                 Tunnelling of Explicit Congestion Notification
                      draft-ietf-tsvwg-ecn-tunnel-05

Abstract

   This document redefines how the explicit congestion notification
   (ECN) field of the IP header should be constructed on entry to and
   exit from any IP in IP tunnel. On encapsulation it updates RFC3168
   to bring all IP in IP tunnels (v4 or v6) into line with RFC4301 IPsec
   ECN processing. On decapsulation it updates both RFC3168 and RFC4301
   to add new behaviours for previously unused combinations of inner and
   outer header. The new rules ensure the ECN field is correctly
   propagated across a tunnel whether it is used to signal one or two
   severity levels of congestion, whereas before only one severity level
   was supported. Tunnel endpoints can be updated in any order without
   affecting pre-existing uses of the ECN field (backward compatible).
   Nonetheless, operators wanting to support two severity levels (e.g.
   for pre-congestion notification--PCN) can require compliance with
   this new specification. A thorough analysis of the reasoning for
   these changes and the implications is included. In the unlikely
   event that the new rules do not meet a specific need, RFC4774 gives
   guidance on designing alternate ECN semantics and this document
   extends that to include tunnelling issues.

Status of This Memo

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

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as Internet-
   Drafts.

   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."

   The list of current Internet-Drafts can be accessed at

http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html.

This Internet-Draft will expire on June 21, 2010.

Copyright Notice

Table of Contents

Request to the RFC Editor (to be removed on publication):

   In the RFC index, RFC3168 should be identified as an update to
   RFC2003.   RFC4301 should be identified as an update to RFC3168.

Changes from previous drafts (to be removed by the RFC Editor)

   Full text differences between IETF draft versions are available at
   <http://tools.ietf.org/wg/tsvwg/draft-ietf-tsvwg-ecn-tunnel/>, and
   between earlier individual draft versions at
   <http://www.briscoe.net/pubs.html#ecn-tunnel>

   From ietf-04 to ietf-05 (current):

      * Functional changes:

         + Section 4.2: ECT(1) outer with Not-ECT inner: reverted to
            forwarding as Not-ECT (as in RFC3168 & RFC4301), rather than
            dropping.

         + Altered rationale in bullet 3 of Section 5.3.2 to justify
            this.

         + Distinguished alarms for dangerous and invalid combinations
            and allowed combinations that are valid in some tunnel
            configurations but dangerous in others to be alarmed at the
            discretion of the implementer and/or operator.

         + Altered advice on designing alternate ECN tunnelling
            semantics to reflect the above changes.

      * Textual changes:

         + Changed "Future non-default schemes" to "Alternate ECN
            Tunnelling Semantics" throughout.

         + Cut down Appendix D and Appendix E for brevity.

         + A number of clarifying edits & updated refs.

   From ietf-03 to ietf-04:

      * Functional changes: none

      * Structural changes:

         + Added "Open Issues" appendix

   * Textual changes:

      + Section title: "Changes from Earlier RFCs" -> "Updates to
         Earlier RFCs"

      + Emphasised that change on decap to previously unused
         combinations will propagate PCN encoding.

      + Acknowledged additional reviewers and updated references

   From ietf-02 to ietf-03:

   * Functional changes:

      + Corrected errors in recap of previous RFCs, which wrongly
         stated the different decapsulation behaviours of RFC3168 &
         RFC4301 with a Not-ECT inner header. This also required
         corrections to the "Changes from Earlier RFCs" and the
         Motivations for these changes.

      + Mandated that any future standards action SHOULD NOT use the
         ECT(0) codepoint as an indication of congestion, without
         giving strong reasons.

      + Added optional alarm when decapsulating ECT(1) outer,
         ECT(0), but noted it would need to be disabled for
         2-severity level congestion (e.g. PCN).

   * Structural changes:

      + Removed Document Roadmap which merely repeated the Contents
         (previously Section 1.2).

      + Moved "Changes from Earlier RFCs" (Section 5) before
         Section 6 on Backward Compatibility and internally organised
         both by RFC, rather than by ingress then egress.

      + Moved motivation for changing existing RFCs (Section 5.3) to
         after the changes are specified.

      + Moved informative "Design Principles for Future Non-Default
         Schemes" after all the normative sections.

      + Added Appendix A on early history of ECN tunnelling RFCs.

      + Removed specialist appendix on "Relative Placement of
         Tunnelling and In-Path Load Regulation" (Appendix D in the
         -02 draft)

+ Moved and updated specialist text on "Compromise on Decap with ECT(1) Inner and ECT(0) Outer" from Security Considerations to Appendix F

* Textual changes:

+ Simplified vocabulary for non-native-english speakers

+ Simplified Introduction and defined regularly used terms in an expanded Terminology section.

+ More clearly distinguished statically configured tunnels from dynamic tunnel endpoint discovery, before explaining operating modes.

+ Simplified, cut-down and clarified throughout

+ Updated references.

From ietf-01 to ietf-02:

* Scope reduced from any encapsulation of an IP packet to solely IP in IP tunnelled encapsulation. Consequently changed title and removed whole section 'Design Guidelines for New Encapsulations of Congestion Notification' (to be included in a future companion informational document).

* Included a new normative decapsulation rule for ECT(0) inner and ECT(1) outer that had previously only been outlined in the non-normative appendix 'Comprehensive Decapsulation Rules'. Consequently:

+ The Introduction has been completely re-written to motivate this change to decapsulation along with the existing change to encapsulation.

+ The tentative text in the appendix that first proposed this change has been split between normative standards text in Section 4 and Appendix D, which explains specifically why this change would streamline PCN. New text on the logic of the resulting decap rules added.

* If inner/outer is Not-ECT/ECT(0), changed decapsulation to propagate Not-ECT rather than drop the packet; and added reasoning.

* Considerably restructured:

+ "Design Constraints" analysis moved to an appendix
  (Appendix B);

+ Added Section 3 to summarise relevant existing RFCs;

+ Structured Section 4 and Section 6 into subsections.

+ Added tables to sections on old and new rules, for precision
  and comparison.

+ Moved Section 7 on Design Principles to the end of the
  section specifying the new default normative tunnelling
  behaviour. Rewritten and shifted text on identifiers and
  in-path load regulators to Appendix B.1 [deleted in revision
  -03].

From ietf-00 to ietf-01:

* Identified two additional alarm states in the decapsulation
  rules (Figure 4) if ECT(X) in outer and inner contradict each
  other.

* Altered Comprehensive Decapsulation Rules (Appendix D) so that
  ECT(0) in the outer no longer overrides ECT(1) in the inner.
  Used the term 'Comprehensive' instead of 'Ideal'. And
  considerably updated the text in this appendix.

* Added Appendix D.1 (removed again in a later revision) to weigh
  up the various ways the Comprehensive Decapsulation Rules might
  be introduced. This replaces the previous contradictory
  statements saying complex backwards compatibility interactions
  would be introduced while also saying there would be no
  backwards compatibility issues.

* Updated references.

From briscoe-01 to ietf-00:

* Re-wrote Appendix C giving much simpler technique to measure
  contribution to congestion across a tunnel.

* Added discussion of backward compatibility of the ideal
  decapsulation scheme in Appendix D

* Updated references. Minor corrections & clarifications
  throughout.

From briscoe-00 to briscoe-01:

* Related everything conceptually to the uniform and pipe models
  of RFC2983 on Diffserv Tunnels, and completely removed the
  dependence of tunnelling behaviour on the presence of any in-
  path load regulation by using the [1 - Before] [2 - Outer]
  function placement concepts from RFC2983;

* Added specific cases where the existing standards limit new
  proposals, particularly Appendix E;

* Added sub-structure to Introduction (Need for Rationalisation,
  Roadmap), added new Introductory subsection on "Scope" and
  improved clarity;

* Added Design Guidelines for New Encapsulations of Congestion
  Notification;

* Considerably clarified the Backward Compatibility section
  (Section 6);

* Considerably extended the Security Considerations section
  (Section 8);

* Summarised the primary rationale much better in the
  conclusions;

* Added numerous extra acknowledgements;

* Added Appendix E. "Why resetting CE on encapsulation harms
  PCN", Appendix C. "Contribution to Congestion across a Tunnel"
  and Appendix D. "Ideal Decapsulation Rules";

* Re-wrote Appendix B [deleted in a later revision], explaining
  how tunnel encapsulation no longer depends on in-path load-
  regulation (changed title from "In-path Load Regulation" to
  "Non-Dependence of Tunnelling on In-path Load Regulation"), but
  explained how an in-path load regulation function must be
  carefully placed with respect to tunnel encapsulation (in a new
  sub-section entitled "Dependence of In-Path Load Regulation on
  Tunnelling").

1. Introduction

Explicit congestion notification (ECN [RFC3168]) allows a forwarding
element to notify the onset of congestion without having to drop
packets. Instead it can explicitly mark a proportion of packets in
the 2-bit ECN field in the IP header (Table 1 recaps the ECN

codepoints).

The outer header of an IP packet can encapsulate one or more IP
headers for tunnelling. A forwarding element using ECN to signify
congestion will only mark the immediately visible outer IP header.
When a tunnel decapsulator later removes this outer header, it
follows rules to propagate congestion markings by combining the ECN
fields of the inner and outer IP header into one outgoing IP header.

This document updates those rules for IPsec [RFC4301] and non-IPsec
[RFC3168] tunnels to add new behaviours for previously unused
combinations of inner and outer header. It also updates the tunnel
ingress behaviour of RFC3168 to match that of RFC4301. The updated
rules are backward compatible with RFC4301 and RFC3168 when
interworking with any other tunnel endpoint complying with any
earlier specification.

