
indicate non-QCN frames and an ingress bridge is required to map
arriving not-QCN-capable IP packets to one of these non-QCN PCPs.

4.3. Encapsulation Guidelines

This section is intended to guide the redesign of any node that
encapsulates IP with a lower layer header when adding native ECN
support to the lower layer protocol. It reflects the approaches used
in [RFC6040] and in [RFC5129]. Therefore IP-in-IP tunnels or IP-in-
MPLS or MPLS-in-MPLS encapsulations that already comply with
[RFC6040] or [RFC5129] will already satisfy this guidance.

1. Egress Capability Check: A subnet ingress needs to be sure that
    the corresponding egress of a subnet will propagate any

congestion notification added to the outer header across the
subnet. This is necessary in addition to checking that an
incoming PDU indicates an ECN-capable (L4) transport. Examples
of how this guarantee might be provided include:

* by configuration (e.g. if any label switches in a domain
  support ECN marking, [RFC5129] requires all egress nodes to
  have been configured to propagate ECN)

* by the ingress explicitly checking that the egress propagates
  ECN (e.g. an early attempt to add ECN support to TRILL used
  IS-IS to check path capabilities before adding ECN extension
  flags to each frame [RFC7780]).

* by inherent design of the protocol (e.g. by encoding ECN
  marking on the outer header in such a way that a legacy egress
  that does not understand ECN will consider the PDU corrupt or
  invalid and discard it, thus at least propagating a form of
  congestion signal).

2. Egress Fails Capability Check: If the ingress cannot guarantee
   that the egress will propagate congestion notification, the
   ingress SHOULD disable ECN at the lower layer when it forwards
   the PDU. An example of how the ingress might disable ECN at the
   lower layer would be by setting the outer header of the PDU to
   identify it as a Not-ECN-PDU, assuming the subnet technology
   supports such a concept.

3. Standard Congestion Monitoring Baseline: Once the ingress to a
   subnet has established that the egress will correctly propagate
   ECN, on encapsulation it SHOULD encode the same level of
   congestion in outer headers as is arriving in incoming headers.
   For example it might copy any incoming congestion notification
   into the outer header of the lower layer protocol.

   This ensures that bulk congestion monitoring of outer headers
   (e.g. by a network management node monitoring ECN in passing
   frames) will measure congestion accumulated along the whole
   upstream path - since the Load Regulator not just since the
   ingress of the subnet. A node that is not the Load Regulator
   SHOULD NOT re-initialize the level of CE markings in the outer to
   zero.

   It would still also be possible to measure congestion introduced
   across one subnet (or tunnel) by subtracting the level of CE
   markings on inner headers from that on outer headers (see
   Appendix C of [RFC6040]). For example:

   * If this guideline has been followed and if the level of CE
      markings is 0.4% on the outer and 0.1% on the inner, 0.4%
      congestion has been introduced across all the networks since
      the load regulator, and 0.3% (= 0.4% - 0.1%) has been
      introduced since the ingress to the current subnet (or
      tunnel);

   * Without this guideline, if the subnet ingress had re-
      initialized the outer congestion level to zero, the outer and
      inner would measure 0.1% and 0.3%. It would still be possible
      to infer that the congestion introduced since the Load
      Regulator was 0.4% (= 0.1% + 0.3%). But only if the
      monitoring system somehow knows whether the subnet ingress re-
      initialized the congestion level.

   As long as subnet and tunnel technologies use the standard
   congestion monitoring baseline in this guideline, monitoring
   systems will know to use the former approach, rather than having
   to "somehow know" which approach to use.

4.4. Decapsulation Guidelines

   This section is intended to guide the redesign of any node that
   decapsulates IP from within a lower layer header when adding native
   ECN support to the lower layer protocol. It reflects the approaches
   used in [RFC6040] and in [RFC5129]. Therefore IP-in-IP tunnels or
   IP-in-MPLS or MPLS-in-MPLS encapsulations that already comply with
   [RFC6040] or [RFC5129] will already satisfy this guidance.

   A subnet egress SHOULD NOT simply copy congestion notification from
   outer headers to the forwarded header. It SHOULD calculate the
   outgoing congestion notification field from the inner and outer
   headers using the following guidelines. If there is any conflict,
   rules earlier in the list take precedence over rules later in the
   list:

   1. If the arriving inner header is a Not-ECN-PDU it implies the L4
       transport will not understand explicit congestion markings.
       Then:

       * If the outer header carries an explicit congestion marking,
          drop is the only indication of congestion that the L4
          transport will understand. If the congestion marking is the
          most severe possible, the packet MUST be dropped. However, if
          congestion can be marked with multiple levels of severity and
          the packet's marking is not the most severe, this requirement
          can be relaxed to: the packet SHOULD be dropped.

