Transport Area Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 25, 2009

B. Briscoe
BT

March 24, 2009

# Tunnelling of Explicit Congestion Notification
# draft-ietf-tsvwg-ecn-tunnel-02

## Status of this Memo



## Copyright Notice



## 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 brings all IP in

IP tunnels (v4 or v6) into line with the way RFC4301 IPsec tunnels now construct the ECN field. On decapsulation it redefines how the ECN field in the forwarded IP header should be calculated for two previously invalid combinations of incoming inner and outer headers, in order that these combinations may be usefully employed in future standards actions. It includes a thorough analysis of the reasoning for these changes and the implications.

Table of Contents

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.cs.ucl.ac.uk/staff/B.Briscoe/pubs.html#ecn-tunnel>

From ietf-01 to ietf-02 (current):

* 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 A);

  + Added Section 3 to summarise relevant existing RFCs;

  + Structured Section 4 and Section 5 into subsections.

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

  + Moved Section 4.3 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.

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 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 -00 to -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 5);

* 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.2, 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

This document redefines how the explicit congestion notification (ECN) field [RFC3168] in the IP header should be constructed for all IP in IP tunnelling. Previously, tunnel endpoints blocked visibility of transitions of the ECN field except the minimum necessary to allow the basic ECN mechanism to work. Three main change are defined, one on entry to and two on exit from any IP in IP tunnel. The newly specified behaviours make all transitions to the ECN field visible across tunnel end-points, so tunnels no longer restrict new uses of the ECN field that were not envisaged when ECN was first designed.

The immediate motivation for opening up the ECN behaviour of tunnels is because otherwise they impede the introduction of pre-congestion notification (PCN [I-D.ietf-pcn-marking-behaviour]) in networks with tunnels (Appendix E explains why). But these changes are not just intended to ease the introduction of PCN; care has been taken to ensure the resulting ECN tunnelling behaviour is simple and generic for other potential future uses.

Given this is a change to behaviour at 'the neck of the hourglass',

an extensive analysis of the trade-offs between control, management and security constraints has been conducted in order to minimise unexpected side-effects both now and in the future. Care has also been taken to ensure the changes are fully backwards compatible with all previous tunnelling behaviours.

The ECN protocol allows a forwarding element to notify the onset of congestion of its resources without having to drop packets. Instead it can explicitly mark a proportion of packets by setting the congestion experienced (CE) codepoint in the 2-bit ECN field in the IP header (see Table 1 for a recap of the ECN codepoints).

| 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

The outer header of an IP packet can encapsulate one (or more) additional IP headers tunnelled within it. A forwarding element that is using ECN to signify congestion will only mark the outer IP header that is immediately visible to it. When a tunnel decapsulator later removes this outer header, it must follow rules to ensure the marking is propagated into the IP header being forwarded onwards, otherwise congestion notifications will disappear into a black hole leading to potential congestion collapse.

The rules for constructing the ECN field to be forwarded after tunnel decapsulation ensure this happens, but they are not wholly straightforward, and neither are the rules for encapsulating one IP header in another on entry to a tunnel. The factor that has introduced most complication at both ends of a tunnel has been the possibility that the ECN field might be used as a covert channel to compromise the integrity of an IPsec tunnel.

A common use for IPsec is to create a secure tunnel between two secure sites across the public Internet. A field like ECN that can change as it traverses the Internet cannot be covered by IPsec's integrity mechanisms. Therefore, the ECN field might be toggled (with two bits per packet) to communicate between a secure site and someone on the public Internet--a covert channel.

Over the years covert channel restrictions have been added to the design of ECN (with consequent backward compatibility complications). However the latest IPsec architecture [RFC4301] takes the view that simplicity is more important than closing off the covert channel threat, which it deems manageable given its bandwidth is limited to two bits per packet.

As a result, an unfortunate sequence of standards actions has left us with nearly the worst of all possible combinations of outcomes, despite the best endeavours of everyone concerned. The new IPsec architecture [RFC4301] only updates the earlier specification of ECN tunnelling behaviour [RFC3168] for the case of IPsec tunnels. For the case of non-IPsec tunnels the earlier RFC3168 specification still applies. At the time RFC3168 was standardised, covert channels through the ECN field were restricted, whether or not IPsec was being used. The perverse position now is that non-IPsec tunnels restrict covert channels, while IPsec tunnels don't.

Actually, this statement needs some qualification. IPsec tunnels only don't restrict the ECN covert channel at the ingress. At the tunnel egress, the presumption that the ECN covert channel should be restricted has not been removed from any tunnelling specifications, whether IPsec or not.

Now that these historic 2-bit covert channel constraints are impeding the introduction of PCN, this specification is designed to remove them and at the same time streamline the whole ECN behaviour for the future.

## 1.1. Scope

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

This document specifies common, default ECN field processing at encapsulation and decapsulation for any IP in IP tunnelling. It applies irrespective of whether IPv4 or IPv6 is used for either of the inner and outer headers. It applies to 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.

Nonetheless, if necessary, 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 4.3 should be followed as closely

as possible. There is no requirement for a PHB to state anything about ECN tunnelling behaviour if the new default behaviour is sufficient.

[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.

### 1.2. Document Roadmap

The body of the document focuses solely on standards actions impacting implementation. Appendices record the analysis that motivates and justifies these actions. The whole document is organised as follows:

- Section 3 recaps relevant existing RFCs and explains exactly why changes are needed, referring to Appendix D and Appendix E in order to explain in detail why current tunnelling behaviours impede PCN deployment, at egress and ingress respectively.

- Section 4 uses precise standards terminology to specify the new ECN tunnelling behaviours. It refers to Appendix A for analysis of the trade-offs between security, control and management design constraints that led to these particular standards actions.

- Extending the new IPsec tunnel ingress behaviour to all IP in IP tunnels requires consideration of backwards compatibility, which is covered in Section 5 and detailed changes from earlier RFCs are brought together in Section 6.

- Finally, a number of security considerations are discussed and conclusions are drawn.

- Additional specialist issues are deferred to appendices in addition to those already referred to above, in particular Appendix B discusses specialist tunnelling issues that could arise when ECN is fed back to a load regulation function on a middlebox, rather than at the source of the path.

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 3. Summary of Pre-Existing RFCs

This section is informative not normative. It merely recaps pre-existing RFCs to help motivate changing these behaviours. Earlier relevant RFCs that were either experimental or incomplete with respect to ECN tunnelling (RFC2481, RFC2401 and RFC2003) are not discussed, although the backwards compatibility considerations in Section 5 take them into account. The question of whether tunnel implementations used in the Internet comply with any of these RFCs is also not discussed.