When ECN and its tunnelling was defined in RFC3168, only the minimum
necessary changes to the ECN field were propagated through tunnel
endpoints--just enough for the basic ECN mechanism to work. This was
due to concerns that the ECN field might be toggled to communicate
between a secure site and someone on the public Internet--a covert
channel. This was because a mutable field like ECN cannot be
protected by IPsec's integrity mechanisms--it has to be able to
change as it traverses the Internet.

Nonetheless, the latest IPsec architecture [RFC4301] considers a
bandwidth limit of 2 bits per packet on a covert channel makes it a
manageable risk. Therefore, for simplicity, an RFC4301 ingress
copies the whole ECN field to encapsulate a packet. It also
dispenses with the two modes of RFC3168, one which partially copied
the ECN field, and the other which blocked all propagation of ECN
changes.

Unfortunately, this entirely reasonable sequence of standards actions
resulted in a perverse outcome; non-IPsec tunnels (RFC3168) blocked
the 2-bit covert channel, while IPsec tunnels (RFC4301) did not--at
least not at the ingress. At the egress, both IPsec and non-IPsec
tunnels still partially restricted propagation of the full ECN field.

The trigger for the changes in this document was the introduction of
pre-congestion notification (PCN [RFC5670]) to the IETF standards
track. PCN needs the ECN field to be copied at a tunnel ingress and
it needs four states of congestion signalling to be propagated at the
egress, but pre-existing tunnels only propagate three in the ECN
field.

This document draws on currently unused (CU) combinations of inner

and outer headers to add tunnelling of four-state congestion
signalling to RFC3168 and RFC4301. Operators of tunnels who
specifically want to support four states can require that all their
tunnels comply with this specification. Nonetheless, all tunnel
endpoint implementations (RFC4301, RFC3168, RFC2481, RFC2401,
RFC2003) can safely be updated to this new specification as part of
general code maintenance. This will gradually add support for four
congestion states to the Internet. Existing three state schemes will
continue to work as before.

At the same time as harmonising covert channel constraints, the
opportunity has been taken to draw together diverging tunnel
specifications into a single consistent behaviour. Then any tunnel
can be deployed unilaterally, and it will support the full range of
congestion control and management schemes without any modes or
configuration. Further, any host or router can expect the ECN field
to behave in the same way, whatever type of tunnel might intervene in
the path.

1.1. Scope

This document only concerns wire protocol processing of the ECN field
at tunnel endpoints and makes no changes or recommendations
concerning algorithms for congestion marking or congestion response.

This document specifies common ECN field processing at encapsulation
and decapsulation for any IP in IP tunnelling, whether IPsec or non-
IPsec tunnels. It applies irrespective of whether IPv4 or IPv6 is
used for either of the inner and outer headers. It applies for
packets with any destination address type, whether unicast or
multicast. It applies as the default for all Diffserv per-hop
behaviours (PHBs), unless stated otherwise in the specification of a
PHB. It is intended to be a good trade off between somewhat
conflicting security, control and management requirements.

[RFC2983] is a comprehensive primer on differentiated services and
tunnels. Given ECN raises similar issues to differentiated services
when interacting with tunnels, useful concepts introduced in RFC2983
are used throughout, with brief recaps of the explanations where
necessary.

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 RFC 2119 [RFC2119].

Table 1 recaps the names of the ECN codepoints [RFC3168].

```
+------------------+---------------+--------------------------+
| Binary codepoint | Codepoint name | Meaning                 |
+------------------+---------------+--------------------------+
|         00       | Not-ECT       | Not ECN-capable transport |
|         01       | ECT(1)        | ECN-capable transport    |
|         10       | ECT(0)        | ECN-capable transport    |
|         11       | CE            | Congestion experienced   |
+------------------+---------------+--------------------------+
```

Table 1: Recap of Codepoints of the ECN Field [RFC3168] in the IP
                              Header

Further terminology used within this document:

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

Decapsulator: The tunnel endpoint function that removes an outer IP
   header from a tunnelled packet (also termed the 'egress tunnel
   endpoint' or just the 'egress' where the context is clear).

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

Outer header: The header added to encapsulate a tunnelled packet.

Inner header: The header encapsulated by the outer header.

Outgoing header: The header constructed by the decapsulator using
   logic that combines the fields in the outer and inner headers.

Copying ECN: On encapsulation, setting the ECN field of the new
   outer header to be a copy of the ECN field in the incoming header.

Zeroing ECN: On encapsulation, clearing the ECN field of the new
   outer header to Not-ECT ("00").

Resetting ECN: On encapsulation, setting the ECN field of the new
   outer header to be a copy of the ECN field in the incoming header
   except the outer ECN field is set to the ECT(0) codepoint if the
   incoming ECN field is CE ("11").

3. Summary of Pre-Existing RFCs

This section is informative not normative, as it recaps pre-existing
RFCs. Earlier relevant RFCs that were either experimental or
incomplete with respect to ECN tunnelling (RFC2481, RFC2401 and

RFC2003) are briefly outlined in Appendix A. The question of whether
tunnel implementations used in the Internet comply with any of these
RFCs is not discussed.

3.1. Encapsulation at Tunnel Ingress

At the encapsulator, the controversy has been over whether to
propagate information about congestion experienced on the path so far
into the outer header of the tunnel.

Specifically, RFC3168 says that, if a tunnel fully supports ECN
(termed a 'full-functionality' ECN tunnel in [RFC3168]), the
encapsulator must not copy a CE marking from the inner header into
the outer header that it creates. Instead the encapsulator must set
the outer header to ECT(0) if the ECN field is marked CE in the
arriving IP header. We term this 'resetting' a CE codepoint.

However, the new IPsec architecture in [RFC4301] reverses this rule,
stating that the encapsulator must simply copy the ECN field from the
incoming header to the outer header.

RFC3168 also provided a Limited Functionality mode that turns off ECN
processing over the scope of the tunnel by setting the outer header
to Not-ECT ("00"). Then such packets will be dropped to indicate
congestion rather than marked with ECN. This is necessary for the
ingress to interwork with legacy decapsulators ([RFC2481], [RFC2401]
and [RFC2003]) that do not propagate ECN markings added to the outer
header. Otherwise such legacy decapsulators would throw away
congestion notifications before they reached the transport layer.

Neither Limited Functionality mode nor Full Functionality mode are
used by an RFC4301 IPsec encapsulator, which simply copies the
incoming ECN field into the outer header. An earlier key-exchange
phase ensures an RFC4301 ingress will not have to interwork with a
legacy egress that does not support ECN.

These pre-existing behaviours are summarised in Figure 1.

```
+----------------+---------------------------------------------------+
| Incoming Header |              Outgoing Outer Header               |
| (also equal to  +---------------+---------------+---------------+
| Outgoing Inner  |  RFC3168 ECN  |  RFC3168 ECN  |  RFC4301 IPsec |
|     Header)     |    Limited    |     Full      |               |
|                 | Functionality | Functionality |               |
+----------------+---------------+---------------+---------------+
|    Not-ECT     |    Not-ECT    |    Not-ECT    |    Not-ECT    |
|    ECT(0)      |    Not-ECT    |    ECT(0)     |    ECT(0)     |
|    ECT(1)      |    Not-ECT    |    ECT(1)     |    ECT(1)     |
|      CE        |    Not-ECT    |    ECT(0)     |      CE       |
+---------------+---------------+---------------+---------------+
```

Figure 1: IP in IP Encapsulation: Recap of Pre-existing Behaviours

3.2. Decapsulation at Tunnel Egress

RFC3168 and RFC4301 specify the decapsulation behaviour summarised in
Figure 2. The ECN field in the outgoing header is set to the
codepoint at the intersection of the appropriate incoming inner
header (row) and incoming outer header (column).

```
         +---------+-------------------------------------------------+
         |Incoming |              Incoming Outer Header              |
         |   Inner +---------+-----------+-----------+-----------+
         |  Header | Not-ECT | ECT(0)    | ECT(1)    |    CE     |
         +---------+---------+-----------+-----------+-----------+
RFC3168->| Not-ECT | Not-ECT |Not-ECT    |Not-ECT    |   drop    |
RFC4301->| Not-ECT | Not-ECT |Not-ECT    |Not-ECT    |Not-ECT    |
         | ECT(0)  | ECT(0)  | ECT(0)    | ECT(0)    |    CE     |
         | ECT(1)  | ECT(1)  | ECT(1)    | ECT(1)    |    CE     |
         |   CE    |    CE   |    CE     |    CE     |    CE     |
         +---------+---------+-----------+-----------+-----------+
                   |              Outgoing Header                  |
         +-------------------------------------------------+
```

Figure 2: IP in IP Decapsulation; Recap of Pre-existing Behaviour

The behaviour in the table derives from the logic given in RFC3168
and RFC4301, briefly recapped as follows:

o On decapsulation, if the inner ECN field is Not-ECT the outer is
   discarded. RFC3168 (but not RFC4301) also specified that the
   decapsulator must drop a packet with a Not-ECT inner and CE in the
   outer.

o In all other cases, if the outer is CE, the outgoing ECN field is
   set to CE, but otherwise the outer is ignored and the inner is

used for the outgoing ECN field.

RFC3168 also made it an auditable event for an IPsec tunnel "if the
ECN Field is changed inappropriately within an IPsec tunnel...".
Inappropriate changes were not specifically enumerated. RFC4301 did
not mention inappropriate ECN changes.

