

**Emulating Border Flow Policing using Re-ECN on Bulk Data
draft-briscoe-tsvwg-re-ecn-border-cheat-01**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of 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 December 28, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

Scaling per flow admission control to the Internet is a hard problem. A recently proposed approach combines Diffserv and pre-congestion notification (PCN) to provide a service slightly better than Intserv controlled load. It scales to networks of any size, but only if domains trust each other to comply with admission control and rate policing. This memo claims to solve this trust problem without losing scalability. It describes bulk border policing that provides a sufficient emulation of per-flow policing with the help of another recently proposed extension to ECN, involving re-echoing ECN feedback

(re-ECN). With only passive bulk measurements at borders, sanctions can be applied against cheating networks.

Status (to be removed by the RFC Editor)

This memo is posted as an Internet-Draft with the intent to eventually progress to informational status. It is envisaged that the necessary standards actions to realise the system described would sit in three other documents currently being discussed (but not on the standards track) in the IETF Transport Area [[Re-TCP](#)], [[RSVP-ECN](#)] & [[PCN](#)]. The authors seek comments from the Internet community on whether combining PCN and re-ECN is a sufficient solution to the admission control problem.

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

From -00 to -01:

Added subsection on Border Accounting Mechanisms ([Section 5.6.1](#))

[Section 4.2](#) on the re-ECN wire protocol clarified and re-organised to separately discuss re-ECN for default ECN marking and for pre-congestion marking (PCN).

Router Forwarding Behaviour subsection added to re-organised section on Protocol Operation ([Section 4.3](#)). Extensions section moved within Protocol Operations.

Emulating Border Policing ([Section 5](#)) reorganised, starting with a new Terminology subsection heading, and a simplified overview section. Added a large new subsection on Border Accounting Mechanisms within a new section bringing together other subsections on Border Mechanisms generally ([Section 5.6](#)). Some text moved from old subsections into these new ones.

Added section on Incremental Deployment ([Section 7](#)), drawing together relevant points about deployment made throughout.

Sections on Design Rationale ([Section 8](#)) and Security Considerations ([Section 9](#)) expanded with some new material, including new attacks and their defences.

Suggested Border Metering Algorithms improved (Appendix A.2) for resilience to newly identified attacks.

Table of Contents

1.	Introduction	5
2.	Requirements Notation	7
3.	The Problem	7
3.1.	The Traditional Per-flow Policing Problem	7
3.2.	Generic Scenario	9
4.	Re-ECN Protocol for an RSVP (or similar) Transport	11
4.1.	Protocol Overview	11
4.2.	Re-ECN Abstracted Network Layer Wire Protocol (IPv4 or v6)	13
4.2.1.	Re-ECN Recap	13
4.2.2.	Re-ECN Combined with Pre-Congestion Notification (re-PCN)	14
4.3.	Protocol Operation	17
4.3.1.	Protocol Operation for an Established Flow	17
4.3.2.	Aggregate Bootstrap	18
4.3.3.	Flow Bootstrap	19
4.3.4.	Router Forwarding Behaviour	20
4.3.5.	Extensions	22
5.	Emulating Border Policing with Re-ECN	22
5.1.	Informal Terminology	22
5.2.	Policing Overview	23
5.3.	Pre-requisite Contractual Arrangements	25
5.4.	Emulation of Per-Flow Rate Policing: Rationale and Limits	28
5.5.	Sanctioning Dishonest Marking	29
5.6.	Border Mechanisms	31
5.6.1.	Border Accounting Mechanisms	31
5.6.2.	Competitive Routing	35
5.6.3.	Fail-safes	35
6.	Analysis	36
7.	Incremental Deployment	39
8.	Design Choices and Rationale	40
9.	Security Considerations	41
10.	IANA Considerations	43
11.	Conclusions	43
12.	Acknowledgements	44
13.	Comments Solicited	44
14.	References	44
14.1.	Normative References	44
14.2.	Informative References	45
Appendix A.	Implementation	46
A.1.	Ingress Gateway Algorithm for Blanking the RE flag	47
A.2.	Downstream Congestion Metering Algorithms	47
A.2.1.	Bulk Downstream Congestion Metering Algorithm	47
A.2.2.	Inflation Factor for Persistently Negative Flows	48
A.3.	Algorithm for Sanctioning Negative Traffic	49

Briscoe

Expires December 28, 2006

[Page 3]

Author's Address	50
Intellectual Property and Copyright Statements	51

1. Introduction

The Internet community largely lost interest in the Intserv architecture after it was clarified that it would be unlikely to scale to the whole Internet [[RFC2208](#)]. Although Intserv mechanisms proved impractical, the bandwidth reservation service it aimed to offer is still very much required.