### 3.1. Encapsulation at Tunnel Ingress

The controversy at tunnel ingress has been over whether to propagate information about congestion experienced on the path upstream of the tunnel ingress 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 tunnel ingress must not copy a CE marking from the inner header into the outer header that it creates. Instead the tunnel ingress must set the outer header to ECT(0) (i.e. codepoint 10) if the ECN field is marked CE (codepoint 11) 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 tunnel ingress must simply copy the ECN field from the arriving to the outer header. The main purpose of the present specification is to carry the new behaviour of IPsec over to all IP in IP tunnels, so all tunnel ingress nodes consistently copy the ECN field.

RFC3168 also provided a Limited Functionality mode that turns off ECN processing over the scope of the tunnel. This is necessary if the ingress does not know whether the tunnel egress supports propagation of ECN markings. Neither Limited Functionality mode nor Full Functionality mode are used in RFC4301 IPsec.

These pre-existing behaviours are summarised in Figure 1.

| Incoming Header (also equal to Outgoing Inner Header) | Outgoing Outer Header | | |
|---|---|---|---|
| | RFC3168 ECN Limited Functionality | RFC3168 ECN Full Functionality | RFC4301 IPsec |
| 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

For encapsulation, the specification in Section 4 below brings all IP in IP tunnels (v4 or v6) into line with the way IPsec tunnels [RFC4301] now construct the ECN field, except where a legacy tunnel egress might not understand ECN at all. This removes the now redundant full functionality mode in the middle column of Figure 1. Wherever possible it ensures that the outer header reveals any congestion experienced so far on the whole path, not just since the last tunnel ingress.

Why does it matter if we have different ECN encapsulation behaviours for IPsec and non-IPsec tunnels? A general answer is that gratuitous inconsistency constrains the available design space and makes it harder to design networks and new protocols that work predictably.

But there is also a specific need not to reset the CE codepoint. The standards track proposal for excess rate pre-congestion notification (PCN [I-D.ietf-pcn-marking-behaviour]) only works correctly in the presence of RFC4301 IPsec encapsulation or [RFC5129] MPLS encapsulation, but not with RFC3168 IP in IP encapsulation (Appendix E explains why). The PCN architecture [I-D.ietf-pcn-architecture] states that the regular RFC3168 rules for IP in IP tunnelling of the ECN field should not be used for PCN. But if non-IPsec tunnels are already present within a network to which PCN is being added, that is not particularly helpful advice.

The present specification provides a clean solution to this problem, so that network operators who want to use PCN and tunnels can specify that all tunnel endpoints in a PCN region need to be upgraded to comply with this specification. Also, whether using PCN or not, as more tunnel endpoints comply with this specification, it should make ECN behaviour simpler, faster and more predictable.

To ensure copying rather than resetting CE on ingress will not cause unintended side-effects, Appendix A assesses whether either harm any security, control or management functions. It finds that resetting CE makes life difficult in a number of directions, while copying CE harms nothing (other than opening a low bit-rate covert channel vulnerability which the IETF Security Area now deems is manageable).

## 3.2. Decapsulation at Tunnel Egress

Both 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 Inner Header | Incoming Outer Header | | | |
|---|---|---|---|---|
| | Not-ECT | ECT(0) | ECT(1) | CE |
| Not-ECT | Not-ECT | drop(!!!) | drop(!!!) | drop(!!!) |
| 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, briefly recapped as follows:

- On decapsulation, if the inner ECN field is Not-ECT but the outer ECN field is anything except Not-ECT the decapsulator must drop the packet. Drop is mandated because known legal protocol transitions should not be able to lead to these cases (indicated in the table by '(!!!)'), therefore the decapsulator may also raise an alarm;

- In all other cases, 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, Not-ECT;

- ECT(0) and ECT(1) are considered of equal severity (indicated by just 'ECT' in the rank order above). Where the inner and outer ECN fields are both ECT but they differ, the packet is forwarded with the codepoint of the inner ECN field, which prevents ECT codepoints being used for a covert channel.

The specification for decapsulation in Section 4 fixes two problems with this pre-existing behaviour:

- Firstly, forwarding the codepoint of the inner header in the cases where both inner and outer are different values of ECT effectively implies that any distinction between ECT(0) and ECT(1) cannot be introduced in the future wherever a tunnel might be deployed. Therefore, the currently specified tunnel decapsulation behaviour unnecessarily wastes one of four codepoints (effectively wasting half a bit) in the IP (v4 & v6) header. As explained in Appendix A.1, the original reason for not using the outer ECT codepoints for onward forwarding 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 a general future-proofing issue, this problem is immediately pressing for standardisation of pre-congestion notification (PCN). PCN solutions generally require three encoding states in addition to Not-ECT: one for 'not marked' and two increasingly severe levels of marking. Although the ECN field gives sufficient codepoints for these three states, they cannot all be used for PCN because a change between ECT(0) and ECT(1) in any tunnelled packet would be lost when the outer header was decapsulated, dangerously discarding congestion signalling. A number of wasteful or convoluted work-rounds to this problem are being considered for standardisation by the PCN working group (see Appendix D), but by far the simplest approach is just to remove the covert channel blockages from tunnelling behaviour, that are now deemed unnecessary anyway. Not only will this streamline PCN standardisation, but it could also streamline other future uses of these codepoints.

- Secondly, mandating drop is not always a good idea just because a combination of headers seems invalid. There are many cases where it has become nearly impossible to deploy new standards because legacy middleboxes drop packets carrying header values they don't expect. Where possible, the new decapsulation behaviour specified in Section 4 below is more liberal in its response to unexpected combinations of headers.

## 4. New ECN Tunnelling Rules

The ECN tunnel processing rules below in Section 4.1 (ingress encapsulation) and Section 4.2 (egress decapsulation) are the default for a packet with any DSCP. If required, different ECN encapsulation

rules MAY be defined as part of the definition of an appropriate Diffserv PHB using the guidelines that follow in Section 4.3. However, the deployment burden of handling exceptional PHBs in implementations of all affected tunnels and lower layer link protocols should not be underestimated.

## 4.1. Default Tunnel Ingress Behaviour

A tunnel ingress compliant with this specification MUST implement a `normal mode'. It might also need to implement a `compatibility mode' for backward compatibility with legacy tunnel egresses that do not understand ECN (see Section 5 for when compatibility mode is required). Note that these are modes of the ingress tunnel endpoint only, not the tunnel as a whole.

Whatever the mode, the tunnel ingress forwards the inner header without changing the ECN field. In normal mode a tunnel ingress compliant with this specification MUST construct the outer encapsulating IP header by copying the 2-bit ECN field of the arriving 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 (also equal to Outgoing Inner Header) | Outgoing Outer Header: Compatibility Mode | Outgoing Outer Header: Normal 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

Compatibility mode is the same per packet behaviour as the ingress end of RFC3168's limited functionality mode. Normal mode is the same per packet behaviour as the ingress end of RFC4301 IPsec.