4. New ECN Tunnelling Rules

The standards actions below in Section 4.1 (ingress encapsulation)
and Section 4.2 (egress decapsulation) define new default ECN tunnel
processing rules for any IP packet (v4 or v6) with any Diffserv
codepoint.

If unavoidable, an alternate congestion encapsulation behaviour can
be introduced as part of the definition of an alternate congestion
marking scheme used by a specific Diffserv PHB (see S.5 of [RFC3168]
and [RFC4774]). When designing such new encapsulation schemes, the
principles in Section 7 should be followed. However, alternate ECN
tunnelling schemes are NOT RECOMMENDED as the deployment burden of
handling exceptional PHBs in implementations of all affected tunnels
should not be underestimated. There is no requirement for a PHB
definition to state anything about ECN tunnelling behaviour if the
default behaviour in the present specification is sufficient.

4.1. Default Tunnel Ingress Behaviour

Two modes of encapsulation are defined here; `normal mode' and
`compatibility mode', which is for backward compatibility with tunnel
decapsulators that do not understand ECN. Section 4.3 explains why
two modes are necessary and specifies the circumstances in which it
is sufficient to solely implement normal mode. Note that these are
modes of the ingress tunnel endpoint only, not the whole tunnel.

Whatever the mode, an encapsulator forwards the inner header without
changing the ECN field.

In normal mode an encapsulator compliant with this specification MUST
construct the outer encapsulating IP header by copying the 2-bit ECN
field of the incoming IP header. In compatibility mode it clears the
ECN field in the outer header to the Not-ECT codepoint. These rules
are tabulated for convenience in Figure 3.

```
        +----------------+------------------------------+
        | Incoming Header |      Outgoing Outer Header   |
        | (also equal to  +--------------+---------------+
        | Outgoing Inner  | Compatibility |    Normal    |
        |     Header)     |     Mode      |     Mode     |
        +----------------+--------------+---------------+
        |    Not-ECT     |   Not-ECT    |    Not-ECT    |
        |     ECT(0)     |   Not-ECT    |    ECT(0)     |
        |     ECT(1)     |   Not-ECT    |    ECT(1)     |
        |       CE       |   Not-ECT    |      CE       |
        +----------------+--------------+---------------+
```

        Figure 3: New IP in IP Encapsulation Behaviours

   An ingress in compatibility mode encapsulates packets identically to
   an ingress in RFC3168's limited functionality mode. An ingress in
   normal mode encapsulates packets identically to an RFC4301 IPsec
   ingress.

4.2. Default Tunnel Egress Behaviour

   To decapsulate the inner header at the tunnel egress, a compliant
   tunnel egress MUST set the outgoing ECN field to the codepoint at the
   intersection of the appropriate incoming inner header (row) and outer
   header (column) in Figure 4 (the IPv4 header checksum also changes
   whenever the ECN field is changed). There is no need for more than
   one mode of decapsulation, as these rules cater for all known
   requirements.

```
        +---------+-------------------------------------------------+
        |Incoming |              Incoming Outer Header               |
        |   Inner +---------+-----------+-----------+------------+
        | Header  | Not-ECT | ECT(0)    | ECT(1)    |     CE     |
        +---------+---------+-----------+-----------+------------+
        | Not-ECT | Not-ECT |Not-ECT(!!!)|Not-ECT(!!!)|  drop(!!!)|
        |  ECT(0) |  ECT(0) | ECT(0)    | ECT(1)    |     CE     |
        |  ECT(1) |  ECT(1) | ECT(1) (!) | ECT(1)    |     CE     |
        |    CE   |      CE |    CE     |   CE(!!!)|     CE     |
        +---------+---------+-----------+-----------+------------+
        |                   Outgoing Header                      |
        +--------------------------------------------------------+
```
         Unexpected combinations are indicated by '(!!!)'

        Figure 4: New IP in IP Decapsulation Behaviour

   This table for decapsulation behaviour is derived from the following
   logic:

o If the inner ECN field is Not-ECT the decapsulator MUST NOT
   propagate any other ECN codepoint onwards. This is because the
   inner Not-ECT marking is set by transports that use drop as an
   indication of congestion and would not understand or respond to
   any other ECN codepoint [RFC4774]. In addition:

   * If the inner ECN field is Not-ECT and the outer ECN field is CE
      the decapsulator MUST drop the packet.

   * If the inner ECN field is Not-ECT and the outer ECN field is
      Not-ECT, ECT(0) or ECT(1) the decapsulator MUST forward the
      outgoing packet with the ECN field cleared to Not-ECT.

o In all other cases where the inner supports ECN, the outgoing ECN
   field is set to the more severe marking of the outer and inner ECN
   fields, where the ranking of severity from highest to lowest is
   CE, ECT(1), ECT(0), Not-ECT. This in no way precludes cases where
   ECT(1) and ECT(0) have the same severity;

o Certain combinations of inner and outer ECN fields cannot result
   from any currently used transition in any current or previous ECN
   tunneling specification. These cases are indicated in Figure 4 by
   '(!!!)' or '(!)', where '(!!!)' means the combination is both
   invalid and always potentially dangerous, while '(!)' means it is
   invalid and possibly dangerous. In these cases, particularly the
   more dangerous ones, the decapsulator SHOULD log the event and MAY
   also raise an alarm. Just because the highlighted combinations
   are always invalid, does not mean that all the other combinations
   are always valid. Some are only valid if they have arrived from a
   particular type of legacy ingress, and dangerous otherwise.
   Therefore an implementation MAY allow an operator to configure
   logging and alarms for such additional header combinations known
   to be dangerous or invalid for the particular configuration of
   tunnel endpoints deployed at run-time.

   Alarms should be rate-limited so that the illegal combinations
   will not amplify into a flood of alarm messages. It MUST be
   possible to suppress alarms or logging, e.g. if it becomes
   apparent that a combination that previously was not used has
   started to be used for legitimate purposes such as a new standards
   action.

The above logic allows for ECT(0) and ECT(1) to both represent the
same severity of congestion marking (e.g. "not congestion marked").
But it also allows future schemes to be defined where ECT(1) is a
more severe marking than ECT(0), in particular enabling the simplest
possible encoding for PCN [I-D.ietf-pcn-3-in-1-encoding]. This
approach is discussed in Appendix D and in the discussion of the ECN

nonce [RFC3540] in Section 8, which in turn refers to Appendix F.

4.3. Encapsulation Modes

Section 4.1 introduces two encapsulation modes, normal mode and
compatibility mode, defining their encapsulation behaviour (i.e.
header copying or zeroing respectively). Note that these are modes
of the ingress tunnel endpoint only, not the tunnel as a whole.

A tunnel ingress MUST at least implement `normal mode' and, if it
might be used with legacy tunnel egress nodes (RFC2003, RFC2401 or
RFC2481 or the limited functionality mode of RFC3168), it MUST also
implement `compatibility mode' for backward compatibility with tunnel
egresses that do not propagate explicit congestion notifications
[RFC4774]. If the egress does support propagation of ECN (full
functionality mode of RFC3168 or RFC4301 or the present
specification), the ingress SHOULD use normal mode, in order to
support ECN where possible.

We can categorise the way that an ingress tunnel endpoint is paired
with an egress as either:

static: those paired together by prior configuration or;

dynamically discovered: those paired together by some form of tunnel
   endpoint discovery, typically driven by the path taken by arriving
   packets.

Static: Some implementations of encapsulator might be constrained to
be statically deployed, and constrained to never be paired with a
legacy decapsulator (RFC2003, RFC2401 or RFC2481 or the limited
functionality mode of RFC3168). In such a case, only normal mode
needs to be implemented.

For instance, RFC4301-compatible IPsec tunnel endpoints invariably
use IKEv2 [RFC4306] for key exchange, which was introduced alongside
RFC4301. Therefore both endpoints of an RFC4301 tunnel can be sure
that the other end is RFC4301-compatible, because the tunnel is only
formed after IKEv2 key management has completed, at which point both
ends will be RFC4301-compliant by definition. Further, an RFC4301
encapsulator behaves identically to the normal mode of the present
specification and does not need to implement compatibility mode as it
will never interact with legacy ECN tunnels.

Dynamic Discovery: This specification does not require or recommend
dynamic discovery and it does not define how dynamic negotiation
might be done, but it recognises that proprietary tunnel endpoint
discovery protocols exist. It therefore sets down some constraints

on discovery protocols to ensure safe interworking.

If dynamic tunnel endpoint discovery might pair an ingress with a
legacy egress (RFC2003, RFC2401 or RFC2481 or the limited
functionality mode of RFC3168), the ingress MUST implement both
normal and compatibility mode. If the tunnel discovery process is
arranged to only ever find a tunnel egress that propagates ECN
(RFC3168 full functionality mode, RFC4301 or this present
specification), then a tunnel ingress can be complaint with the
present specification without implementing compatibility mode.

If a compliant tunnel ingress is discovering an egress, it MUST send
packets in compatibility mode in case the egress it discovers is a
legacy egress. If, through the discovery protocol, the egress
indicates that it is compliant with the present specification, with
RFC4301 or with RFC3168 full functionality mode, the ingress can
switch itself into normal mode. If the egress denies compliance with
any of these or returns an error that implies it does not understand
a request to work to any of these ECN specifications, the tunnel
ingress MUST remain in compatibility mode.

An ingress cannot claim compliance with this specification simply by
disabling ECN processing across the tunnel (i.e. only implementing
compatibility mode). It is true that such a tunnel ingress is at
least safe with the ECN behaviour of any egress it may encounter, but
it does not meet the aim of introducing ECN support to tunnels.