       * If the outer is an ECN-PDU that carries no indication of
         congestion or a Not-ECN-PDU the PDU SHOULD be forwarded, but
         still as a Not-ECN-PDU.

2.   If the outer header does not support explicit congestion
    notification (a Not-ECN-PDU), but the inner header does (an ECN-
    PDU), the inner header SHOULD be forwarded unchanged.

3.   In some lower layer protocols congestion may be signalled as a
    numerical level, such as in the control frames of quantized
    congestion notification (QCN [IEEE802.1Q]). If such a multi-bit
    encoding encapsulates an ECN-capable IP data packet, a function
    will be needed to convert the quantized congestion level into the
    frequency of congestion markings in outgoing IP packets.

4.   Congestion indications might be encoded by a severity level. For
    instance increasing levels of congestion might be encoded by
    numerically increasing indications, e.g. pre-congestion
    notification (PCN) can be encoded in each PDU at three severity
    levels in IP or MPLS [RFC6660] and the default encapsulation and
    decapsulation rules [RFC6040] are compatible with this
    interpretation of the ECN field.

    If the arriving inner header is an ECN-PDU, where the inner and
    outer headers carry indications of congestion of different
    severity, the more severe indication SHOULD be forwarded in
    preference to the less severe.

5.   The inner and outer headers might carry a combination of
    congestion notification fields that should not be possible given
    any currently used protocol transitions. For instance, if
    Encapsulation Guideline 3 in Section 4.3 had been followed, it
    should not be possible to have a less severe indication of
    congestion in the outer than in the inner. It MAY be appropriate
    to log unexpected combinations of headers and possibly raise an
    alarm.

    If a safe outgoing codepoint can be defined for such a PDU, the
    PDU SHOULD be forwarded rather than dropped. Some implementers
    discard PDUs with currently unused combinations of headers just
    in case they represent an attack. However, an approach using
    alarms and policy-mediated drop is preferable to hard-coded drop,
    so that operators can keep track of possible attacks but
    currently unused combinations are not precluded from future use
    through new standards actions.

4.5. Sequences of Similar Tunnels or Subnets

   In some deployments, particularly in 3GPP networks, an IP packet may
   traverse two or more IP-in-IP tunnels in sequence that all use
   identical technology (e.g. GTP).

   In such cases, it would be sufficient for every encapsulation and
   decapsulation in the chain to comply with RFC 6040. Alternatively,
   as an optimisation, a node that decapsulates a packet and immediately
   re-encapsulates it for the next tunnel MAY copy the incoming outer
   ECN field directly to the outgoing outer and the incoming inner ECN
   field directly to the outgoing inner. Then the overall behavior
   across the sequence of tunnel segments would still be consistent with
   RFC 6040.

   Appendix C of RFC6040 describes how a tunnel egress can monitor how
   much congestion has been introduced within a tunnel. A network
   operator might want to monitor how much congestion had been
   introduced within a whole sequence of tunnels. Using the technique
   in Appendix C of RFC6040 at the final egress, the operator could
   monitor the whole sequence of tunnels, but only if the above
   optimisation were used consistently along the sequence of tunnels, in
   order to make it appear as a single tunnel. Therefore, tunnel
   endpoint implementations SHOULD allow the operator to configure
   whether this optimisation is enabled.

   When ECN support is added to a subnet technology, consideration
   SHOULD be given to a similar optimisation between subnets in sequence
   if they all use the same technology.

4.6. Reframing and Congestion Markings

   The guidance in this section is worded in terms of framing
   boundaries, but it applies equally whether the protocol data units
   are frames, cells or packets.

   Where an AQM marks the ECN field of IP packets as they queue into a
   layer-2 link, there will be no problem with framing boundaries,
   because the ECN markings would be applied directly to IP packets.
   The guidance in this section is only applicable where an ECN
   capability is being added to a layer-2 protocol so that layer-2
   frames can be ECN-marked by an AQM at layer-2. This would only be
   necessary where AQM will be applied at pure layer-2 nodes (without
   IP-awareness). Where framing boundaries do not necessarily align
   with packet boundaries, the following guidance will be needed. It
   explains how to propagate ECN markings from layer-2 frame headers
   when they are stripped off and IP PDUs with different boundaries are
   reassembled for forwarding.