A recently proposed approach [[CL-deploy](#)] combines Diffserv and pre-congestion notification (PCN) to provide a service slightly better than Intserv controlled load [[RFC2211](#)]. It scales to any size network, but only if domains trust their neighbours to have checked that upstream customers aren't taking more bandwidth than they reserved, either accidentally or deliberately. This memo describes border policing measures so that one network can protect its interests, even if networks around it are deliberately trying to cheat. The approach provides a sufficient emulation of flow rate policing at trust boundaries but without per-flow processing. The emulation is not perfect, but it is sufficient to ensure that the punishment is at least proportionate to the severity of the cheat.

The aim is to be able to scale controlled load service to any number of endpoints, even though such scaling must take account of the increasing numbers of networks and users who may all have conflicting interests. To achieve such scaling, this memo combines two recent proposals, both of which it briefly recaps:

- o A deployment model for admission control over Diffserv using pre-congestion notification [[CL-deploy](#)] describes how bulk pre-congestion notification on routers within an edge-to-edge Diffserv region can emulate the precision of per-flow admission control to provide controlled load service without unscalable per-flow processing;
- o Re-ECN: Adding Accountability to TCP/IP [[Re-TCP](#)]. The trick that addresses cheating at borders is to recognise that border policing is mainly necessary because cheating upstream networks will admit traffic when they shouldn't only as long as they don't directly experience the downstream congestion their misbehaviour can cause. The re-ECN protocol requires upstream nodes to declare expected downstream congestion in all forwarded packets and it makes it in their interests to declare it honestly. Operators can then monitor downstream congestion in bulk at borders to emulate policing.

Rather than the end-to-end arrangement used when re-ECN was specified for the TCP transport [[Re-TCP](#)], this memo specifies re-ECN in an edge-to-edge arrangement, making it applicable to the above

deployment model for admission control over Diffserv. Also, rather than using a TCP transport for regular congestion feedback, this memo specifies re-ECN using RSVP as the transport for feedback [[RSVP-ECN](#)]. A similar deployment model, but with a different transport for signalling congestion feedback could be used (e.g. RMD [[NSIS-RMD](#)] uses NSIS).

This memo aims to do two things: i) define how to apply the re-ECN protocol to the admission control over Diffserv scenario; and ii) explain why re-ECN sufficiently emulates border policing in that scenario. Most of the memo is taken up with the second aim; explaining why it works. Applying re-ECN to the scenario actually involves quite a trivial modification to the ingress gateway. Our immediate goal is to convince everyone to build that modification in to ingress gateways from the start, whether first deployments require policing or not. Otherwise, when we want to add policing, we will have built ourselves a legacy problem. In other words, we aim to convince people to "Build in security from the start."

The body of this memo is structured as follows:

[Section 3](#) describes the border policing problem. We recap the traditional, unscalable view of how to solve the problem, and we recap the admission control solution which has the scalability we do not want to lose when we add border policing;

[Section 4](#) specifies the re-ECN protocol solution in detail;

[Section 5](#) explains how to use the protocol to emulate border policing, and why it works;

[Section 6](#) analyses the security of the proposed solution;

[Section 8](#) explains the sometimes subtle rationale behind our design decisions;

[Section 9](#) comments on the overall robustness of the security assumptions and lists specific security issues.

It must be emphasised that we are not evangelical about removing per-flow processing from borders. Network operators may choose to do per-flow processing at their borders for their own reasons, such as to support business models that require per-flow accounting. Our aim is to show that per-flow processing at borders is no longer /necessary/ in order to provide end-to-end QoS using flow admission control. Indeed, we are absolutely opposed to standardisation of technology that embeds particular business models into the Internet. Our aim is merely to provide a new useful metric (downstream

congestion) at trust boundaries. Given the well-known significance of congestion in economics, operators can then use this new metric in their interconnection contracts if they choose. This will enable competitive evolution of new business models (for examples see [\[IXQoS\]](#)), alongside more traditional models that depend on more costly per-flow processing at borders.