Implementation note: if a compliant node is the ingress for multiple
tunnels, a mode setting will need to be stored for each tunnel
ingress. However, if a node is the egress for multiple tunnels, none
of the tunnels will need to store a mode setting, because a compliant
egress can only be in one mode.

4.4. Single Mode of Decapsulation

A compliant decapsulator only has one mode of operation. However, if
a complaint egress is implemented to be dynamically discoverable, it
may need to respond to discovery requests from various types of
legacy tunnel ingress. This specification does not define how
dynamic negotiation might be done by (proprietary) discovery
protocols, but it sets down some constraints to ensure safe
interworking.

Through the discovery protocol, a tunnel ingress compliant with the
present specification might ask if the egress is compliant with the
present specification, with RFC4301 or with RFC3168 full
functionality mode. Or an RFC3168 tunnel ingress might try to
negotiate to use limited functionality or full functionality mode

[RFC3168]. In all these cases, a decapsulating tunnel egress
compliant with this specification MUST agree to any of these
requests, since it will behave identically in all these cases.

If no ECN-related mode is requested, a compliant tunnel egress MUST
continue without raising any error or warning as its egress behaviour
is compatible with all the legacy ingress behaviours that do not
negotiate capabilities.

For 'forward compatibility', a compliant tunnel egress SHOULD raise a
warning alarm about any requests to enter modes it does not
recognise, but it SHOULD continue operating.

5. Updates to Earlier RFCs

5.1. Changes to RFC4301 ECN processing

Ingress: An RFC4301 IPsec encapsulator is not changed at all by the
   present specification

Egress: The new decapsulation behaviour in Figure 4 updates RFC4301.
   However, it solely updates combinations of inner and outer that
   would never result from any protocol defined in the RFC series so
   far, even though they were catered for in RFC4301 for
   completeness. Therefore, the present specification adds new
   behaviours to RFC4301 decapsulation without altering existing
   behaviours. The following specific updates have been made:

   * The outer, not the inner, is propagated when the outer is
      ECT(1) and the inner is ECT(0);

   * A packet with Not-ECT in the inner and an outer of CE is
      dropped rather than forwarded as Not-ECT;

   * Certain combinations of inner and outer ECN field have been
      identified as currently unused. These can trigger logging
      and/or raise alarms.

Modes: RFC4301 does not need modes and is not updated by the modes
   in the present specification. The normal mode of encapsulation is
   unchanged from RFC4301 encapsulation and an RFC4301 IPsec ingress
   will never need compatibility mode as explained in Section 4.3
   (except in one corner-case described below).
   One corner case can exist where an RFC4301 ingress does not use
   IKEv2, but uses manual keying instead. Then an RFC4301 ingress
   could conceivably be configured to tunnel to an egress with
   limited functionality ECN handling. Strictly, for this corner-
   case, the requirement to use compatibility mode in this

specification updates RFC4301. However, this is such a remote
possibility that RFC4301 IPsec implementations are NOT REQUIRED to
implement compatibility mode.

5.2. Changes to RFC3168 ECN processing

Ingress: On encapsulation, the new rule in Figure 3 that a normal
   mode tunnel ingress copies any ECN field into the outer header
   updates the ingress behaviour of RFC3168. Nonetheless, the new
   compatibility mode is identical to the limited functionality mode
   of RFC3168.

Egress: The new decapsulation behaviour in Figure 4 updates RFC3168.
   However, the present specification solely updates combinations of
   inner and outer that would never result from any protocol defined
   in the RFC series so far, even though they were catered for in
   RFC4301 for completeness. Therefore, the present specification
   adds new behaviours to RFC3168 decapsulation without altering
   existing behaviours. The following specific updates have been
   made:

   * The outer, not the inner, is propagated when the outer is
      ECT(1) and the inner is ECT(0);

   * Certain combinations of inner and outer ECN field have been
      identified as currently unused. These can trigger logging
      and/or raise alarms.

Modes: RFC3168 defines a (required) limited functionality mode and
   an (optional) full functionality mode for a tunnel. In RFC3168,
   modes applied to both ends of the tunnel, while in the present
   specification, modes are only used at the ingress--a single egress
   behaviour covers all cases. The normal mode of encapsulation
   updates the encapsulation behaviour of the full functionality mode
   of RFC3168. The compatibility mode of encapsulation is identical
   to the encapsulation behaviour of the limited functionality mode
   of RFC3168. The constraints on how tunnel discovery protocols set
   modes in Section 4.3 and Section 4.4 are an update to RFC3168.

5.3. Motivation for Changes

An overriding goal is to ensure the same ECN signals can mean the
same thing whatever tunnels happen to encapsulate an IP packet flow.
This removes gratuitous inconsistency, which otherwise constrains the
available design space and makes it harder to design networks and new
protocols that work predictably.

5.3.1. Motivation for Changing Encapsulation

The normal mode in Section 4 updates RFC3168 to make all IP in IP encapsulation of the ECN field consistent--consistent with the way both RFC4301 IPsec [RFC4301] and IP in MPLS or MPLS in MPLS encapsulation [RFC5129] construct the ECN field.

Compatibility mode has also been defined so a non-RFC4301 ingress can still switch to using drop across a tunnel for backwards compatibility with legacy decapsulators that do not propagate ECN correctly.

The trigger that motivated this update to RFC3168 encapsulation was a standards track proposal for pre-congestion notification (PCN [RFC5670]). PCN excess rate marking only works correctly if the ECN field is copied on encapsulation (as in RFC4301 and RFC5129); it does not work if ECN is reset (as in RFC3168). This is because PCN excess rate marking depends on the outer header revealing any congestion experienced so far on the whole path, not just since the last tunnel ingress (see Appendix E for a full explanation).

PCN allows a network operator to add flow admission and termination for inelastic traffic at the edges of a Diffserv domain, but without any per-flow mechanisms in the interior and without the generous provisioning typical of Diffserv, aiming to significantly reduce costs. The PCN architecture [RFC5559] states that RFC3168 IP in IP tunnelling of the ECN field cannot be used for any tunnel ingress in a PCN domain. Prior to the present specification, this left a stark choice between not being able to use PCN for inelastic traffic control or not being able to use the many tunnels already deployed for Mobile IP, VPNs and so forth.

The present specification provides a clean solution to this problem, so that network operators who want to use both PCN and tunnels can specify that every tunnel ingress in a PCN region must comply with this latest specification.

Rather than allow tunnel specifications to fragment further into one for PCN, one for IPsec and one for other tunnels, the opportunity has been taken to consolidate the diverging specifications back into a single tunnelling behaviour. Resetting ECN was originally motivated by a covert channel concern that has been deliberately set aside in RFC4301 IPsec. Therefore the reset behaviour of RFC3168 is an anomaly that we do not need to keep. Copying ECN on encapsulation is anyway simpler than resetting. So, as more tunnel endpoints comply with this single consistent specification, encapsulation will be simpler as well as more predictable.

Appendix B assesses whether copying rather than resetting CE on
ingress will cause any unintended side-effects, from the three
perspectives of security, control and management. In summary this
analysis finds that:

o From the control perspective either copying or resetting works for
   existing arrangements, but copying has more potential for
   simplifying control and resetting breaks at least one proposal
   already on the standards track.

o From the management and monitoring perspective copying is
   preferable.

o From the traffic security perspective (enforcing congestion
   control, mitigating denial of service etc) copying is preferable.

o From the information security perspective resetting is preferable,
   but the IETF Security Area now considers copying acceptable given
   the bandwidth of a 2-bit covert channel can be managed.

Therefore there are two points against resetting CE on ingress while
copying CE causes no significant harm.

5.3.2. Motivation for Changing Decapsulation

The specification for decapsulation in Section 4 fixes three problems
with the pre-existing behaviours of both RFC3168 and RFC4301:

1. The pre-existing rules prevented the introduction of alternate
    ECN semantics to signal more than one severity level of
    congestion [RFC4774], [RFC5559]. The four states of the 2-bit
    ECN field provide room for signalling two severity levels in
    addition to not-congested and not-ECN-capable states. But, the
    pre-existing rules assumed that two of the states (ECT(0) and
    ECT(1)) are always equivalent. This unnecessarily restricts the
    use of one of four codepoints (half a bit) in the IP (v4 & v6)
    header. The new rules are designed to work in either case;
    whether ECT(1) is more severe than or equivalent to ECT(0).

    As explained in Appendix B.1, the original reason for not
    forwarding the outer ECT codepoints was to limit the covert
    channel across a decapsulator to 1 bit per packet. However, now
    that the IETF Security Area has deemed that a 2-bit covert
    channel through an encapsulator is a manageable risk, the same
    should be true for a decapsulator.

    As well as being useful for general future-proofing, this problem
    is immediately pressing for standardisation of pre-congestion

notification (PCN), which uses two severity levels of congestion.
If a congested queue used ECT(1) in the outer header to signal
more severe congestion than ECT(0), the pre-existing
decapsulation rules would have thrown away this congestion
signal, preventing tunnelled traffic from ever knowing that it
should reduce its load.

The PCN working group has had to consider a number of wasteful or
convoluted work-rounds to this problem (see Appendix D). But by
far the simplest approach is just to remove the covert channel
blockages from tunnelling behaviour--now deemed unnecessary
anyway. Then network operators that want to support two
congestion severity-levels for PCN can specify that every tunnel
egress in a PCN region must comply with this latest
specification.

Not only does this make two congestion severity-levels available
for PCN standardisation, but also for other potential uses of the
extra ECN codepoint (e.g. [VCP]).