Congestion indications SHOULD be propagated on the basis that an
encapsulator or decapsulator SHOULD approximately preserve the
proportion of PDUs with congestion indications arriving and leaving.

The mechanism for propagating congestion indications SHOULD ensure
that any incoming congestion indication is propagated immediately,
not held awaiting the possibility of further congestion indications
to be sufficient to indicate congestion on an outgoing PDU.

5. Feed-Up-and-Forward Mode: Guidelines for Adding Congestion Notification

The guidance in this section is applicable, for example, when IP
packets:

o are encapsulated in Ethernet headers, which have no support for
   ECN;

o are forwarded by the eNode-B (base station) of a 3GPP radio access
   network, which is required to apply ECN marking during congestion,
   [LTE-RA], [UTRAN], but the Packet Data Convergence Protocol (PDCP)
   that encapsulates the IP header over the radio access has no
   support for ECN.

This guidance also generalizes to encapsulation by other subnet
technologies with no native support for explicit congestion
notification at the lower layer, but with support for finding and
processing an IP header. It is unlikely to be applicable or
necessary for IP-in-IP encapsulation, where feed-forward-and-up mode
based on [RFC6040] would be more appropriate.

Marking the IP header while switching at layer-2 (by using a layer-3
switch) or while forwarding in a radio access network seems to
represent a layering violation. However, it can be considered as a
benign optimisation if the guidelines below are followed. Feed-up-
and-forward is certainly not a general alternative to implementing
feed-forward congestion notification in the lower layer, because:

o IPv4 and IPv6 are not the only layer-3 protocols that might be
   encapsulated by lower layer protocols

o Link-layer encryption might be in use, making the layer-2 payload
   inaccessible

o Many Ethernet switches do not have 'layer-3 switch' capabilities
   so they cannot read or modify an IP payload

o   It might be costly to find an IP header (v4 or v6) when it may be
    encapsulated by more than one lower layer header, e.g. Ethernet
    MAC in MAC ([IEEE802.1Q]; previously 802.1ah).

Nonetheless, configuring lower layer equipment to look for an ECN
field in an encapsulated IP header is a useful optimisation. If the
implementation follows the guidelines below, this optimisation does
not have to be confined to a controlled environment such as within a
data centre; it could usefully be applied on any network--even if the
operator is not sure whether the above issues will never apply:

1. If a native lower-layer congestion notification mechanism exists
    for a subnet technology, it is safe to mix feed-up-and-forward
    with feed-forward-and-up on other switches in the same subnet.
    However, it will generally be more efficient to use the native
    mechanism.

2. The depth of the search for an IP header SHOULD be limited. If
    an IP header is not found soon enough, or an unrecognized or
    unreadable header is encountered, the switch SHOULD resort to an
    alternative means of signalling congestion (e.g. drop, or the
    native lower layer mechanism if available).

3. It is sufficient to use the first IP header found in the stack;
    the egress of the relevant tunnel can propagate congestion
    notification upwards to any more deeply encapsulated IP headers
    later.

6. Feed-Backward Mode: Guidelines for Adding Congestion Notification

It can be seen from Section 3.3 that congestion notification in a
subnet using feed-backward mode has generally not been designed to be
directly coupled with IP layer congestion notification. The subnet
attempts to minimize congestion internally, and if the incoming load
at the ingress exceeds the capacity somewhere through the subnet, the
layer 3 buffer into the ingress backs up. Thus, a feed-backward mode
subnet is in some sense similar to a null mode subnet, in that there
is no need for any direct interaction between the subnet and higher
layer congestion notification. Therefore no detailed protocol design
guidelines are appropriate. Nonetheless, a more general guideline is
appropriate:

   A subnetwork technology intended to eventually interface to IP
   SHOULD NOT be designed using only the feed-backward mode, which is
   certainly best for a stand-alone subnet, but would need to be
   modified to work efficiently as part of the wider Internet,
   because IP uses feed-forward-and-up mode.

The feed-backward approach at least works beneath IP, where the term
'works' is used only in a narrow functional sense because feed-
backward can result in very inefficient and sluggish congestion
control--except if it is confined to the subnet directly connected to
the original data source, when it is faster than feed-forward. It
would be valid to design a protocol that could work in feed-backward
mode for paths that only cross one subnet, and in feed-forward-and-up
mode for paths that cross subnets.