2. Requirements Notation

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

3. The Problem

3.1. The Traditional Per-flow Policing Problem

If we claim to be able to emulate per-flow policing with bulk policing at trust boundaries, we need to know exactly what we are emulating. So, even though we expect it to become a historic practice, we will start from the traditional scenario with per-flow policing at trust boundaries to explain why it has always been considered necessary.

To be able to take advantage of a reservation-based service such as controlled load, a source must reserve resources using a signalling protocol such as RSVP [\[RFC2205\]](#). An RSVP signalling request refers to a flow of packets by its flow ID tuple (filter spec [\[RFC2205\]](#)) (or its security parameter index (SPI) [\[RFC2207\]](#) if port numbers are hidden by IPsec encryption). Other signalling protocols use similar flow identifiers. But, it is insufficient to merely authorise and admit a flow based on its identifiers, for instance merely opening a pin-hole for packets with identifiers that match an admitted flow ID. Once a flow is admitted, it cannot necessarily be trusted to send packets within the rate profile it requested.

The packet rate must also be policed to keep the flow within the requested flow spec [\[RFC2205\]](#). For instance, without data rate policing, a source could reserve resources for an 8kbps audio flow but transmit a 6Mbps video (theft of service). More subtly, the sender could generate bursts that were outside the profile it had requested.

In traditional architectures, per-flow packet rate-policing is expensive and unscalable but, without it, a network is vulnerable to such theft of service (whether malicious or accidental). Perhaps

more importantly, if flows are allowed to send more data than they were permitted, the ability of admission control to give assurances to other flows will break.

Just as sources need not be trusted to keep within their requested flow spec, whole networks might also try to cheat. We will now set up a concrete scenario to illustrate such cheats. Imagine reservations for unidirectional flows from senders, through at least two networks, an edge network and its downstream transit provider. Imagine the edge network charges its retail customers per reservation but also has to pay its transit provider a charge per reservation. Typically, both its selling and buying charges might depend on the duration and rate of each reservation. The level of the actual selling and buying prices are irrelevant to our discussion (most likely the network will sell at a higher price than it buys, of course).

A cheating ingress network could systematically reduce the size of its retail customers' reservation signalling requests before forwarding them to its transit provider (and systematically reinstate the responses on the way back). It would then receive an honest income from its upstream retail customer but only pay for fraudulently smaller reservations downstream. Equivalently, a cheating ingress network may feed the traffic from a number of flows into an aggregate reservation over the transit that is smaller than the total of all the flows. Because of these fraud possibilities, in traditional QoS reservation architectures the downstream network polices at each border. The policer checks that the actual sent data rate of each flow is within the signalled reservation.

Reservation signalling could be authenticated end to end, but this wouldn't prevent the aggregation cheat just described. For this reason, and to avoid the need for a global PKI, signalling integrity is typically only protected on a hop-by-hop basis [[RFC2747](#)].

A variant of the above cheat is where a router in an honest downstream network denies admission to a new reservation, but a cheating upstream network still admits the flow. For instance, the networks may be using Diffserv internally, but Intserv admission control at their borders [[RFC2998](#)]. The cheat would only work if they were using bulk Diffserv traffic policing at their borders, perhaps to avoid the cost/complexity of Intserv border policing. As far as the cheating upstream network is concerned, it gets the revenue from the reservation, but it doesn't have to pay any downstream wholesale charges and the congestion is in someone else's network. The cheating network may calculate that most of the flows affected by congestion in the downstream network aren't likely to be its own. It may also calculate that the downstream router has been

An ingress and egress gateway (Ingr G/W and Egr G/W in Figure 1)

connect the interior Diffserv region to the edge access networks where routers (not shown) use per-flow reservation processing. Within the Diffserv region are three interior domains, A, B and C, as well as the inward facing interfaces of the ingress and egress gateways. An ingress and egress border router (BR) is shown interconnecting each interior domain with the next. There may be other interior routers (not shown) within each interior domain.

In two paragraphs we now briefly recap how pre-congestion notification is intended to be used to control flow admission to a large Diffserv region. The first paragraph describes data plane functions and the second describes signalling in the control plane. We omit many details from [\[CL-deploy\]](#) including behaviour during routing changes. For brevity here we assume other flows are already in progress across a path through the Diffserv region before a new one arrives, but how bootstrap works is described in [Section 4.3.2](#).