2. Cases are documented where a middlebox (e.g. a firewall) drops
    packets with header values that were currently unused (CU) when
    the box was deployed, often on the grounds that anything
    unexpected might be an attack. This tends to bar future use of
    CU values. The new decapsulation rules specify optional logging
    and/or alarms for specific combinations of inner and outer header
    that are currently unused. The aim is to give implementers a
    recourse other than drop if they are concerned about the security
    of CU values. It recognises legitimate security concerns about
    CU values but still eases their future use. If the alarms are
    interpreted as an attack (e.g. by a management system) the
    offending packets can be dropped. But alarms can be turned off
    if these combinations come into regular use (e.g. a through a
    future standards action).

3. While reviewing currently unused combinations of inner and outer,
    the opportunity was taken to define a single consistent behaviour
    for the three cases with a Not-ECT inner header but a different
    outer. RFC3168 and RFC4301 had diverged in this respect. None
    of these combinations should result from Internet protocols in
    the RFC series, but future standards actions might put any or all
    of them to good use. Therefore it was decided that a
    decapsulator must forward a Not-ECT inner unchanged, even if the
    arriving outer was ECT(0) or ECT(1). But for safety it should
    drop the Not-ECT inner if the arriving outer was CE. Then, if
    some unfortunate misconfiguration resulted in a congested router
    marking CE on a packet that was originally Not-ECT, drop would be
    the only appropriate signal for the egress to propagate--the only

signal a non-ECN-capable transport (Not-ECT) would understand.

ECT(1) is being proposed as an intermediate level of congestion in a scheme progressing through the IETF [I-D.ietf-pcn-3-in-1-encoding]. But it was decided that it would still be safe to mandate forwarding as Not-ECT for a Not-ECT inner with an ECT(1) outer, thus keeping this combination available for future use. The rationale was as follows: if any misconfiguration led to ECT(1) congestion signals with a Not-ECT inner, it would be safe for the egress to suppress these signals. This is because the congestion would then escalate to CE marking, which the egress would drop, thus avoiding any risk of congestion collapse.

Problems 2 & 3 alone would not warrant a change to decapsulation, but it was decided they are worth fixing and making consistent at the same time as decapsulation code is changed to fix problem 1 (two congestion severity-levels).

6. Backward Compatibility

A tunnel endpoint compliant with the present specification is backward compatible when paired with any tunnel endpoint compliant with any previous tunnelling RFC, whether RFC4301, RFC3168 (see Section 3) or the earlier RFCs summarised in Appendix A (RFC2481, RFC2401 and RFC2003). Each case is enumerated below.

6.1. Non-Issues Updating Decapsulation

At the egress, this specification only augments the per-packet calculation of the ECN field (RFC3168 and RFC4301) for combinations of inner and outer headers that have so far not been used in any IETF protocols.

Therefore, all other things being equal, if an RFC4301 IPsec egress is updated to comply with the new rules, it will still interwork with any RFC4301 compliant ingress and the packet outputs will be identical to those it would have output before (fully backward compatible).

And, all other things being equal, if an RFC3168 egress is updated to comply with the same new rules, it will still interwork with any ingress complying with any previous specification (both modes of RFC3168, both modes of RFC2481, RFC2401 and RFC2003) and the packet outputs will be identical to those it would have output before (fully backward compatible).

A compliant tunnel egress merely needs to implement the one behaviour

in [Section 4](#) with no additional mode or option configuration at the
ingress or egress nor any additional negotiation with the ingress.
The new decapsulation rules have been defined in such a way that
congestion control will still work safely if any of the earlier
versions of ECN processing are used unilaterally at the encapsulating
ingress of the tunnel (any of [RFC2003](#), [RFC2401](#), either mode of
[RFC2481](#), either mode of [RFC3168](#), [RFC4301](#) and this present
specification).

[6.2](#). Non-Update of [RFC4301](#) IPsec Encapsulation

An [RFC4301](#) IPsec ingress can comply with this new specification
without any update and it has no need for any new modes, options or
configuration. So, all other things being equal, it will continue to
interwork identically with any egress it worked with before (fully
backward compatible).

[6.3](#). Update to [RFC3168](#) Encapsulation

The encapsulation behaviour of the new normal mode copies the ECN
field whereas [RFC3168](#) full functionality mode reset it. However, all
other things being equal, if [RFC3168](#) ingress is updated to the
present specification, the outgoing packets from any tunnel egress
will still be unchanged. This is because all variants of tunnelling
at either end ([RFC4301](#), both modes of [RFC3168](#), both modes of [RFC2481](#),
[RFC2401](#), [RFC2003](#) and the present specification) have always
propagated an incoming CE marking through the inner header and onward
into the outgoing header, whether the outer header is reset or
copied. Therefore, If the tunnel is considered as a black box, the
packets output from any egress will be identical with or without an
update to the ingress. Nonetheless, if packets are observed within
the black box (between the tunnel endpoints), CE markings copied by
the updated ingress will be visible within the black box, whereas
they would not have been before. Therefore, the update to
encapsulation can be termed 'black-box backwards compatible' (i.e.
identical unless you look inside the tunnel).

This specification introduces no new backward compatibility issues
when a compliant ingress talks with a legacy egress, but it has to
provide similar safeguards to those already defined in [RFC3168](#).
[RFC3168](#) laid down rules to ensure that an [RFC3168](#) ingress turns off
ECN (limited functionality mode) if it is paired with a legacy egress
([RFC 2481](#), [RFC2401](#) or [RFC2003](#)), which would not propagate ECN
correctly. The present specification carries forward those rules
([Section 4.3](#)). It uses compatibility mode whenever [RFC3168](#) would
have used limited functionality mode, and their per-packet behaviours
are identical. Therefore, all other things being equal, an ingress
using the new rules will interwork with any legacy tunnel egress in

exactly the same way as an RFC3168 ingress (still black-box backward
compatible).

7. Design Principles for Alternate ECN Tunnelling Semantics

This section is informative not normative.

S.5 of RFC3168 permits the Diffserv codepoint (DSCP)[RFC2474] to
'switch in' alternative behaviours for marking the ECN field, just as
it switches in different per-hop behaviours (PHBs) for scheduling.
[RFC4774] gives best current practice for designing such alternative
ECN semantics and very briefly mentions in section 5.4 that
tunnelling should be considered. The guidance below extends RFC4774,
giving additional guidance on designing any alternate ECN semantics
that would also require alternate tunnelling semantics.

The overriding guidance is: "Avoid designing alternate ECN tunnelling
semantics, if at all possible." If a scheme requires tunnels to
implement special processing of the ECN field for certain DSCPs, it
will be hard to guarantee that every implementer of every tunnel will
have added the required exception or that operators will have
ubiquitously deployed the required updates. It is unlikely a single
authority is even aware of all the tunnels in a network, which may
include tunnels set up by applications between endpoints, or
dynamically created in the network. Therefore it is highly likely
that some tunnels within a network or on hosts connected to it will
not implement the required special case.

That said, if a non-default scheme for tunnelling the ECN field is
really required, the following guidelines may prove useful in its
design:

On encapsulation in any new scheme:

   1. The ECN field of the outer header should be cleared to Not-ECT
       ("00") unless it is guaranteed that the corresponding tunnel
       egress will correctly propagate congestion markings introduced
       across the tunnel in the outer header.

   2. If it has established that ECN will be correctly propagated,
       an encapsulator should also copy incoming congestion
       notification into the outer header. The general principle
       here is that the outer header should reflect congestion
       accumulated along the whole upstream path, not just since the
       tunnel ingress (Appendix B.3 on management and monitoring
       explains).

       In some circumstances (e.g. pseudowires, PCN), the whole path

is divided into segments, each with its own congestion
notification and feedback loop. In these cases, the function
that regulates load at the start of each segment will need to
reset congestion notification for its segment. Often the
point where congestion notification is reset will also be
located at the start of a tunnel. However, the resetting
function should be thought of as being applied to packets
after the encapsulation function--two logically separate
functions even though they might run on the same physical box.
Then the code module doing encapsulation can keep to the
copying rule and the load regulator module can reset
congestion, without any code in either module being
conditional on whether the other is there.

On decapsulation in any new scheme:

1. If the arriving inner header is Not-ECT it implies the
    transport will not understand other ECN codepoints. If the
    outer header carries an explicit congestion marking, the
    alternate scheme will probably need to drop the packet--the
    only indication of congestion the transport will understand.
    If the outer carries any other ECN codepoint that does not
    indicate congestion, the alternate scheme can forward the
    packet, but probably only as Not-ECT.

2. If the arriving inner header is other than Not-ECT, the ECN
    field that the alternate decapsulation scheme forwards should
    reflect the more severe congestion marking of the arriving
    inner and outer headers.

3. Any alternate scheme MUST define a behaviour for all
    combinations of inner and outer headers, even those that would
    not be expected to result from standards known at the time or
    from the expected behaviour of the tunnel ingress paired with
    the egress at run-time. Consideration should be given to
    logging such unexpected combinations and raising an alarm,
    particularly if there is a danger that the invalid combination
    implies congestion signals are not being propagated correctly.
    The presence of currently unused combinations may represent an
    attack, but the new scheme should try to define a way to
    forward such packets, at least if a safe outgoing codepoint
    can be defined. Raising an alarm to warn of the possibility
    of an attack is a preferable approach to dropping that ensures
    these combinations can be usable in future standards actions.

IANA Considerations (to be removed on publication):

   This memo includes no request to IANA.

8. Security Considerations

   Appendix B.1 discusses the security constraints imposed on ECN tunnel
   processing. The new rules for ECN tunnel processing (Section 4)
   trade-off between information security (covert channels) and
   congestion monitoring & control. In fact, ensuring congestion
   markings are not lost is itself another aspect of security, because
   if we allowed congestion notification to be lost, any attempt to
   enforce a response to congestion would be much harder.