In the early days of TCP/IP, a similar feed-backward approach was
tried for explicit congestion signalling, using source-quench (SQ)
ICMP control packets. However, SQ fell out of favour and is now
formally deprecated [RFC6633]. The main problem was that it is hard
for a data source to tell the difference between a spoofed SQ message
and a quench request from a genuine buffer on the path. It is also
hard for a lower layer buffer to address an SQ message to the
original source port number, which may be buried within many layers
of headers, and possibly encrypted.

QCN (also known as backward congestion notification, BCN; see
Sections 30--33 of [IEEE802.1Q]; previously known as 802.1Qau) uses a
feed-backward mode structurally similar to ATM's relative rate
mechanism. However, QCN confines its applicability to scenarios such
as some data centres where all endpoints are directly attached by the
same Ethernet technology. If a QCN subnet were later connected into
a wider IP-based internetwork (e.g. when attempting to interconnect
multiple data centres) it would suffer the inefficiency shown in
Figure 3.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

If a lower layer wire protocol is redesigned to include explicit
congestion signalling in-band in the protocol header, care SHOULD be
take to ensure that the field used is specified as mutable during
transit. Otherwise interior nodes signalling congestion would
invalidate any authentication protocol applied to the lower layer
header--by altering a header field that had been assumed as
immutable.

The redesign of protocols that encapsulate IP in order to propagate
congestion signals between layers raises potential signal integrity
concerns. Experimental or proposed approaches exist for assuring the
end-to-end integrity of in-band congestion signals, e.g.:

o Congestion exposure (ConEx) for networks to audit that their
   congestion signals are not being suppressed by other networks or
   by receivers, and for networks to police that senders are
   responding sufficiently to the signals, irrespective of the L4
   transport protocol used [RFC7713].

o A test for a sender to detect whether a network or the receiver is
   suppressing congestion signals (for example see 2nd para of
   Section 20.2 of [RFC3168]).

Given these end-to-end approaches are already being specified, it
would make little sense to attempt to design hop-by-hop congestion
signal integrity into a new lower layer protocol, because end-to-end
integrity inherently achieves hop-by-hop integrity.

Section 6 gives vulnerability to spoofing as one of the reasons for
deprecating feed-backward mode.

9. Conclusions

Following the guidance in this document enables ECN support to be
extended to numerous protocols that encapsulate IP (v4 & v6) in a
consistent way, so that IP continues to fulfil its role as an end-to-
end interoperability layer. This includes:

o A wide range of tunnelling protocols including those with various
   forms of shim header between two IP headers, possibly also
   separated by a L2 header;

o A wide range of subnet technologies, particularly those that work
   in the same 'feed-forward-and-up' mode that is used to support ECN
   in IP and MPLS.

Guidelines have been defined for supporting propagation of ECN
between Ethernet and IP on so-called Layer-3 Ethernet switches, using
a 'feed-up-and-forward' mode. This approach could enable other
subnet technologies to pass ECN signals into the IP layer, even if
they do not support ECN natively.

Finally, attempting to add ECN to a subnet technology in feed-
backward mode is deprecated except in special cases, due to its
likely sluggish response to congestion.

10. Acknowledgements

Thanks to Gorry Fairhurst and David Black for extensive reviews.
Thanks also to the following reviewers: Joe Touch, Andrew McGregor,
Richard Scheffenegger, Ingemar Johansson, Piers O'Hanlon, Donald

11. Contributors

     Pat Thaler
     Broadcom Corporation (retired)
     CA
     USA

Pat was a co-author of this draft, but retired before its
publication.

12. Comments Solicited

Comments and questions are encouraged and very welcome. They can be
addressed to the IETF Transport Area working group mailing list
<tsvwg@ietf.org>, and/or to the authors.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119,
           DOI 10.17487/RFC2119, March 1997,
           <https://www.rfc-editor.org/info/rfc2119>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition
           of Explicit Congestion Notification (ECN) to IP",
           RFC 3168, DOI 10.17487/RFC3168, September 2001,
           <https://www.rfc-editor.org/info/rfc3168>.

[RFC3819] Karn, P., Ed., Bormann, C., Fairhurst, G., Grossman, D.,
           Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L.
           Wood, "Advice for Internet Subnetwork Designers", BCP 89,
           RFC 3819, DOI 10.17487/RFC3819, July 2004,
           <https://www.rfc-editor.org/info/rfc3819>.

[RFC4774] Floyd, S., "Specifying Alternate Semantics for the
           Explicit Congestion Notification (ECN) Field", BCP 124,
           RFC 4774, DOI 10.17487/RFC4774, November 2006,
           <https://www.rfc-editor.org/info/rfc4774>.

[RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion
           Marking in MPLS", RFC 5129, DOI 10.17487/RFC5129, January
           2008, <https://www.rfc-editor.org/info/rfc5129>.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion
           Notification", RFC 6040, DOI 10.17487/RFC6040, November
           2010, <https://www.rfc-editor.org/info/rfc6040>.

13.2. Informative References

[ATM-TM-ABR]
           Cisco, "Understanding the Available Bit Rate (ABR) Service
           Category for ATM VCs", Design Technote 10415, June 2005.

[Buck00]   Buckwalter, J., "Frame Relay: Technology and Practice",
           Pub. Addison Wesley ISBN-13: 978-0201485240, 2000.

[Clos53]   Clos, C., "A Study of Non-Blocking Switching Networks",
           Bell Systems Technical Journal 32(2):406--424, March 1953.

[GTPv1]   3GPP, "GPRS Tunnelling Protocol (GTP) across the Gn and Gp
           interface", Technical Specification TS 29.060.

[GTPv1-U] 3GPP, "General Packet Radio System (GPRS) Tunnelling
           Protocol User Plane (GTPv1-U)", Technical Specification TS
           29.281.

[GTPv2-C] 3GPP, "Evolved General Packet Radio Service (GPRS)
           Tunnelling Protocol for Control plane (GTPv2-C)",
           Technical Specification TS 29.274.

[I-D.ietf-intarea-gue]
           Herbert, T., Yong, L., and O. Zia, "Generic UDP
           Encapsulation", draft-ietf-intarea-gue-09 (work in
           progress), October 2019.

   [I-D.ietf-nvo3-geneve]
              Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic
              Network Virtualization Encapsulation", draft-ietf-
              nvo3-geneve-16 (work in progress), March 2020.

   [I-D.ietf-trill-ecn-support]
              Eastlake, D. and B. Briscoe, "TRILL (TRansparent
              Interconnection of Lots of Links): ECN (Explicit
              Congestion Notification) Support", draft-ietf-trill-ecn-
              support-06 (work in progress), February 2018.

   [I-D.ietf-tsvwg-ecn-l4s-id]
              Schepper, K. and B. Briscoe, "Identifying Modified
              Explicit Congestion Notification (ECN) Semantics for
              Ultra-Low Queuing Delay (L4S)", draft-ietf-tsvwg-ecn-l4s-
              id-12 (work in progress), November 2020.

   [I-D.ietf-tsvwg-rfc6040update-shim]
              Briscoe, B., "Propagating Explicit Congestion Notification
              Across IP Tunnel Headers Separated by a Shim", draft-ietf-
              tsvwg-rfc6040update-shim-12 (work in progress), November
              2020.

   [IEEE802.1Q]
              IEEE, "IEEE Standard for Local and Metropolitan Area
              Networks--Virtual Bridged Local Area Networks--Amendment
              6: Provider Backbone Bridges", IEEE Std 802.1Q-2018, July
              2018, <https://ieeexplore.ieee.org/document/8403927>.

   [ITU-T.I.371]
              ITU-T, "Traffic Control and Congestion Control in B-ISDN",
              ITU-T Rec. I.371 (03/04), March 2004,
              <http://ieeexplore.ieee.org/xpl/
              mostRecentIssue.jsp?punumber=5454061>.

   [Leiserson85]
              Leiserson, C., "Fat-trees: universal networks for
              hardware-efficient supercomputing", IEEE Transactions on
              Computers 34(10):892-901, October 1985.

   [LTE-RA]   3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA)
              and Evolved Universal Terrestrial Radio Access Network
              (E-UTRAN); Overall description; Stage 2", Technical
              Specification TS 36.300.

   [RFC2003]   Perkins, C., "IP Encapsulation within IP", RFC 2003,
              DOI 10.17487/RFC2003, October 1996,
              <https://www.rfc-editor.org/info/rfc2003>.

   [RFC2473]   Conta, A. and S. Deering, "Generic Packet Tunneling in
               IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473,
               December 1998, <https://www.rfc-editor.org/info/rfc2473>.

   [RFC2637]   Hamzeh, K., Pall, G., Verthein, W., Taarud, J., Little,
               W., and G. Zorn, "Point-to-Point Tunneling Protocol
               (PPTP)", RFC 2637, DOI 10.17487/RFC2637, July 1999,
               <https://www.rfc-editor.org/info/rfc2637>.