Figure 1 shows a single simplex reserved flow from the sending (Sx) end host to the receiving (Rx) end host. The ingress gateway polices incoming traffic within its admitted reservation and remarks it to turn on an ECN-capable codepoint [\[RFC3168\]](#) and the controlled load (CL) Diffserv codepoint. Together, these codepoints define which traffic is entitled to the enhanced scheduling of the CL behaviour aggregate on routers within the Diffserv region. The CL PHB of interior routers consists of a scheduling behaviour and a new ECN marking behaviour that we call 'pre-congestion notification' [\[PCN\]](#). The CL PHB simply re-uses the definition of expedited forwarding (EF) [\[RFC3246\]](#) for its scheduling behaviour. But it incorporates a new ECN marking behaviour, which sets the ECN field of an increasing number of CL packets to the admission marked (AM) codepoint as they approach a threshold rate that is lower than the line rate. The use of virtual queues ensures real queues have hardly built up any congestion delay. The level of marking detected at the egress of the Diffserv region is then used by the signalling system in order to determine admission control as follows.

The end-to-end QoS signalling (e.g. RSVP) for a new reservation takes one giant hop from ingress to egress gateway, because interior routers within the Diffserv region are configured to ignore RSVP. The egress gateway holds flow state because it takes part in the end-to-end reservation. So it can classify all packets by flow and it can identify all flows that have the same previous RSVP hop (a CL-region-aggregate). For each CL-region-aggregate of flows in progress, the egress gateway maintains a per-packet moving average of the fraction of pre-congestion-marked traffic. Once an RSVP PATH message for a new reservation has hopped across the Diffserv region and reached the destination, an RSVP RESV message is returned. As the RESV message passes, the egress gateway piggy-backs the relevant

pre-congestion level onto it [[RSVP-ECN](#)]. Again, interior routers ignore the RSVP message, but the ingress gateway strips off the pre-congestion level. If the pre-congestion level is above a threshold, the ingress gateway denies admission to the new reservation, otherwise it returns the original RESV signal back towards the data sender.

Once a reservation is admitted, its traffic will always receive low delay service for the duration of the reservation. This is because ingress gateways ensure that traffic not under a reservation cannot pass into the Diffserv region with the CL DSCP set. So non-reserved traffic will always be treated with a lower priority PHB at each interior router. And even if some disaster re-routes traffic after it has been admitted, if the traffic through any resource tips over a fail-safe threshold, pre-congestion notification will trigger flow-pre-emption to very quickly bring every router within the whole Diffserv region back below its operating point.

The whole admission control system just described deliberately confines per-flow processing to the access edges of the network, where it will not limit the system's scalability. But ideally we want to extend this approach to multiple networks, to take even more advantage of its scaling potential. We would still need per-flow processing at the access edges of each network, but not at the high speed interfaces where they interconnect. Even though such an admission control system would work technically, it would gain us no scaling advantage if each network also wanted to police the rate of each admitted flow for itself---border routers would still have to do complex packet operations per-flow anyway, given they don't trust upstream networks to do their policing for them.

This memo describes how to emulate per-flow rate policing using bulk mechanisms at border routers, so the full scalability potential of pre-congestion notification is not limited by the need for per-flow policing mechanisms at borders, which would make borders the most cost-critical pinch-points. Then we can achieve the long sought-for vision of secure Internet-wide bandwidth reservations without needing per-flow processing at all in core and border routers---where scalability is most critical.

[4. Re-ECN Protocol for an RSVP \(or similar\) Transport](#)

[4.1. Protocol Overview](#)

First we need to recap the way routers accumulate congestion marking along a path. Each ECN-capable router marks some packets with CE, the marking probability increasing with the length of the queue at

its egress link. The only difference with pre-congestion marking [[PCN](#)] is that marking is based on the length of a virtual queue, so that the real queue occupancy can remain very low. We will use the terms congestion and pre-congestion interchangeably in the following unless it is important to distinguish between them.

With multiple ECN-capable routers on a path, the ECN field accumulates the fraction of CE marking that each router adds. The combined effect of the packet marking of all the routers along the path signals congestion of the whole path to the receiver. So, for example, if one router early in a path is marking 1% of packets and another later in a path is marking 2%, flows that pass through both routers will experience approximately 3% marking.