   Specialist security issues:

   Tunnels intersecting Diffserv regions with alternate ECN semantics:
      If alternate congestion notification semantics are defined for a
      certain Diffserv PHB, the scope of the alternate semantics might
      typically be bounded by the limits of a Diffserv region or
      regions, as envisaged in [RFC4774] (e.g. the pre-congestion
      notification architecture [RFC5559]). The inner headers in
      tunnels crossing the boundary of such a Diffserv region but ending
      within the region can potentially leak the external congestion
      notification semantics into the region, or leak the internal
      semantics out of the region. [RFC2983] discusses the need for
      Diffserv traffic conditioning to be applied at these tunnel
      endpoints as if they are at the edge of the Diffserv region.
      Similar concerns apply to any processing or propagation of the ECN
      field at the edges of a Diffserv region with alternate ECN
      semantics. Such edge processing must also be applied at the
      endpoints of tunnels with one end inside and the other outside the
      domain. [RFC5559] gives specific advice on this for the PCN case,
      but other definitions of alternate semantics will need to discuss
      the specific security implications in each case.

   ECN nonce tunnel coverage: The new decapsulation rules improve the
      coverage of the ECN nonce [RFC3540] relative to the previous rules
      in RFC3168 and RFC4301. However, nonce coverage is still not
      perfect, as this would have led to a safety problem in another
      case. Both are corner-cases, so discussion of the compromise
      between them is deferred to Appendix F.

   Covert channel not turned off: A legacy (RFC3168) tunnel ingress
      could ask an RFC3168 egress to turn off ECN processing as well as
      itself turning off ECN. An egress compliant with the present
      specification will agree to such a request from a legacy ingress,
      but it relies on the ingress solely sending Not-ECT in the outer.

If the egress receives other ECN codepoints in the outer it will
process them as normal, so it will actually still copy congestion
markings from the outer to the outgoing header. Referring for
example to Figure 5 (Appendix B.1), although the tunnel ingress
'I' will set all ECN fields in outer headers to Not-ECT, 'M' could
still toggle CE or ECT(1) on and off to communicate covertly with
'B', because we have specified that 'E' only has one mode
regardless of what mode it says it has negotiated. We could have
specified that 'E' should have a limited functionality mode and
check for such behaviour. But we decided not to add the extra
complexity of two modes on a compliant tunnel egress merely to
cater for an historic security concern that is now considered
manageable.

9. Conclusions

This document uses previously unused combinations of inner and outer
header to augment the rules for calculating the ECN field when
decapsulating IP packets at the egress of IPsec (RFC4301) and non-
IPsec (RFC3168) tunnels. In this way it allows tunnels to propagate
an extra level of congestion severity.

This document also updates the ingress tunnelling encapsulation of
RFC3168 ECN to bring all IP in IP tunnels into line with the new
behaviour in the IPsec architecture of RFC4301, which copies rather
than resets the ECN field when creating outer headers.

The need for both these updated behaviours was triggered by the
introduction of pre-congestion notification (PCN) onto the IETF
standards track. Operators wanting to support PCN or other alternate
ECN schemes that use an extra severity level can require that their
tunnels comply with the present specification. Nonetheless, as part
of general code maintenance, any tunnel can safely be updated to
comply with this specification, because it is backward compatible
with all previous tunnelling behaviours which will continue to work
as before--just using one severity level.

The new rules propagate changes to the ECN field across tunnel end-
points that previously blocked them to restrict the bandwidth of a
potential covert channel. Limiting the channel's bandwidth to 2 bits
per packet is now considered sufficient.

At the same time as removing these legacy constraints, the
opportunity has been taken to draw together diverging tunnel
specifications into a single consistent behaviour. Then any tunnel
can be deployed unilaterally, and it will support the full range of
congestion control and management schemes without any modes or
configuration. Further, any host or router can expect the ECN field

to behave in the same way, whatever type of tunnel might intervene in
the path. This new certainty could enable new uses of the ECN field
that would otherwise be confounded by ambiguity.

10. Acknowledgements

Thanks to Anil Agawaal for pointing out a case where it's safe for a
tunnel decapsulator to forward a combination of headers it does not
understand. Thanks to David Black for explaining a better way to
think about function placement. Also thanks to Arnaud Jacquet for
the idea for Appendix C. Thanks to Michael Menth, Bruce Davie, Toby
Moncaster, Gorry Fairhurst, Sally Floyd, Alfred Hoenes, Gabriele
Corliano, Ingemar Johansson, David Black and Phil Eardley for their
thoughts and careful review comments.

Comments Solicited (to be removed by the RFC Editor):

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.

11. References

11.1. Normative References

[RFC2003]   Perkins, C., "IP Encapsulation
                                 within IP", RFC 2003, October 1996.

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

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

[RFC4301]   Kent, S. and K. Seo, "Security
                                 Architecture for the Internet
                                 Protocol", RFC 4301, December 2005.

11.2. Informative References

   [I-D.ietf-pcn-3-in-1-encoding]   Briscoe, B. and T. Moncaster, "PCN
                                    3-State Encoding Extension in a
                                    single DSCP",
                                    draft-ietf-pcn-3-in-1-encoding-00
                                    (work in progress), July 2009.

   [I-D.ietf-pcn-3-state-encoding] Moncaster, T., Briscoe, B., and M.
                                    Menth, "A PCN encoding using 2
                                    DSCPs to provide 3 or more states",
                                    draft-ietf-pcn-3-state-encoding-00
                                    (work in progress), April 2009.

   [I-D.ietf-pcn-psdm-encoding]   Menth, M., Babiarz, J., Moncaster,
                                    T., and B. Briscoe, "PCN Encoding
                                    for Packet-Specific Dual Marking
                                    (PSDM)",
                                    draft-ietf-pcn-psdm-encoding-00
                                    (work in progress), June 2009.

   [I-D.ietf-pcn-sm-edge-behaviour] Charny, A., Karagiannis, G., Menth,
                                    M., and T. Taylor, "PCN Boundary
                                    Node Behaviour for the Single
                                    Marking (SM) Mode of Operation",
                                    draft-ietf-pcn-sm-edge-behaviour-01
                                    (work in progress), October 2009.

   [I-D.satoh-pcn-st-marking]   Satoh, D., Ueno, H., Maeda, Y., and
                                    O. Phanachet, "Single PCN Threshold
                                    Marking by using PCN baseline
                                    encoding for both admission and
                                    termination controls",
                                    draft-satoh-pcn-st-marking-02 (work
                                    in progress), September 2009.

   [RFC2401]   Kent, S. and R. Atkinson, "Security
                                    Architecture for the Internet
                                    Protocol", RFC 2401, November 1998.

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

   [RFC2481]   Ramakrishnan, K. and S. Floyd, "A
                                    Proposal to add Explicit Congestion

                                   Notification (ECN) to IP",
                                   RFC 2481, January 1999.

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

   [RFC3540]   Spring, N., Wetherall, D., and D.
                                   Ely, "Robust Explicit Congestion
                                   Notification (ECN) Signaling with
                                   Nonces", RFC 3540, June 2003.

   [RFC4306]   Kaufman, C., "Internet Key Exchange
                                   (IKEv2) Protocol", RFC 4306,
                                   December 2005.

   [RFC4774]   Floyd, S., "Specifying Alternate
                                   Semantics for the Explicit
                                   Congestion Notification (ECN)
                                   Field", BCP 124, RFC 4774,
                                   November 2006.

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

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

   [RFC5670]   Eardley, P., "Metering and Marking
                                   Behaviour of PCN-Nodes", RFC 5670,
                                   November 2009.

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

   [VCP]   Xia, Y., Subramanian, L., Stoica,
                                   I., and S. Kalyanaraman, "One more
                                   bit is enough", Proc. SIGCOMM'05,
                                   ACM CCR 35(4)37--48, 2005, <http://
                                   doi.acm.org/10.1145/
                                   1080091.1080098>.

Appendix A. Early ECN Tunnelling RFCs

   IP in IP tunnelling was originally defined in [RFC2003]. On
   encapsulation, the incoming header was copied to the outer and on
   decapsulation the outer was simply discarded. Initially, IPsec
   tunnelling [RFC2401] followed the same behaviour.

   When ECN was introduced experimentally in [RFC2481], legacy (RFC2003
   or RFC2401) tunnels would have discarded any congestion markings
   added to the outer header, so RFC2481 introduced rules for
   calculating the outgoing header from a combination of the inner and
   outer on decapsulation. RC2481 also introduced a second mode for
   IPsec tunnels, which turned off ECN processing(Not-ECT) in the outer
   header on encapsulation because an RFC2401 decapsulator would discard
   the outer on decapsulation. For RFC2401 IPsec this had the side-
   effect of completely blocking the covert channel.

   In RFC2481 the ECN field was defined as two separate bits. But when
   ECN moved from the experimental to the standards track [RFC3168], the
   ECN field was redefined as four codepoints. This required a
   different calculation of the ECN field from that used in RFC2481 on
   decapsulation. RFC3168 also had two modes; a 'full functionality
   mode' that restricted the covert channel as much as possible but
   still allowed ECN to be used with IPsec, and another that completely
   turned off ECN processing across the tunnel. This 'limited
   functionality mode' both offered a way for operators to completely
   block the covert channel and allowed an RFC3168 ingress to interwork
   with a legacy tunnel egress (RFC2481, RFC2401 or RFC2003).