## 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.

| Incoming Inner Header | Incoming Outer Header | | | |
|---|---|---|---|---|
| | Not-ECT | ECT(0) | ECT(1) | CE |
| Not-ECT | Not-ECT | Not-ECT(!!!) | drop(!!!) | 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 | | | |

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 in the outer header onwards. This is because the inner Not-ECT marking is set by transports that would not understand the ECN protocol. Instead:

  * If the inner ECN field is Not-ECT and the outer ECN field is ECT(1) or CE the decapsulator MUST drop the packet.
    Reasoning: these combinations of codepoints either imply some illegal protocol transition has occurred within the tunnel, or that some locally defined mechanism is being used within the tunnel that might be signalling congestion. In either case, the only appropriate signal to the transport is a packet drop. It would have been nice to allow packets with ECT(1) in the outer to be forwarded, but drop has had to be mandated in case future multi-level ECN schemes are defined. Then ECT(1) and CE can be used in the future to signify two levels of congestion severity.

  * If the inner ECN field is Not-ECT and the outer ECN field is ECT(0) or Not-ECT the decapsulator MUST forward the packet with the ECN field cleared to Not-ECT.
    Reasoning: Although no known legal protocol transition would lead to ECT(0) in the outer and Not-ECT in the inner, no known or proposed protocol uses ECT(0) as a congestion signal either. Therefore in this case the packet can be forwarded rather than dropped, which will allow future standards actions to use this combination.

- o In all other cases, 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;

- o There are cases where no currently legal transition in any current or previous ECN tunneling specification would result in certain combinations of inner and outer ECN fields. These cases are indicated in Figure 4 by '(!!!)'). In these cases, the decapsulator SHOULD log the event and MAY also raise an alarm, but not so often that the illegal combinations would amplify into a flood of alarm messages.

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). This approach is discussed in Appendix D and in the discussion of the ECN nonce [RFC3540] in Section 8.

### 4.3. Design Principles for Future Non-Default Schemes

This section is informative not normative.

S.5 of RFC3168 permits the Diffserv codepoint (DSCP)[RFC2474] to 'switch in' different behaviours for marking the ECN field, just as it switches in different per-hop behaviours (PHBs) for scheduling. Therefore here we give guidance for designing possibly different marking schemes.

In one word the guidance is "Don't". If a scheme requires tunnels to implement special processing of the ECN field for certain DSCPs, it is highly unlikely that every implementer of every tunnel will want to add the required exception and that operators will want to deploy the required configuration options. Therefore it is highly likely that some tunnels within a network will not implement this special case. Therefore, designers should avoid non-default tunnelling schemes if at all possible.

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

- o For any new scheme, a tunnel ingress should not set the ECN field of the outer header if it cannot guarantee that any corresponding tunnel egress will understand how to handle such an ECN field.

- On encapsulation in any new scheme, an outer header capable of carrying congestion markings should reflect accumulated congestion since the last interface designed to regulate load (see Appendix A.2 for the definition of a Load Regulator, which is usually but not always the data source). This implies that new schemes for tunnelling congestion notification should copy congestion notification into the outer header of each new encapsulating header that supports it.

  Reasoning: The constraints from the three perspectives of security, control and management in Appendix A are somewhat in tension as to whether a tunnel ingress should copy congestion markings into the outer header it creates or reset them. From the control perspective either copying or resetting works for existing arrangements, but copying has more potential for simplifying control. From the management perspective copying is preferable. From the security perspective resetting is preferable but copying is now considered acceptable given the bandwidth of a 2-bit covert channel can be managed. Therefore, on balance, copying is simpler and more useful than resetting and does minimal harm.

- For any new scheme, a tunnel egress should not forward any ECN codepoint if the arriving inner header implies the transport will not understand how to process it.

- On decapsulation in any new scheme, if a combination of inner and outer headers is encountered that should not have been possible, this event should be logged and an alarm raised. But the packet should still be forwarded with a safe codepoint setting if at all possible. This increases the chances of 'forward compatibility' with possible future protocol extensions.

- On decapsulation in any new scheme, the ECN field that the tunnel egress forwards should reflect the more severe congestion marking of the arriving inner and outer headers.

## 5. Backward Compatibility

Note: in RFC3168, a whole tunnel was considered in one of two modes: limited functionality or full functionality. The new modes defined in this specification are only modes of the tunnel ingress. The new tunnel egress behaviour has only one mode and doesn't need to know what mode the ingress is in.

### 5.1. Non-Issues Upgrading Any Tunnel Decapsulation

This specification only changes the egress per-packet calculation of the ECN field for combinations of inner and outer headers that have so far not been used in any IETF protocols. Therefore, a tunnel egress complying with any previous specification (RFC4301, both modes of RFC3168, both modes of RFC2481, RFC2401 and RFC2003) can be upgraded to comply with this new decapsulation specification without any backwards compatibility issues.

The proposed tunnel egress behaviour also requires no additional mode or option configuration at the ingress or egress nor any additional negotiation with the ingress. A compliant tunnel egress merely needs to implement the one behaviour in Section 4. The reduction to one mode at the egress has no backwards compatibility issues, because previously the egress produced the same output whichever mode the tunnel was in.

These 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). If a tunnel ingress tries to negotiate to use limited functionality mode or full functionality mode [RFC3168], a decapsulating tunnel egress compliant with this specification MUST agree to either request, as its behaviour will be the same in both cases.

For 'forward compatibility', a compliant tunnel egress SHOULD raise a warning about any requests to enter modes it doesn't recognise, but it can continue operating. If no ECN-related mode is requested, a compliant tunnel egress can continue without raising any error or warning as its egress behaviour is compatible with all the legacy ingress behaviours that don't negotiate capabilities.

### 5.2. Non-Issues for RFC4301 IPsec Encapsulation

The new normal mode of ingress behaviour defined above (Section 4.1) brings all IP in IP tunnels into line with [RFC4301]. If one end of an IPsec tunnel is compliant with [RFC4301], the other end is guaranteed to also be RFC4301-compliant (there could be corner cases where manual keying is used, but they will be set aside here). Therefore the new normal ingress behaviour introduces no backward compatibility isses with IKEv2 [RFC4306] IPsec [RFC4301] tunnels, and no need for any new modes, options or configuration.