The packets crossing an inter-domain trust boundary within the Diffserv region will all have come from different ingress gateways and will all be destined for different egress gateways. We will show that the key to policing against theft of service is for a border router to be able to directly measure the congestion that is about to be caused by the traffic it forwards. That is, it can measure locally the congestion on each of the downstream paths between itself and the egress gateways that its traffic is destined for.

With the original ECN protocol, if CE markings crossing the border had been counted over a period, they would have represented the accumulated upstream congestion that had already been experienced by those packets. The general idea of re-ECN is for the ingress gateway to continuously encode path congestion into the IP header where, in this case, 'path' means from ingress to egress gateway. Then at any point on that path (e.g. between domains A & B in Figure 2 below), IP headers can be monitored to subtract upstream congestion from expected path congestion in order to give the expected downstream congestion still to be experienced until the egress gateway.

Importantly, it turns out that there is no need to monitor downstream congestion on a per-flow basis. We will show that accounting for it in bulk across all flows will be sufficient.

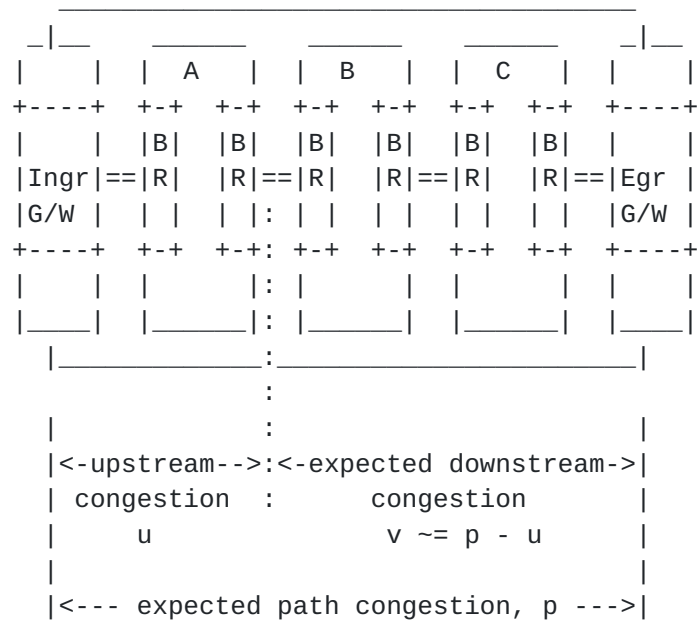


Figure 2: Re-ECN concept

4.2. Re-ECN Abstracted Network Layer Wire Protocol (IPv4 or v6)

In this section we define the names of the various codepoints of the re-ECN protocol when used with pre-congestion notification, deferring description of their semantics to the following sections. But first we recap the re-ECN wire protocol proposed in [Re-TCP].

4.2.1. Re-ECN Recap

Re-ECN uses the two bit ECN field broadly as in RFC3168 [RFC3168]. It also uses a new re-ECN extension (RE) flag. The actual position of the RE flag is different between IPv4 & v6 headers so we will use an abstraction of the IPv4 and v6 wire protocols by just calling it the RE flag. [Re-TCP] proposes using bit 48 (currently unused) in the IPv4 header for the RE flag, while for IPv6 it proposes an ECN extension header.

Unlike the ECN field, the RE flag is intended to be set by the sender and remain unchanged along the path, although it can be read by network elements that understand the re-ECN protocol. In the scenario used in this memo, the ingress gateway acts as a proxy for the sender, setting the RE flag as permitted in the specification of re-ECN.

Note that general-purpose routers do not have to read the RE flag, only special policing elements at borders do. And no general-purpose routers have to change the RE flag, although the ingress and egress

gateways do because in the edge-to-edge deployment model we are using, they act as proxies for the endpoints. Therefore the RE flag does not even have to be visible to interior routers. So the RE flag has no implications on protocols like MPLS. Congested label switching routers (LSRs) would have to be able to notify their congestion with an ECN/PCN codepoint in the MPLS shim [ECN-MPLS], but like any interior IP router, they can be oblivious to the RE flag, which need only be read by border policing functions.