   The present specification includes a similar compatibility mode to
   interwork safely with tunnels compliant with any of these three
   earlier RFCs. However, unlike RFC3168, it is only a mode of the
   ingress, as decapsulation behaviour is the same in either case.

Appendix B. Design Constraints

   Tunnel processing of a congestion notification field has to meet
   congestion control and management needs without creating new
   information security vulnerabilities (if information security is
   required). This appendix documents the analysis of the tradeoffs
   between these factors that led to the new encapsulation rules in
   Section 4.1.

B.1. Security Constraints

   Information security can be assured by using various end to end
   security solutions (including IPsec in transport mode [RFC4301]), but
   a commonly used scenario involves the need to communicate between two

physically protected domains across the public Internet. In this
case there are certain management advantages to using IPsec in tunnel
mode solely across the publicly accessible part of the path. The
path followed by a packet then crosses security 'domains'; the ones
protected by physical or other means before and after the tunnel and
the one protected by an IPsec tunnel across the otherwise unprotected
domain. We will use the scenario in Figure 5 where endpoints 'A' and
'B' communicate through a tunnel. The tunnel ingress 'I' and egress
'E' are within physically protected edge domains, while the tunnel
spans an unprotected internetwork where there may be 'men in the
middle', M.

```
            physically        unprotected      physically
          <-protected domain-><--domain--><-protected domain->
          +------------------+           +------------------+
          |                  |     M     |                  |
          |      A-------->I=========>==========>E-------->B   |
          |                  |           |                  |
          +------------------+           +------------------+
                      <----IPsec secured---->
                             tunnel
```
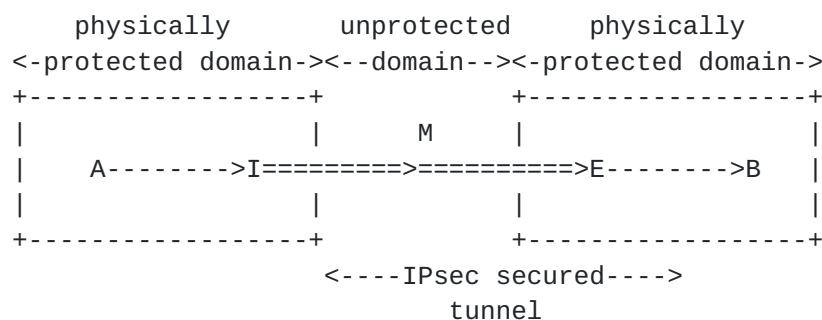
Figure 5: IPsec Tunnel Scenario

IPsec encryption is typically used to prevent 'M' seeing messages
from 'A' to 'B'. IPsec authentication is used to prevent 'M'
masquerading as the sender of messages from 'A' to 'B' or altering
their contents. But 'I' can also use IPsec tunnel mode to allow 'A'
to communicate with 'B', but impose encryption to prevent 'A' leaking
information to 'M'. Or 'E' can insist that 'I' uses tunnel mode
authentication to prevent 'M' communicating information to 'B'.
Mutable IP header fields such as the ECN field (as well as the TTL/
Hop Limit and DS fields) cannot be included in the cryptographic
calculations of IPsec. Therefore, if 'I' copies these mutable fields
into the outer header that is exposed across the tunnel it will have
allowed a covert channel from 'A' to M that bypasses its encryption
of the inner header. And if 'E' copies these fields from the outer
header to the inner, even if it validates authentication from 'I', it
will have allowed a covert channel from 'M' to 'B'.

ECN at the IP layer is designed to carry information about congestion
from a congested resource towards downstream nodes. Typically a
downstream transport might feed the information back somehow to the
point upstream of the congestion that can regulate the load on the
congested resource, but other actions are possible (see [RFC3168]
S.6). In terms of the above unicast scenario, ECN effectively
intends to create an information channel (for congestion signalling)
from 'M' to 'B' (for 'B' to feed back to 'A'). Therefore the goals

of IPsec and ECN are mutually incompatible, requiring some
compromise.

With respect to the DS or ECN fields, S.5.1.2 of RFC4301 says,
"controls are provided to manage the bandwidth of this [covert]
channel". Using the ECN processing rules of RFC4301, the channel
bandwidth is two bits per datagram from 'A' to 'M' and one bit per
datagram from 'M' to 'A' (because 'E' limits the combinations of the
2-bit ECN field that it will copy). In both cases the covert channel
bandwidth is further reduced by noise from any real congestion
marking. RFC4301 implies that these covert channels are sufficiently
limited to be considered a manageable threat. However, with respect
to the larger (6b) DS field, the same section of RFC4301 says not
copying is the default, but a configuration option can allow copying
"to allow a local administrator to decide whether the covert channel
provided by copying these bits outweighs the benefits of copying".
Of course, an administrator considering copying of the DS field has
to take into account that it could be concatenated with the ECN field
giving an 8b per datagram covert channel.

For tunnelling the 6b Diffserv field two conceptual models have had
to be defined so that administrators can trade off security against
the needs of traffic conditioning [RFC2983]:

The uniform model: where the Diffserv field is preserved end-to-end
   by copying into the outer header on encapsulation and copying from
   the outer header on decapsulation.

The pipe model: where the outer header is independent of that in the
   inner header so it hides the Diffserv field of the inner header
   from any interaction with nodes along the tunnel.

However, for ECN, the new IPsec security architecture in RFC4301 only
standardised one tunnelling model equivalent to the uniform model.
It deemed that simplicity was more important than allowing
administrators the option of a tiny increment in security, especially
given not copying congestion indications could seriously harm
everyone's network service.

B.2. Control Constraints

Congestion control requires that any congestion notification marked
into packets by a resource will be able to traverse a feedback loop
back to a function capable of controlling the load on that resource.
To be precise, rather than calling this function the data source, we
will call it the Load Regulator. This will allow us to deal with
exceptional cases where load is not regulated by the data source, but
usually the two terms will be synonymous. Note the term "a function

capable of controlling the load" deliberately includes a source
application that doesn't actually control the load but ought to (e.g.
an application without congestion control that uses UDP).


```
        A--->R--->I=========>M=========>E-------->B
```


                    Figure 6: Simple Tunnel Scenario

We now consider a similar tunnelling scenario to the IPsec one just
described, but without the different security domains so we can just
focus on ensuring the control loop and management monitoring can work
(Figure 6). If we want resources in the tunnel to be able to
explicitly notify congestion and the feedback path is from 'B' to
'A', it will certainly be necessary for 'E' to copy any CE marking
from the outer header to the inner header for onward transmission to
'B', otherwise congestion notification from resources like 'M' cannot
be fed back to the Load Regulator ('A'). But it does not seem
necessary for 'I' to copy CE markings from the inner to the outer
header. For instance, if resource 'R' is congested, it can send
congestion information to 'B' using the congestion field in the inner
header without 'I' copying the congestion field into the outer header
and 'E' copying it back to the inner header. 'E' can still write any
additional congestion marking introduced across the tunnel into the
congestion field of the inner header.

It might be useful for the tunnel egress to be able to tell whether
congestion occurred across a tunnel or upstream of it. If outer
header congestion marking was reset by the tunnel ingress ('I'), at
the end of a tunnel ('E') the outer headers would indicate congestion
experienced across the tunnel ('I' to 'E'), while the inner header
would indicate congestion upstream of 'I'. But similar information
can be gleaned even if the tunnel ingress copies the inner to the
outer headers. At the end of the tunnel ('E'), any packet with an
extra mark in the outer header relative to the inner header
indicates congestion across the tunnel ('I' to 'E'), while the inner
header would still indicate congestion upstream of ('I'). Appendix C
gives a simple and precise method for a tunnel egress to infer the
congestion level introduced across a tunnel.

All this shows that 'E' can preserve the control loop irrespective of
whether 'I' copies congestion notification into the outer header or
resets it.

That is the situation for existing control arrangements but, because
copying reveals more information, it would open up possibilities for
better control system designs. For instance, Appendix E describes

how resetting CE marking on encapsulation breaks a proposed
congestion marking scheme on the standards track. It ends up
removing excessive amounts of traffic unnecessarily. Whereas copying
CE markings at ingress leads to the correct control behaviour.

B.3. Management Constraints

As well as control, there are also management constraints.
Specifically, a management system may monitor congestion markings in
passing packets, perhaps at the border between networks as part of a
service level agreement. For instance, monitors at the borders of
autonomous systems may need to measure how much congestion has
accumulated so far along the path, perhaps to determine between them
how much of the congestion is contributed by each domain.

In this document we define the baseline of congestion marking (or the
Congestion Baseline) as the source of the layer that created (or most
recently reset) the congestion notification field. When monitoring
congestion it would be desirable if the Congestion Baseline did not
depend on whether packets were tunnelled or not. Given some tunnels
cross domain borders (e.g. consider M in Figure 6 is monitoring a
border), it would therefore be desirable for 'I' to copy congestion
accumulated so far into the outer headers, so that it is exposed
across the tunnel.

Appendix C. Contribution to Congestion across a Tunnel

This specification mandates that a tunnel ingress determines the ECN
field of each new outer tunnel header by copying the arriving header.
Concern has been expressed that this will make it difficult for the
tunnel egress to monitor congestion introduced only along a tunnel,
which is easy if the outer ECN field is reset at a tunnel ingress
(RFC3168 full functionality mode). However, in fact copying CE marks
at ingress will still make it easy for the egress to measure
congestion introduced across a tunnel, as illustrated below.