### 5.3. Upgrading Other IP in IP Tunnel Encapsulators

At the tunnel ingress, this specification effectively extends the scope of RFC4301's ingress behaviour to any IP in IP tunnel. If any other IP in IP tunnel ingress (i.e. not RFC4301 IPsec) is upgraded to be compliant with this specification, it has to cater for the possibility that it is talking to a legacy tunnel egress that may not know how to process the ECN field. If ECN capable outer headers were sent towards a legacy (e.g. [RFC2003]) egress, it would most likely simply disregard the outer headers, dangerously discarding information about congestion experienced within the tunnel. ECN-capable traffic sources would not see any congestion feedback and instead continually ratchet up their share of the bandwidth without realising that cross-flows from other ECN sources were continually having to ratchet down.

This specification introduces no new backward compatibility issues when a compliant ingress talks with a legacy egress, but it has to provide similar sfaeguards to those already defined in RFC3168. Therefore, to comply with this specification, a tunnel ingress that does not always know the ECN capability of its tunnel egress MUST implement a 'normal' mode and a 'compatibility' mode, and for safety it MUST initiate each negotiated tunnel in compatibility mode.

However, a tunnel ingress can be compliant even if it only implements the 'normal mode' of encapsulation behaviour, but only as long as it is designed or configured so that all possible tunnel egress nodes it will ever talk to will have at least full ECN functionality (complying with either RFC3168 full functionality mode, RFC4301 or this present specification).

Before switching to normal mode, a compliant tunnel ingress that does not know the egress ECN capability MUST negotiate with the tunnel egress. If the egress says it is compliant with this specification or with RFC3168 full functionality mode, the ingress puts itself into normal mode. If the egress denies compliance with all of these or doesn't understand the question, the tunnel ingress MUST remain in compatibility mode.

The encapsulation rules for normal mode and compatibility mode are defined in Section 4 (i.e. header copying or zeroing respectively).

An ingress cannot claim compliance with this specification simply by disabling ECN processing across the tunnel (only implementing compatibility mode). Although such a tunnel ingress is at least safe with the ECN behaviour of any egress it may encounter (any of RFC2003, RFC2401, either mode of RFC2481 and RFC3168's limited functionality mode), it doesn't meet the aim of introducing ECN.

Therefore, a compliant tunnel ingress MUST at least implement `normal mode' and, if it might be used with arbitrary tunnel egress nodes, it MUST also implement `compatibility mode'.

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.

## 6. Changes from Earlier RFCs

On encapsulation, the rule that a normal mode tunnel ingress MUST copy any ECN field into the outer header is a change to the ingress behaviour of RFC3168, but it is the same as the rules for IPsec tunnels in RFC4301.

On decapsulation, the rules for calculating the outgoing ECN field at a tunnel egress are similar to the full functionality mode of ECN in RFC3168 and to RFC4301, with the following exceptions:

- 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 may be forwarded as Not-ECT rather than dropped, if the outer is ECT(0);

- The following extra illegal combinations have been identified, which may require logging and/or an alarm: outer ECT(1) with inner CE; outer ECT(0) with inner ECT(1)

The rules for how a tunnel establishes whether the egress has full functionality ECN capabilities are an update to RFC3168. For all the typical cases, RFC4301 is not updated by the ECN capability check in this specification, because a typical RFC4301 tunnel ingress will have already established that it is talking to an RFC4301 tunnel egress (e.g. if it uses IKEv2). However, there may be some corner cases (e.g. manual keying) where an RFC4301 tunnel ingress talks with an egress with limited functionality ECN handling. Strictly, for such corner cases, the requirement to use compatibility mode in this specification updates RFC4301, but this is unlikely to be necessary to implement for this corner case in practice.

The optional ECN Tunnel field in the IPsec security association database (SAD) and the optional ECN Tunnel Security Association Attribute defined in RFC3168 are no longer needed. The security association (SA) has no policy on ECN usage, because all RFC4301

tunnels now support ECN without any policy choice.

RFC3168 defines a (required) limited functionality mode and an (optional) full functionality mode for a tunnel, but RFC4301 doesn't need modes. In this specification only the ingress might need two modes: a normal mode (required) and a compatibility mode (required in some scenarios, optional in others). The egress needs only one mode which correctly handles any ingress ECN behaviour.

Additional changes to the RFC Index (to be removed by the RFC Editor):

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

This specification updates RFC3168 and RFC4301.

## 7. IANA Considerations

This memo includes no request to IANA.

## 8. Security Considerations

Appendix A.1 discusses the security constraints imposed on ECN tunnel processing. The new rules for ECN tunnel processing (Section 4) trade-off between 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.

If alternate congestion notification semantics are defined for a certain PHB (e.g. the pre-congestion notification architecture [I-D.ietf-pcn-architecture]), the scope of the alternate semantics might typically be bounded by the limits of a Diffserv region or regions, as envisaged in [RFC4774]. 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. [I-D.ietf-pcn-architecture] 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.

With the decapsulation rules as they stood in RFC3168 and RFC4301, a small part of the protection of the ECN nonce [RFC3540] was compromised. The new decapsulation rules do not solve this problem.

The minor problem is as follows: The ECN nonce was defined to enable the data source to detect if a CE marking had been applied then subsequently removed. The source could detect this by weaving a pseudo-random sequence of ECT(0) and ECT(1) values into a stream of packets, which is termed an ECN nonce. By the decapsulation rules in RFC3168 and RFC4301, if the inner and outer headers carry contradictory ECT values only the inner header is preserved for onward forwarding. So if a CE marking added to the outer ECN field in a tunnel has been illegally (or accidentally) suppressed by a subsequent node in the tunnel, the decapsulator will revert the ECN field to its value before tampering, hiding all evidence of the crime from the onward feedback loop. We chose not to close this minor loophole for all the following reasons:

1. This loophole is only applicable in the corner case where the attacker controls a network node downstream of a congested node in the same tunnel;

2. In tunnelling scenarios, the ECN nonce is already vulnerable to suppression by nodes downstream of a congested node in the same tunnel, if they can copy the ECT value in the inner header to the outer header (any node in the tunnel can do this if the inner header is not encrypted, and an IPsec tunnel egress can do it whether or not the tunnel is encrypted);

3. Although the new decapsulation behaviour removes evidence of congestion suppression from the onward feedback loop, the decapsulator itself can at least detect that congestion within the tunnel has been suppressed;

4. The ECN nonce [RFC3540] currently has experimental status and there has been no evidence that anyone has implemented it beyond the author's prototype.

We could have fixed this loophole by specifying that the outer header should always be propagated onwards if inner and outer are both ECT. Although this would close the minor loophole in the nonce, it would raise a minor safety issue if multilevel ECN or PCN were used. A less severe marking in the inner header would override a more severe one in the outer. Both are corner cases so it is difficult to decide which is more important:

1. The loophole in the nonce is only for a minor case of one tunnel node attacking another in the same tunnel;

2. The severity inversion for multilevel congestion notification would not result from any legal codepoint transition.

We decided safety against misconfiguration was slightly more important than securing against an attack that has little, if any, clear motivation.