Although the RE flag is a separate, single bit field, it can be read as an extension to the two-bit ECN field; the three concatenated bits in what we will call the extended ECN field (EECN) make eight codepoints available. When the RE flag setting is "don't care", we use the RFC3168 names of the ECN codepoints, but [Re-TCP] proposes the following six codepoint names for when there is a need to be more specific.

ECN field	RFC3168 codepoint	RE flag	Extended ECN codepoint	Re-ECN meaning
00	Not-ECT	0	Not-RECT	Not re-ECN-capable transport
00	Not-ECT	1	FNE	Feedback not established
01	ECT(1)	0	Re-Echo	Re-echoed congestion and RECT
01	ECT(1)	1	RECT	Re-ECN capable transport
10	ECT(0)	0	---	Legacy ECN use only
10	ECT(0)	1	--CU--	Currently unused
11	CE	0	CE(0)	Congestion experienced with Re-Echo
11	CE	1	CE(-1)	Congestion experienced

Table 1: Re-cap of Default Extended ECN Codepoints Proposed for Re-ECN

4.2.2. Re-ECN Combined with Pre-Congestion Notification (re-PCN)

As permitted by the ECN specification [RFC3168], a proposal is currently being advanced in the IETF to define different semantics for how routers might mark the ECN field of certain packets. The

idea is to be able to notify congestion when the router's load approaches a logical limit, rather than the physical limit of the line. This new marking is called pre-congestion notification [[PCN](#)] and we will use the term PCN-enabled router for a router that can apply pre-congestion notification marking to the ECN fields of packets.

[RFC3168] recommends that a packet's Diffserv codepoint should determine which type of ECN marking it receives. A Diffserv per-hop behaviour (PHB) can specify that routers should apply pre-congestion notification marking to PCN-capable packets. We will call this a PCN-enhanced PHB. A PCN-capable packet must meet two conditions, it must carry a DSCP that maps to a PCN-enhanced PHB and it must carry an ECN field that turns on PCN marking.

As an example, the controlled load (CL) PHB might specify expedited forwarding as its scheduling behaviour and PCN marking as its congestion marking behaviour. Then we would say the CL PHB is a PCN-enhanced PHB, and that packets with a DSCP that maps to the CL PHB and with ECN turned on are PCN-capable packets.

[PCN] actually proposes that two logical limits should be used for pre-congestion notification, with the higher limit as a back-stop for dealing with anomalous events. It envisages PCN will be used to admission control inelastic real-time traffic, so marking at the lower limit will trigger admission control, while at the higher limit it will trigger flow pre-emption.

Because it needs two types of congestion marking, PCN seems to need five states: Not-ECT, ECT (ECN-capable transport), the ECN Nonce, Admission Marking (AM) and Flow Pre-emption Marking (PM). [[PCN](#)] proposes various alternative encodings of the ECN field, attempting various compromises to fit these five states into the four available ECN codepoints.

One of the five states to make room for is the ECN Nonce [[RFC3540](#)], but the capability we describe in this memo supersedes any need for the Nonce. The ECN Nonce is an elegant scheme, but it only allows a sending node (or its proxy) to detect suppression of congestion marking in the feedback loop. Thus the Nonce requires the sender or its proxy to be trusted to respond correctly to congestion. But this is precisely the main cheat we want to protect against (as well as many others).

One of the compromise protocol encodings that [[PCN](#)] explores ("Alternative 5") leaves out support for the ECN Nonce. Therefore we use that one. This encoding of PCN markings is shown on the left of Table 2. Note that these codepoints of the ECN field only take on

the semantics of pre-congestion notification if they are combined with a Diffserv codepoint that the operator has configured to cause PCN marking, by mapping it to a PCN-enhanced PHB.

For the rest of this memo, we will not distinguish between Admission Marking and Pre-emption Marking unless we need to be specific. We will call both "congestion marking". With the above encoding, congestion marking can be read to mean any packet with the left-most bit of the ECN field set.

The re-ECN protocol can be used to control misbehaving sources whether congestion is with respect to a logical threshold (PCN) or the physical line rate (ECN). In either case the RE flag can be used to create an extended ECN field. For PCN-capable packets, the 8 possible encodings of this 3-bit extended ECN (EECN) field are defined on the right of Table 2 below. The purposes of these different codepoints will be introduced in subsequent sections.