   [RFC2661]   Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn,
               G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"",
               RFC 2661, DOI 10.17487/RFC2661, August 1999,
               <https://www.rfc-editor.org/info/rfc2661>.

   [RFC2784]   Farinacci, D., Li, T., Hanks, S., Meyer, D., and P.
               Traina, "Generic Routing Encapsulation (GRE)", RFC 2784,
               DOI 10.17487/RFC2784, March 2000,
               <https://www.rfc-editor.org/info/rfc2784>.

   [RFC2884]   Hadi Salim, J. and U. Ahmed, "Performance Evaluation of
               Explicit Congestion Notification (ECN) in IP Networks",
               RFC 2884, DOI 10.17487/RFC2884, July 2000,
               <https://www.rfc-editor.org/info/rfc2884>.

   [RFC2983]   Black, D., "Differentiated Services and Tunnels",
               RFC 2983, DOI 10.17487/RFC2983, October 2000,
               <https://www.rfc-editor.org/info/rfc2983>.

   [RFC3931]   Lau, J., Ed., Townsley, M., Ed., and I. Goyret, Ed.,
               "Layer Two Tunneling Protocol - Version 3 (L2TPv3)",
               RFC 3931, DOI 10.17487/RFC3931, March 2005,
               <https://www.rfc-editor.org/info/rfc3931>.

   [RFC4301]   Kent, S. and K. Seo, "Security Architecture for the
               Internet Protocol", RFC 4301, DOI 10.17487/RFC4301,
               December 2005, <https://www.rfc-editor.org/info/rfc4301>.

   [RFC4380]   Huitema, C., "Teredo: Tunneling IPv6 over UDP through
               Network Address Translations (NATs)", RFC 4380,
               DOI 10.17487/RFC4380, February 2006,
               <https://www.rfc-editor.org/info/rfc4380>.

   [RFC5415]   Calhoun, P., Ed., Montemurro, M., Ed., and D. Stanley,
               Ed., "Control And Provisioning of Wireless Access Points
               (CAPWAP) Protocol Specification", RFC 5415,
               DOI 10.17487/RFC5415, March 2009,
               <https://www.rfc-editor.org/info/rfc5415>.

   [RFC6633] Gont, F., "Deprecation of ICMP Source Quench Messages",
              RFC 6633, DOI 10.17487/RFC6633, May 2012,
              <https://www.rfc-editor.org/info/rfc6633>.

   [RFC6660] Briscoe, B., Moncaster, T., and M. Menth, "Encoding Three
              Pre-Congestion Notification (PCN) States in the IP Header
              Using a Single Diffserv Codepoint (DSCP)", RFC 6660,
              DOI 10.17487/RFC6660, July 2012,
              <https://www.rfc-editor.org/info/rfc6660>.

   [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The
              Locator/ID Separation Protocol (LISP)", RFC 6830,
              DOI 10.17487/RFC6830, January 2013,
              <https://www.rfc-editor.org/info/rfc6830>.

   [RFC7323] Borman, D., Braden, B., Jacobson, V., and R.
              Scheffenegger, Ed., "TCP Extensions for High Performance",
              RFC 7323, DOI 10.17487/RFC7323, September 2014,
              <https://www.rfc-editor.org/info/rfc7323>.

   [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger,
              L., Sridhar, T., Bursell, M., and C. Wright, "Virtual
              eXtensible Local Area Network (VXLAN): A Framework for
              Overlaying Virtualized Layer 2 Networks over Layer 3
              Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014,
              <https://www.rfc-editor.org/info/rfc7348>.

   [RFC7567] Baker, F., Ed. and G. Fairhurst, Ed., "IETF
              Recommendations Regarding Active Queue Management",
              BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015,
              <https://www.rfc-editor.org/info/rfc7567>.

   [RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network
              Virtualization Using Generic Routing Encapsulation",
              RFC 7637, DOI 10.17487/RFC7637, September 2015,
              <https://www.rfc-editor.org/info/rfc7637>.

   [RFC7713] Mathis, M. and B. Briscoe, "Congestion Exposure (ConEx)
              Concepts, Abstract Mechanism, and Requirements", RFC 7713,
              DOI 10.17487/RFC7713, December 2015,
              <https://www.rfc-editor.org/info/rfc7713>.

   [RFC7780] Eastlake 3rd, D., Zhang, M., Perlman, R., Banerjee, A.,
              Ghanwani, A., and S. Gupta, "Transparent Interconnection
              of Lots of Links (TRILL): Clarifications, Corrections, and
              Updates", RFC 7780, DOI 10.17487/RFC7780, February 2016,
              <https://www.rfc-editor.org/info/rfc7780>.