If a legacy security policy configures a legacy tunnel ingress to negotiate to turn off ECN processing, a compliant tunnel egress will agree to a request to turn off ECN processing but it will actually still copy CE markings from the outer to the forwarded header. Although the tunnel ingress 'I' in Figure 5 (Appendix A.1) will set all ECN fields in outer headers to Not-ECT, 'M' could still toggle CE 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 a legacy security concern that is now considered manageable.

## 9. Conclusions

This document updates the ingress tunnelling encapsulation of RFC3168 ECN for all IP in IP tunnels to bring it into line with the new behaviour in the IPsec architecture of RFC4301. It copies rather than resets a congestion experienced (CE) marking when creating outer headers.

It also specifies new rules that update both RFC3168 and RFC4301 for calculating the outgoing ECN field on tunnel decapsulation. The new rules update egress behaviour for two specific combinations of inner and outer header that have no current legal usage, but will now be possible to use in future standards actions, rather than being wasted by current tunnelling behaviour.

The new rules propagate changes to the ECN field across tunnel end-points that were previously blocked due to a perceived covert channel vulnerability. The new IPsec architecture deems the two-bit covert channel that the ECN field opens up is a manageable threat, so these new rules bring all IP in IP tunnelling into line with this new more permissive attitude. The result is a single specification for all future tunnelling of ECN, whether IPsec or not. Then equipment can be specified against a single ECN behaviour and ECN markings can have

a well-defined meaning wherever they are measured in a network. This new certainty will enable new uses of the ECN field that would otherwise be confounded by ambiguity.

The immediate motivation for making these changes is to allow the introduction of multi-level pre-congestion notification (PCN). But great care has been taken to ensure the resulting ECN tunnelling behaviour is simple and generic for other potential future uses.

The change to encapsulation has been analysed from the three perspectives of security, control and management. They are somewhat in tension as to whether a tunnel ingress should copy congestion markings into the outer header it creates or reset them. 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. From the management and monitoring perspective copying is preferable. From the network security perspective (theft of service etc) copying is preferable. 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 no points against copying and a number against resetting CE on ingress.

The only downside of the changes to decapsulation is that the same 2-bit covert channel is opened up as at the ingress, but this is now deemed to be a manageable threat. The changes at decapsulation have been found to be free of any backwards compatibility issues.

## 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 doesn't understand. Thanks to David Black for explaining a better way to think about function placement and to Louise Burness for a better way to think about multilayer transports and networks, having read [Patterns_Arch]. 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 and Gabriele Corliano for their thoughts and careful review comments.

Bob Briscoe is partly funded by Trilogy, a research project (ICT-216372) supported by the European Community under its Seventh Framework Programme. The views expressed here are those of the author only.

## 11. 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.

## 12. References

### 12.1. 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.

[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.

[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.

### 12.2. Informative References

[I-D.briscoe-pcn-3-in-1-encoding]
Briscoe, B., "PCN 3-State Encoding Extension in a single DSCP", draft-briscoe-pcn-3-in-1-encoding-00 (work in progress), October 2008.

[I-D.charny-pcn-single-marking]
Charny, A., Zhang, X., Faucheur, F., and V. Liatsos, "Pre-Congestion Notification Using Single Marking for Admission and Termination", draft-charny-pcn-single-marking-03 (work in progress), November 2007.

[I-D.ietf-pcn-architecture]
Eardley, P., "Pre-Congestion Notification (PCN) Architecture", draft-ietf-pcn-architecture-10 (work in progress), March 2009.

[I-D.ietf-pcn-baseline-encoding]

Moncaster, T., Briscoe, B., and M. Menth, "Baseline Encoding and Transport of Pre-Congestion Information", draft-ietf-pcn-baseline-encoding-02 (work in progress), February 2009.

[I-D.ietf-pcn-marking-behaviour]
Eardley, P., "Marking behaviour of PCN-nodes", draft-ietf-pcn-marking-behaviour-02 (work in progress), March 2009.

[I-D.ietf-pwe3-congestion-frmwk]
Bryant, S., Davie, B., Martini, L., and E. Rosen, "Pseudowire Congestion Control Framework", draft-ietf-pwe3-congestion-frmwk-01 (work in progress), May 2008.

[I-D.menth-pcn-psdm-encoding]
Menth, M., Babiarz, J., Moncaster, T., and B. Briscoe, "PCN Encoding for Packet-Specific Dual Marking (PSDM)", draft-menth-pcn-psdm-encoding-00 (work in progress), July 2008.

[I-D.moncaster-pcn-3-state-encoding]
Moncaster, T., Briscoe, B., and M. Menth, "A three state extended PCN encoding scheme", draft-moncaster-pcn-3-state-encoding-01 (work in progress), March 2009.

[I-D.satoh-pcn-st-marking]
Satoh, D., Maeda, Y., Phanachet, O., and H. Ueno, "Single PCN Threshold Marking by using PCN baseline encoding for both admission and termination controls", draft-satoh-pcn-st-marking-01 (work in progress), March 2009.

[IEEE802.1au]
IEEE, "IEEE Standard for Local and Metropolitan Area Networks--Virtual Bridged Local Area Networks - Amendment 10: Congestion Notification", 2008, <http://www.ieee802.org/1/pages/802.1au.html>.

(Work in Progress; Access Controlled link within page)

[ITU-T.I.371]
ITU-T, "Traffic Control and Congestion Control in B-ISDN", ITU-T Rec. I.371 (03/04), March 2004.

[PCNcharter]

IETF, "Congestion and Pre-Congestion Notification (pcn)", IETF w-g charter , Feb 2007, <http://www.ietf.org/html.charters/pcn-charter.html>.

[Patterns_Arch] Day, J., "Patterns in Network Architecture: A Return to Fundamentals", Pub: Prentice Hall ISBN-13: 9780132252423, Jan 2008.

[RFC1254] Mankin, A. and K. Ramakrishnan, "Gateway Congestion Control Survey", RFC 1254, August 1991.

[RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

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

[RFC3426] Floyd, S., "General Architectural and Policy Considerations", RFC 3426, November 2002.

[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.

[RFC4423] Moskowitz, R. and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423, May 2006.

[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.

[Shayman] "Using ECN to Signal Congestion Within an MPLS Domain", 2000, <http://www.ee.umd.edu/~shayman/papers.d/draft-shayman-mpls-ecn-00.txt>.