ECN field	PCN codepoint (Alternative 5)	RE flag	Extended ECN codepoint	Re-ECN meaning
00	Not-ECT	0	Not-RECT	Not re-ECN-capable transport
00	Not-ECT	1	FNE	Feedback not established
01	ECT(1)	0	Re-Echo	Re-echoed congestion and RECT
01	ECT(1)	1	RECT	Re-ECN capable transport
10	AM	0	AM(0)	Admission Marking with Re-Echo
10	AM	1	AM(-1)	Admission Marking
11	PM	0	PM(0)	Pre-emption Marking with Re-Echo
11	PM	1	PM(-1)	Pre-emption Marking

Table 2: Extended ECN Codepoints if the Diffserv codepoint uses Pre-congestion Notification (PCN)

4.3. Protocol Operation

4.3.1. Protocol Operation for an Established Flow

The re-ECN protocol involves a simple tweak to the action of the gateway at the ingress edge of the CL region. In the deployment model just described [[CL-deploy](#)], for each active traffic aggregate across the CL region (CL-region-aggregate) the ingress gateway will hold a fairly recent Congestion-Level-Estimate that the egress gateway will have fed back to it, piggybacked on the signalling that sets up each flow. For instance, one aggregate might have been experiencing 3% pre-congestion (that is, congestion marked octets whether Admission Marked or Pre-emption Marked). In this case, the ingress gateway MUST clear the RE flag to "0" for the same percentage of octets of CL-packets (3%) and set it to "1" in the rest (97%). [Appendix A.1](#) gives a simple pseudo-code algorithm that the ingress gateway may use to do this.

The RE flag is set and cleared this way round for incremental deployment reasons (see [[Re-TCP](#)]). To avoid confusion we will use the term 'blanking' (rather than marking) when the RE flag is cleared to "0", so we will talk of the 'RE blanking fraction' as the fraction of octets with the RE flag cleared to "0".

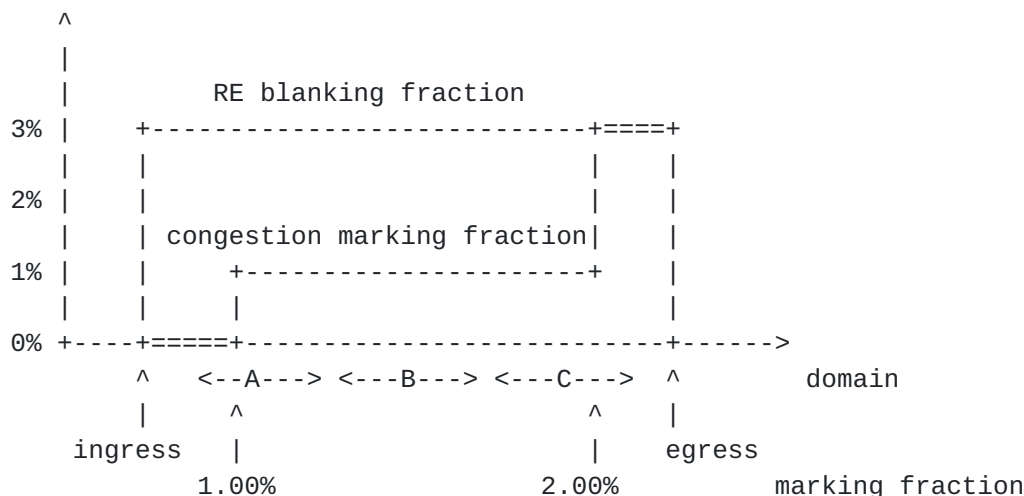


Figure 3: Example Extended ECN codepoint Marking fractions (Imprecise)

Figure 3 illustrates our example. The horizontal axis represents the index of each congestible resource (typically queues) along a path through the Internet. The two superimposed plots show the fraction of each ECN codepoint observed along this path, assuming there are two congested routers somewhere within domains A and C. And Table 3 below shows the downstream pre-congestion measured at various border

observation points along the path. Figure 4 (later) shows the same results of these subtractions, but in graphical form like the above figure. The tabulated figures are actually reasonable approximations derived from more precise formulae given in [Appendix A](#) of [\[Re-TCP\]](#). The RE flag is not changed by interior routers, so it can be seen that it acts as a reference against which the congestion marking fraction can be compared along the path.