   [RFC8084] Fairhurst, G., "Network Transport Circuit Breakers",
              BCP 208, RFC 8084, DOI 10.17487/RFC8084, March 2017,
              <https://www.rfc-editor.org/info/rfc8084>.

   [RFC8087] Fairhurst, G. and M. Welzl, "The Benefits of Using
              Explicit Congestion Notification (ECN)", RFC 8087,
              DOI 10.17487/RFC8087, March 2017,
              <https://www.rfc-editor.org/info/rfc8087>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

   [RFC8257] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L.,
              and G. Judd, "Data Center TCP (DCTCP): TCP Congestion
              Control for Data Centers", RFC 8257, DOI 10.17487/RFC8257,
              October 2017, <https://www.rfc-editor.org/info/rfc8257>.

   [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed.,
              "Network Service Header (NSH)", RFC 8300,
              DOI 10.17487/RFC8300, January 2018,
              <https://www.rfc-editor.org/info/rfc8300>.

   [RFC8311] Black, D., "Relaxing Restrictions on Explicit Congestion
              Notification (ECN) Experimentation", RFC 8311,
              DOI 10.17487/RFC8311, January 2018,
              <https://www.rfc-editor.org/info/rfc8311>.

   [UTRAN]   3GPP, "UTRAN Overall Description", Technical
              Specification TS 25.401.

Appendix A. Changes in This Version (to be removed by RFC Editor)

   From ietf-12 to ietf-13

      * Following 3rd tsvwg WGLC:

         + Formalized update to RFC 3819 in its own subsection (1.1)
            and referred to it in the abstract

         + Scope: Clarified that the specification of alternative ECN
            semantics using ECT(1) was not in RFC 4774, but rather in
            RFC 8311, and that the problem with using a DSCP to indicate
            alternative semantics has issues at domain boundaries as
            well as tunnels.

         + Terminology: tighted up definitions of ECN-PDU and Not-ECN-
            PDU, and removed definition of Congestion Baseline, given it
            was only used once.

         + Mentioned QCN where feed-backward is first introduced (S.3),
            referring forward to where it is discussed more deeply
            (S.4).

         + Clarified that IS-IS solution to adding ECN support to TRILL
            was not pursued

         + Completely rewrote the rationale for the guideline about a
            Standard Congestion Monitoring Baseline, to focus on
            standardization of the otherwise unknown scenario used,
            rather than the relative usefulness of the info in each
            approach

         + Explained the re-framing problem better and added
            fragmentation as another possible cause of the problem

         + Acknowledged new reviewers

         + Updated references, replaced citations of 802.1Qau and
            802.1ah with rolled up 802.1Q, and added citations of Fat
            trees and Clos Networks

         + Numerous other editorial improvements

   From ietf-11 to ietf-12

      * Updated references

   From ietf-10 to ietf-11

   *   Removed short section (was 3) 'Guidelines for All Cases'
       because it was out of scope, being covered by RFC 4774.
       Expanded the Scope section (1.2) to explain all this.
       Explained that the default encap/decap rules already support
       certain alternative semantics, particularly all three of the
       alternative semantics for ECT(1): equivalent to ECT(0) , higher
       severity than ECT(0), and unmarked but implying different
       marking semantics from ECT(0).

   *   Clarified why the QCN example was being given even though not
       about increment deployment of ECN

   *   Pointed to the spoofing issue with feed-backward mode from the
       Security Considerations section, to aid security review.

   *   Removed any ambiguity in the word 'transport' throughout

   From ietf-09 to ietf-10

   *   Updated section 5.1 on "IP-in-IP tunnels with Shim Headers" to
       be consistent with updates to draft-ietf-tsvwg-rfc6040update-
       shim.

   *   Removed reference to the ECN nonce, which has been made
       historic by RFC 8311

   *   Removed "Open Issues" Appendix, given all have been addressed.

   From ietf-08 to ietf-09

   *   Updated para in Intro that listed all the IP-in-IP tunnelling
       protocols, to instead refer to draft-ietf-tsvwg-rfc6040update-
       shim

   *   Updated section 5.1 on "IP-in-IP tunnels with Shim Headers" to
       summarize guidance that has evolved as rfc6040update-shim has
       developed.

   From ietf-07 to ietf-08: Refreshed to avoid expiry. Updated
      references.

   From ietf-06 to ietf-07:

   *   Added the people involved in liaisons to the acknowledgements.