(Expired)

## Appendix A. 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.

### A.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.

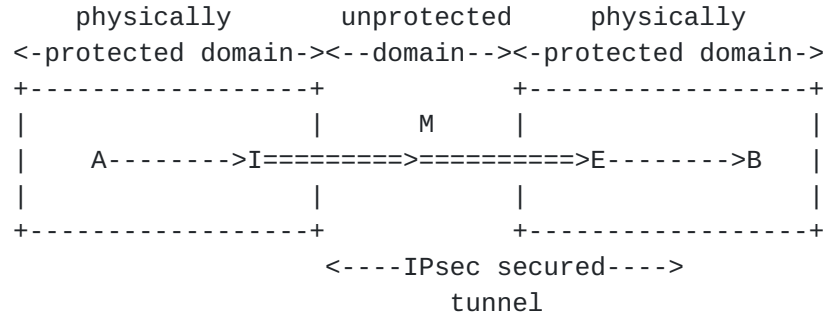


Figure 5: IPsec Tunnel Scenario

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 is typically intended to create an information channel from 'M' to 'B' (for 'B' to feed back to 'A'). Therefore the goals of IPsec and ECN are mutually incompatible.

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 therefore 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.

Thus, 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.

## A.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).

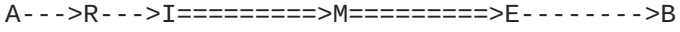


Figure 6: Simple Tunnel Scenario

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 doesn't 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 at a tunnel ingress confuses 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.

### A.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 since the original source, perhaps to determine between them how much of the congestion is contributed by each domain.

Therefore, when monitoring the middle of a path, it should be possible to establish how far back in the path congestion markings have accumulated from. In this document we term this the baseline of congestion marking (or the Congestion Baseline), i.e. the source of the layer that last reset (or created) the congestion notification field. 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 exposed across the tunnel.

Appendix B.2 discusses various scenarios where the Load Regulator lies in-path, not at the source host as we would typically expect. It concludes that a Congestion Baseline is determined by where the Load Regulator function is, which should be identified in the transport layer, not by addresses in network layer headers. This applies whether the Load Regulator is at the source host or within the path. The appendix also discusses where a Load Regulator function should be located relative to a local tunnel encapsulation function.

# Appendix B. Relative Placement of Tunnelling and In-Path Load Regulation

## B.1. Identifiers and In-Path Load Regulators

The Load Regulator is the node to which congestion feedback should be returned by the next downstream node with a transport layer feedback function (typically but not always the data receiver). The Load Regulator is often, but not always the data source. It is not always (or even typically) the same thing as the node identified by the source address of the outermost exposed header. In general the addressing of the outermost encapsulation header says nothing about the identifiers of either the upstream or the downstream transport layer functions. As long as the transport functions know each other's addresses, they don't have to be identified in the network layer or in any link layer. It was only a convenience that a TCP receiver assumed that the address of the source transport is the same as the network layer source address of an IP packet it receives.

More generally, the return transport address for feedback could be identified solely in the transport layer protocol. For instance, a signalling protocol like RSVP [RFC2205] breaks up a path into transport layer hops and informs each hop of the address of its transport layer neighbour without any need to identify these hops in the network layer. RSVP can be arranged so that these transport layer hops are bigger than the underlying network layer hops. The host identity protocol (HIP) architecture [RFC4423] also supports the same principled separation (for mobility amongst other things), where the transport layer sender identifies its transport address for feedback to be sent to, using an identifier provided by a shim below the transport layer.

Keeping to this layering principle deliberately doesn't require a network layer packet header to reveal the origin address from where congestion notification accumulates (its Congestion Baseline). It is not necessary for the network and lower layers to know the address of the Load Regulator. Only the destination transport needs to know that. With forward congestion notification, the network and link layers only notify congestion forwards; they aren't involved in feeding it backwards. If they are (e.g. backward congestion notification (BCN) in Ethernet [IEEE802.1au] or EFCI in ATM [ITU-T.I.371]), that should be considered as a transport function added to the lower layer, which must sort out its own addressing. Indeed, this is one reason why ICMP source quench is now deprecated [RFC1254]; when congestion occurs within a tunnel it is complex (particularly in the case of IPsec tunnels) to return the ICMP messages beyond the tunnel ingress back to the Load Regulator.

Similarly, if a management system is monitoring congestion and needs to know the Congestion Baseline, the management system has to find this out from the transport; in general it cannot tell solely by looking at the network or link layer headers.

### B.2. Non-Dependence of Tunnelling on In-path Load Regulation

We have said that at any point in a network, the Congestion Baseline (where congestion notification starts from zero) should be the previous upstream Load Regulator. We have also said that the ingress of an IP in IP tunnel must copy congestion indications to the encapsulating outer headers it creates. If the Load Regulator is in-path rather than at the source, and also a tunnel ingress, these two requirements seem to be contradictory. A tunnel ingress must not reset incoming congestion, but a Load Regulator must be the Congestion Baseline, implying it needs to reset incoming congestion.

In fact, the two requirements are not contradictory, because a Load Regulator and a tunnel ingress are not the names of machines, but the names of functions within a machine that typically occur in sequence on a stream of packets, not at the same point. Figure 7 is borrowed from [RFC2983] (which was making a similar point about the location of Diffserv traffic conditioning relative to the encapsulation function of a tunnel). An in-path Load Regulator can act on packets either at [1 - Before] encapsulation or at [2 - Outer] after encapsulation. Load Regulation does not ever need to be integrated with the [Encapsulate] function (but it can be for efficiency). Therefore we can still mandate that the [Encapsulate] function always copies CE into the outer header.

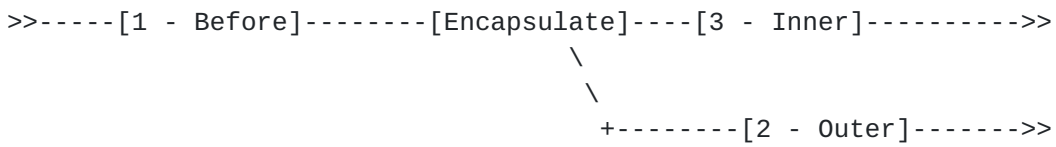


Figure 7: Placement of In-Path Load Regulator Relative to Tunnel Ingress

Then separately, if there is a Load Regulator at location [2 - Outer], it might reset CE to ECT(0), say. Then the Congestion Baseline for the lower layer (outer) will be [2 - Outer], while the Congestion Baseline of the inner layer will be unchanged. But how encapsulation works has nothing to do with whether a Load Regulator is present or where it is.

If on the other hand a Load Regulator resets CE at [1 - Before], the

Congestion Baseline of both the inner and outer headers will be [1 - Before]. But again, encapsulation is independent of load regulation.

### B.3. Dependence of In-Path Load Regulation on Tunnelling

Although encapsulation doesn't need to depend on in-path load regulation, the reverse is not true. The placement of an in-path Load Regulator must be carefully considered relative to encapsulation. Some examples are given in the following for guidance.