Consider 100 packets measured at the egress. Say it measures that 30
are CE marked in the inner and outer headers and 12 have additional
CE marks in the outer but not the inner. This means packets arriving
at the ingress had already experienced 30% congestion. However, it
does not mean there was 12% congestion across the tunnel. The
correct calculation of congestion across the tunnel is $p_t = 12/
(100-30) = 12/70 = 17\%$. This is easy for the egress to measure. It
is simply the packets with additional CE marking in the outer header
(12) as a proportion of packets not marked in the inner header (70).

Figure 7 illustrates this in a combinatorial probability diagram.
The square represents 100 packets. The 30% division along the bottom

represents marking before the ingress, and the p_t division up the
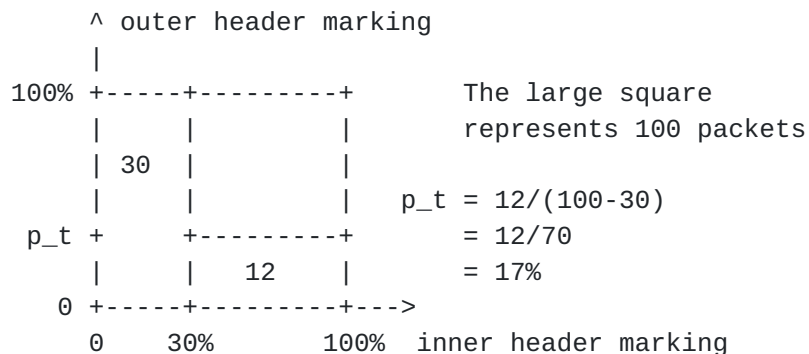side represents marking introduced across the tunnel.

```
     ^ outer header marking
     |
100% +-----+---------+        The large square
     |     |         |        represents 100 packets
     | 30  |         |
     |     |         |   p_t = 12/(100-30)
 p_t +     +---------+       = 12/70
     |     |   12    |       = 17%
   0 +-----+---------+--->
     0    30%       100%  inner header marking
```

          Figure 7: Tunnel Marking of Packets Already Marked at Ingress

Appendix D. Why Losing ECT(1) on Decapsulation Impedes PCN

Congestion notification with two severity levels is currently on the
IETF's standards track agenda in the Congestion and Pre-Congestion
Notification (PCN) working group. PCN needs all four possible states
of congestion signalling in the 2-bit ECN field to be propagated at
the egress, but pre-existing tunnels only propagate three. The four
PCN states are: not PCN-enabled, not marked and two increasingly
severe levels of congestion marking. The less severe marking means
'stop admitting new traffic' and the more severe marking means
'terminate some existing flows', which may be needed after reroutes
(see [RFC5559] for more details). (Note on terminology: wherever
this document counts four congestion states, the PCN working group
would count this as three PCN states plus a not-PCN-enabled state.)

Figure 2 (Section 3.2) shows that pre-existing decapsulation
behaviour would have discarded any ECT(1) markings in outer headers
if the inner was ECT(0). This prevented the PCN working group from
using ECT(1) -- if a PCN node used ECT(1) to indicate one of the
severity levels of congestion, any later tunnel egress would revert
the marking to ECT(0) as if nothing had happened. Effectively the
decapsulation rules of RFC4301 and RFC3168 waste one ECT codepoint;
they treat the ECT(0) and ECT(1) codepoints as a single codepoint.

A number of work-rounds to this problem were proposed in the PCN w-g;
to add the fourth state another way or avoid needing it. Without
wishing to disparage the ingenuity of these work-rounds, none were
chosen for the standards track because they were either somewhat
wasteful, imprecise or complicated:

o One uses a pair of Diffserv codepoint(s) in place of each PCN DSCP
   to encode the extra state [I-D.ietf-pcn-3-state-encoding], using

        up the rapidly exhausting DSCP space while leaving an ECN
        codepoint unused.

    o Another survives tunnelling without an extra DSCP
        [I-D.ietf-pcn-psdm-encoding], but it requires the PCN edge
        gateways to share the initial state of a packet out of band.

    o Another proposes a more involved marking algorithm in forwarding
        elements to encode the three congestion notification states using
        only two ECN codepoints [I-D.satoh-pcn-st-marking].

    o Another takes a different approach; it compromises the precision
        of the admission control mechanism in some network scenarios, but
        manages to work with just three encoding states and a single
        marking algorithm [I-D.ietf-pcn-sm-edge-behaviour].

    Rather than require the IETF to bless any of these experimental
    encoding work-rounds, the present specification fixes the root cause
    of the problem so that operators deploying PCN can simply require
    that tunnel end-points within a PCN region should comply with this
    new ECN tunnelling specification. On the public Internet it would
    not be possible to know whether all tunnels complied with this new
    specification, but universal compliance is feasible for PCN, because
    it is intended to be deployed in a controlled Diffserv region.

    Given the present specification, the PCN w-g could progress a
    trivially simple four-state ECN encoding
    [I-D.ietf-pcn-3-in-1-encoding]. This would replace the interim
    standards track baseline encoding of just three states [RFC5696]
    which makes a fourth state available for any of the experimental
    alternatives.

Appendix E. Why Resetting ECN on Encapsulation Impedes PCN

    The PCN architecture says "...if encapsulation is done within the
    PCN-domain: Any PCN-marking is copied into the outer header. Note: A
    tunnel will not provide this behaviour if it complies with [RFC3168]
    tunnelling in either mode, but it will if it complies with [RFC4301]
    IPsec tunnelling. "

    The specific issue here concerns PCN excess rate marking [RFC5670].
    The purpose of excess rate marking is to provide a bulk mechanism for
    interior nodes within a PCN domain to mark traffic that is exceeding
    a configured threshold bit-rate, perhaps after an unexpected event
    such as a reroute, a link or node failure, or a more widespread
    disaster. Reroutes are a common cause of QoS degradation in IP
    networks. After reroutes it is common for multiple links in a
    network to become stressed at once. Therefore, PCN excess rate

marking has been carefully designed to ensure traffic marked at one
queue will not be counted again for marking at subsequent queues (see
the `Excess traffic meter function' of [RFC5670]).

However, if an RFC3168 tunnel ingress intervenes, it resets the ECN
field in all the outer headers. This will cause excess traffic to be
counted more than once, leading to many flows being removed that did
not need to be removed at all. This is why the an RFC3168 tunnel
ingress cannot be used in a PCN domain.

The ECN reset in RFC3168 is no longer deemed necessary, it is
inconsistent with RFC4301, it is not as simple as RFC4301 and it is
impeding deployment of new protocols like PCN. The present
specification corrects this perverse situation.

Appendix F. Compromise on Decap with ECT(1) Inner and ECT(0) Outer

A packet with an ECT(1) inner and an ECT(0) outer should never arise
from any known IETF protocol. Without giving a reason, RFC3168 and
RFC4301 both say the outer should be ignored when decapsulating such
a packet. This appendix explains why it was decided not to change
this advice.

In summary, ECT(0) always means 'not congested' and ECT(1) may imply
the same [RFC3168] or it may imply a higher severity congestion
signal [RFC4774], [I-D.ietf-pcn-3-in-1-encoding], depending on the
transport in use. Whether they mean the same or not, at the ingress
the outer should have started the same as the inner and only a broken
or compromised router could have changed the outer to ECT(0).

The decapsulator can detect this anomaly. But the question is,
should it correct the anomaly by ignoring the outer, or should it
reveal the anomaly to the end-to-end transport by forwarding the
outer?

On balance, it was decided that the decapsulator should correct the
anomaly, but log the event and optionally raise an alarm. This is
the safe action if ECT(1) is being used as a more severe marking than
ECT(0), because it passes the more severe signal to the transport.
However, it is not a good idea to hide anomalies, which is why an
optional alarm is suggested. It should be noted that this anomaly
may be the result of two changes to the outer: a broken or
compromised router within the tunnel might be erasing congestion
markings introduced earlier in the same tunnel by a congested router.
In this case, the anomaly would be losing congestion signals, which
needs immediate attention.

The original reason for defining ECT(0) and ECT(1) as equivalent was

so that the data source could use the ECN nonce [RFC3540] to detect
if congestion signals were being erased. However, in this case, the
decapsulator does not need a nonce to detect any anomalies introduced
within the tunnel, because it has the inner as a record of the header
at the ingress. Therefore, it was decided that the best compromise
would be to give precedence to solving the safety issue over
revealing the anomaly, because the anomaly could at least be detected
and dealt with internally.

Superficially, the opposite case where the inner and outer carry
different ECT values, but with an ECT(1) outer and ECT(0) inner seems
to require a similar compromise. However, because that case is
reversed, no compromise is necessary; it is best to forward the outer
whether the transport expects the ECT(1) to mean a higher severity
than ECT(0) or the same severity. Forwarding the outer either
preserves a higher value (if it is higher) or it reveals an anomaly
to the transport (if the two ECT codepoints mean the same severity).

Appendix G. Open Issues

The new decapsulation behaviour defined in Section 4.2 adds support
for propagation of 2 severity levels of congestion. However
transports have no way to discover whether there are any legacy
tunnels on their path that will not propagate 2 severity levels. It
would have been nice to add a feature for transports to check path
support, but this remains an open issue that will have to be
addressed in any future standards action to define an end-to-end
scheme that requires 2-severity levels of congestion. PCN avoids
this problem, because it is only for a controlled region, so all
legacy tunnels can be upgraded by the same operator that deploys PCN.

Author's Address

   Bob Briscoe
   BT
   B54/77, Adastral Park
   Martlesham Heath
   Ipswich IP5 3RE
   UK

   Phone: +44 1473 645196
   EMail: bob.briscoe@bt.com
   URI: http://bobbriscoe.net/