   From ietf-05 to ietf-06:

   * Introduction: Added GUE and Geneve as examples of tightly
      coupled shims between IP headers that cite RFC 6040. And added
      VXLAN to list of those that do not.

   * Replaced normative text about tightly coupled shims between IP
      headers, with reference to new draft-ietf-tsvwg-rfc6040update-
      shim

   * Wire Protocol Design: Indication of ECN Support: Added TRILL as
      an example of a well-design protocol that does not need an
      indication of ECN support in the wire protocol.

   * Encapsulation Guidelines: In the case of a Not-ECN-PDU with a
      CE outer, replaced SHOULD be dropped, with explanations of when
      SHOULD or MUST are appropriate.

   * Feed-Up-and-Forward Mode: Explained examples more carefully,
      referred to PDCP and cited UTRAN spec as well as E-UTRAN.

   * Updated references.

   * Marked open issues as resolved, but did not delete Open Issues
      Appendix (yet).

   From ietf-04 to ietf-05:

   * Explained why tightly coupled shim headers only "SHOULD" comply
      with RFC 6040, not "MUST".

   * Updated references

   From ietf-03 to ietf-04:

   * Addressed Richard Scheffenegger's review comments: primarily
      editorial corrections, and addition of examples for clarity.

   From ietf-02 to ietf-03:

   * Updated references, ad cited RFC4774.

   From ietf-01 to ietf-02:

   * Added Section for guidelines that are applicable in all cases.

   * Updated references.

   From ietf-00 to ietf-01: Updated references.

From briscoe-04 to ietf-00: Changed filename following tsvwg
    adoption.

From briscoe-03 to 04:

   * Re-arranged the introduction to describe the purpose of the
      document first before introducing ECN in more depth. And
      clarified the introduction throughout.

   * Added applicability to 3GPP TS 36.300.

From briscoe-02 to 03:

   * Scope section:

      + Added dependence on correct propagation of traffic class
         information

      + For the feed-backward mode, deemed multicast and anycast out
         of scope

   * Ensured all guidelines referring to subnet technologies also
      refer to tunnels and vice versa by adding applicability
      sentences at the start of sections 4.1, 4.2, 4.3, 4.4, 4.6 and
      5.

   * Added Security Considerations on ensuring congestion signal
      fields are classed as immutable and on using end-to-end
      congestion signal integrity technologies rather than hop-by-
      hop.

From briscoe-01 to 02:

   * Added authors: JK & PT

   * Added

      + Section 4.1 "IP-in-IP Tunnels with Tightly Coupled Shim
         Headers"

      + Section 4.5 "Sequences of Similar Tunnels or Subnets"

      + roadmap at the start of Section 4, given the subsections
         have become quite fragmented.

      + Section 9 "Conclusions"

* Clarified why transports are starting to be able to saturate
  interior links

* Under Section 1.1, addressed the question of alternative signal
  semantics and included multicast & anycast.

* Under Section 3.1, included a 3GPP example.

* Section 4.2. "Wire Protocol Design":

    + Altered guideline 2. to make it clear that it only applies
      to the immediate subnet egress, not later ones

    + Added a reminder that it is only necessary to check that ECN
      propagates at the egress, not whether interior nodes mark
      ECN

    + Added example of how QCN uses 802.1p to indicate support for
      QCN.

* Added references to Appendix C of RFC6040, about monitoring the
  amount of congestion signals introduced within a tunnel

* Appendix A: Added more issues to be addressed, including plan
  to produce a standards track update to IP-in-IP tunnel
  protocols.

* Updated acks and references

From briscoe-00 to 01:

* Intended status: BCP (was Informational) & updates 3819 added.

* Briefer Introduction: Introductory para justifying benefits of
  ECN. Moved all but a brief enumeration of modes of operation
  to their own new section (from both Intro & Scope). Introduced
  incr. deployment as most tricky part.

* Tightened & added to terminology section

* Structured with Modes of Operation, then Guidelines section for
  each mode.

* Tightened up guideline text to remove vagueness / passive voice
  / ambiguity and highlight main guidelines as numbered items.

* Added Outstanding Document Issues Appendix

      * Updated references

Authors' Addresses

   Bob Briscoe
   Independent
   UK

   EMail: ietf@bobbriscoe.net
   URI: http://bobbriscoe.net/

   John Kaippallimalil
   Futurewei
   5700 Tennyson Parkway, Suite 600
   Plano, Texas 75024
   USA

   EMail: kjohn@futurewei.com