In the traditional Internet architecture one tends to think of the source host as the Load Regulator for a path. It is generally not desirable or practical for a node part way along the path to regulate the load. However, various reasonable proposals for in-path load regulation have been made from time to time (e.g. fair queuing, traffic engineering, flow admission control). The IETF has recently chartered a working group to standardise admission control across a part of a path using pre-congestion notification (PCN) [PCNcharter]. This is of particular relevance here because it involves congestion notification with an in-path Load Regulator, it can involve tunnelling and it certainly involves encapsulation more generally.

We will use the more complex scenario in Figure 8 to tease out all the issues that arise when combining congestion notification and tunnelling with various possible in-path load regulation schemes. In this case 'I1' and 'E2' break up the path into three separate congestion control loops. The feedback for these loops is shown going right to left across the top of the figure. The 'V's are arrow heads representing the direction of feedback, not letters. But there are also two tunnels within the middle control loop: 'I1' to 'E1' and 'I2' to 'E2'. The two tunnels might be VPNs, perhaps over two MPLS core networks. M is a congestion monitoring point, perhaps between two border routers where the same tunnel continues unbroken across the border.

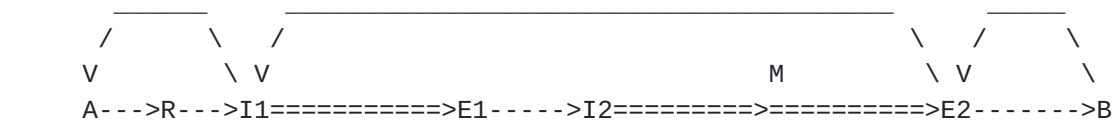


Figure 8: Complex Tunnel Scenario

The question is, should the congestion markings in the outer exposed headers of a tunnel represent congestion only since the tunnel ingress or over the whole upstream path from the source of the inner header (whatever that may mean)? Or put another way, should 'I1' and 'I2' copy or reset CE markings?

Based on the design principles in Section 4.3, the answer is that the Congestion Baseline should be the nearest upstream interface designed to regulate traffic load--the Load Regulator. In Figure 8 'A', 'I1' or 'E2' are all Load Regulators. We have shown the feedback loops returning to each of these nodes so that they can regulate the load causing the congestion notification. So the Congestion Baseline exposed to M should be 'I1' (the Load Regulator), not 'I2'. Therefore I1 should reset any arriving CE markings. In this case, 'I1' knows the tunnel to 'E1' is unrelated to its load regulation function. So the load regulation function within 'I1' should be placed at [1 - Before] tunnel encapsulation within 'I1' (using the terminology of Figure 7). Then the Congestion Baseline all across the networks from 'I1' to 'E2' in both inner and outer headers will be 'I1'.

The following further examples illustrate how this answer might be applied:

- We argued in Appendix E that resetting CE on encapsulation could harm PCN excess rate marking, which marks excess traffic for removal in subsequent round trips. This marking relies on not marking packets if another node upstream has already marked them for removal. If there were a tunnel ingress between the two which reset CE markings, it would confuse the downstream node into marking far too much traffic for removal. So why do we say that 'I1' should reset CE, while a tunnel ingress shouldn't? The answer is that it is the Load Regulator function at 'I1' that is resetting CE, not the tunnel encapsulator. The Load Regulator needs to set itself as the Congestion Baseline, so the feedback it gets will only be about congestion on links it can relieve itself (by regulating the load into them). When it resets CE markings, it knows that something else upstream will have dealt with the congestion notifications it removes, given it is part of an end-to-end admission control signalling loop. It therefore knows that previous hops will be covered by other Load Regulators. Meanwhile, the tunnel ingresses at both 'I1' and 'I2' should follow the new rule for any tunnel ingress and copy congestion marking into the outer tunnel header. The ingress at 'I1' will happen to copy headers that have already been reset just beforehand. But it doesn't need to know that.

- [Shayman] suggested feedback of ECN accumulated across an MPLS domain could cause the ingress to trigger re-routing to mitigate congestion. This case is more like the simple scenario of Figure 6, with a feedback loop across the MPLS domain ('E' back to 'I'). I is a Load Regulator because re-routing around congestion is a load regulation function. But in this case 'I' should only reset itself as the Congestion Baseline in outer headers, as it is

not handling congestion outside its domain, so it must preserve the end-to-end congestion feedback loop for something else to handle (probably the data source). Therefore the Load Regulator within 'I' should be placed at [2 - Outer] to reset CE markings just after the tunnel ingress has copied them from arriving headers. Again, the tunnel encapsulation function at 'I' simply copies incoming headers, unaware that the load regulator will subsequently reset its outer headers.

- The PWE3 working group of the IETF is considering the problem of how and whether an aggregate edge-to-edge pseudo-wire emulation should respond to congestion [I-D.ietf-pwe3-congestion-frmwk]. Although the study is still at the requirements stage, some (controversial) solution proposals include in-path load regulation at the ingress to the tunnel that could lead to tunnel arrangements with similar complexity to that of Figure 8.

These are not contrived scenarios--they could be a lot worse. For instance, a host may create a tunnel for IPsec which is placed inside a tunnel for Mobile IP over a remote part of its path. And around this all we may have MPLS labels being pushed and popped as packets pass across different core networks. Similarly, it is possible that subnets could be built from link technology (e.g. future Ethernet switches) so that link headers being added and removed could involve congestion notification in future Ethernet link headers with all the same issues as with IP in IP tunnels.

One reason we introduced the concept of a Load Regulator was to allow for in-path load regulation. In the traditional Internet architecture one tends to think of a host and a Load Regulator as synonymous, but when considering tunnelling, even the definition of a host is too fuzzy, whereas a Load Regulator is a clearly defined function. Similarly, the concept of innermost header is too fuzzy to be able to (wrongly) say that the source address of the innermost header should be the Congestion Baseline. Which is the innermost header when multiple encapsulations may be in use? Where do we stop? If we say the original source in the above IPsec-Mobile IP case is the host, how do we know it isn't tunnelling an encrypted packet stream on behalf of another host in a p2p network?

We have become used to thinking that only hosts regulate load. The end to end design principle advises that this is a good idea [RFC3426], but it also advises that it is solely a guiding principle intended to make the designer think very carefully before breaking it. We do have proposals where load regulation functions sit within a network path for good, if sometimes controversial, reasons, e.g. PCN edge admission control gateways [I-D.ietf-pcn-architecture] or traffic engineering functions at domain borders to re-route around

congestion [Shayman]. Whether or not we want in-path load regulation, we have to work round the fact that it will not go away.

## 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. 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 to measure. It is the packets with additional CE marking in the outer header (12) as a proportion of packets not marked in the inner header (70).

Figure 9 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 along the tunnel.

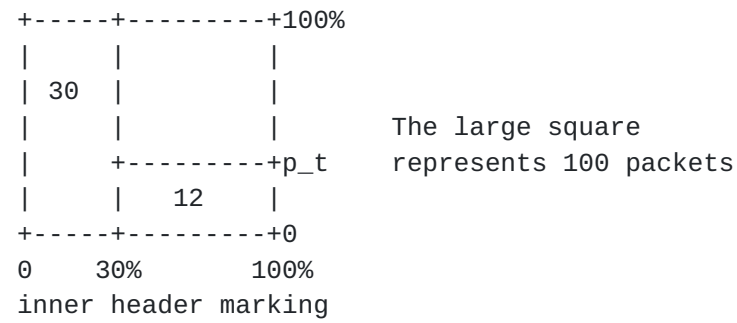


Figure 9: Tunnel Marking of Packets Already Marked at Ingress

## Appendix D. Why Not Propagating ECT(1) on Decapsulation Impedes PCN

Multi-level congestion notification is currently on the IETF's standards track agenda in the Congestion and Pre-Congestion Notification (PCN) working group. The PCN working group eventually requires three congestion states (not marked and two increasingly severe levels of congestion marking) [I-D.ietf-pcn-architecture]. The aim is for the less severe level of marking to stop admitting new traffic and the more severe level to terminate sufficient existing flows to bring a network back to its operating point after a serious failure.

Although the ECN field gives sufficient codepoints for these three states, current ECN tunnelling RFCs prevent the PCN working group from using three ECN states in case any tunnel decapsulations occur within a PCN region (see Appendix A of [I-D.ietf-pcn-baseline-encoding]). If a node in a tunnel sets the ECN field to ECT(0) or ECT(1), this change will be discarded by a tunnel egress compliant with RFC4301 or RFC3168. This can be seen in Figure 2 (Section 3.2), where ECT values in the outer header are ignored unless the inner header is the same. Effectively one ECT codepoint is wasted; the ECT(0) and ECT(1) codepoints have to be treated as just one codepoint when they could otherwise have been used for their intended purpose of congestion notification.

As a consequence, the PCN w-g has initially confined itself to two encoding states as a baseline encoding [I-D.ietf-pcn-baseline-encoding]. And it has had to propose an experimental extension using extra Diffserv codepoint(s) to encode the extra states [I-D.moncaster-pcn-3-state-encoding], using up the rapidly exhausting DSCP space while leaving ECN codepoints unused. Another PCN encoding has been proposed that would survive tunnelling without an extra DSCP [I-D.menth-pcn-psdm-encoding], but it requires the PCN edge gateways to somehow share state so the egress can determine which marking a packet started with at the ingress. Also a PCN ingress node can game the system by initiating packets with inappropriate markings. Yet another work-round to the ECN tunnelling problem proposes a more involved marking algorithm in the forwarding plane to encode the three congestion notification states using only two ECN codepoints [I-D.satoh-pcn-st-marking]. Still another proposal compromises the precision of the admission control mechanism, but manages to work with just two encoding states and a single marking algorithm [I-D.charny-pcn-single-marking].

Rather than require the IETF to bless any of these work-rounds, this specification fixes the root cause of the problem so that operators deploying PCN can simply ask that tunnel end-points within a PCN region should comply with this new ECN tunnelling specification.

Then PCN can use the trivially simple experimental 3-state ECN encoding defined in [I-D.briscoe-pcn-3-in-1-encoding].

### D.1. Alternative Ways to Introduce the New Decapsulation Rules

There are a number of ways for the new decapsulation rules to be introduced:

- They could be specified in the present standards track proposal (preferred) or in an experimental extension;

- They could be specified as a new default for all Diffserv PHBs (preferred) or as an option to be configured only for Diffserv PHBs requiring them (e.g. PCN).

The argument for making this change now, rather than in a separate experimental extension, is to avoid the burden of an extra standard to be compliant with and to be backwards compatible with--so we don't add to the already complex history of ECN tunnelling RFCs. The argument for a separate experimental extension is that we may never need this change (if PCN is never successfully deployed and if no-one ever needs three ECN or PCN encoding states rather than two). However, the change does no harm to existing mechanisms and stops tunnels wasting of quarter of a bit (a 2-bit codepoint).

The argument for making this new decapsulation behaviour the default for all PHBs is that it doesn't change any expected behaviour that existing mechanisms rely on already. Also, by ending the present waste of a codepoint, in the future a use of that codepoint could be proposed for all PHBs, even if PCN isn't successfully deployed.

In practice, if these new decapsulation rules are specified straightaway as the normative default for all PHBs, a network operator deploying 3-state PCN would be able to request that tunnels comply with the latest specification. Implementers of non-PCN tunnels would not need to comply but, if they did, their code would be future proofed and no harm would be done to legacy operations. Therefore, rather than branching their code base, it would be easiest for implementers to make all their new tunnel code comply with this specfication, whether or not it was for PCN. But they could leave old code untouched, unless it was for PCN.

The alternatives are worse. Implementers would otherwise have to provide configurable decapsulation options and operators would have to configure all IPsec and IP in IP tunnel endpoints for the exceptional behaviour of certain PHBs. The rules for tunnel endpoints to handle both the Diffserv field and the ECN field should 'just work' when handling packets with any Diffserv codepoint.

## Appendix E. Why Resetting CE on Encapsulation Impedes PCN

Regarding encapsulation, the section of the PCN architecture [I-D.ietf-pcn-architecture] on tunnelling says that header copying (RFC4301) allows PCN to work correctly. Whereas resetting CE markings confuses PCN marking.

The specific issue here concerns PCN excess rate marking [I-D.ietf-pcn-marking-behaviour], i.e. the bulk marking of traffic that exceeds a configured threshold rate. One of the goals of excess rate marking is to enable the speedy removal of excess admission controlled traffic following re-routes caused by link failures or other disasters. This maintains a share of the capacity for traffic in lower priority classes. After failures, traffic re-routed onto remaining links can often stress multiple links along a path. Therefore, traffic can arrive at a link under stress with some proportion already marked for removal by a previous link. By design, marked traffic will be removed by the overall system in subsequent round trips. So when the excess rate marking algorithm decides how much traffic to mark for removal, it doesn't include traffic already marked for removal by another node upstream (the `Excess traffic meter function' of [I-D.ietf-pcn-marking-behaviour]).

However, if an RFC3168 tunnel ingress intervenes, it resets the ECN field in all the outer headers, hiding all the evidence of problems upstream. Thus, although excess rate marking works fine with RFC4301 IPsec tunnels, with RFC3168 tunnels it typically removes large volumes of traffic that it didn't need to remove at all.

## 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://www.cs.ucl.ac.uk/staff/B.Briscoe/