Border observation point	Approximate Downstream pre-congestion
ingress -- A	3% - 0% = 3%
A -- B	3% - 1% = 2%
B -- C	3% - 1% = 2%
C -- egress	3% - 3% = 0%

Table 3: Downstream Congestion Measured at Example Observation Points

Note that the ingress determines the RE blanking fraction for each aggregate using the most recent feedback from the relevant egress, arriving with each new reservation, or each refresh. These updates arrive relatively infrequently compared to the speed with which congestion changes. Although this feedback will always be out of date, on average positive errors should cancel out negative over a sufficiently long duration.

In summary, the network adds pre-congestion marking in the forward data path, the egress feeds its level back to the ingress in RSVP (or similar signalling), then the ingress gateway re-echoes it into the forward data path by blanking the RE flag. Hence the name re-ECN. Then at any border within the Diffserv region, the pre-congestion marking that every passing packet will be expected to experience downstream can be measured to be the RE blanking fraction minus the congestion marking fraction.

[4.3.2.](#) Aggregate Bootstrap

When a new reservation PATH message arrives at the egress, if there are currently no flows in progress from the same ingress, there will be no state maintaining the current level of pre-congestion marking for the aggregate. While the reservation signalling continues onward towards the receiving host, the egress gateway returns an RSVP message to the ingress with a flag [\[RSVP-ECN\]](#) asking the ingress to send a specified number of data probes between them. This bootstrap behaviour is all described in the deployment model [\[CL-deploy\]](#).

However, with our new re-ECN scheme, the ingress does not know what

proportion of the data probes should have the RE flag blanked, because it has no estimate yet of pre-congestion for the path across the Diffserv region.

To be conservative, following the guidance for specifying other re-ECN transports in [\[Re-TCP\]](#), the ingress SHOULD set the FNE codepoint of the extended ECN header in all probe packets (Table 2). As per the deployment model, the egress gateway measures the fraction of congestion-marked probe octets and feeds back the resulting pre-congestion level to the ingress, piggy-backed on the returning reservation response (RESV) for the new flow. Probe packets are identifiable by the egress because they have the ingress as the source and the egress as the destination in the IP header.

It may seem inadvisable to expect the FNE codepoint to be set on probes, given legacy firewalls etc. might discard such packets (because this flag had no previous legitimate use). However, in the deployment scenarios envisaged, each domain in the Diffserv region has to be explicitly configured to support the controlled load service. So, before deploying the service, the operator MUST reconfigure such a misbehaving middlebox to allow through packets with the RE flag set.

Note that we have said SHOULD rather than MUST for the FNE setting behaviour of the ingress for probe packets. This entertains the possibility of an ingress implementation having the benefit of other knowledge of the path, which it re-uses for a newly starting aggregate. For instance, it may hold cached information from a recent use of the aggregate that is still sufficiently current to be useful.

It might seem pedantic worrying about these few probe packets, but this behaviour ensures the system is safe, even if the proportion of probe packets becomes large.

[4.3.3.](#) Flow Bootstrap

It might be expected that a new flow within an active aggregate would need no special bootstrap behaviour. If there was an aggregate already in progress between the gateways the new flow was about to use, it would inherit the prevailing RE blanking fraction. And if there were no active aggregate, the bootstrap behaviour for an aggregate would be appropriate and sufficient for the new flow.

However, for a number of reasons, at least the first packet of each new flow SHOULD be set to the FNE codepoint, irrespective of whether it is joining an active aggregate or not. If the first packet is unlikely to be reliably delivered, a number of FNE packets MAY be

sent to increase the probability that at least one is delivered to the egress gateway.

If each flow does not start with an FNE packet, it will be seen later that sanctions may be too strict at the interface before the egress gateway. It will often be possible to apply sanctions at the granularity of aggregates rather than flows, but in an internetworked environment it cannot be guaranteed that aggregates will be identifiable in remote networks. So setting FNE at the start of each flow is a safe strategy. For instance, a remote network may have equal cost multi-path (ECMP) routing enabled, causing different flows between the same gateways to traverse different paths.

After an idle period of more than 1 second, the ingress gateway SHOULD set the EECN field of the next packet it sends to FNE. This allows the design of network policers to be deterministic (see [Re-TCP]).

However, if the ingress gateway can guarantee that the network(s) that will carry the flow to its egress gateway all use a common identifier for the aggregate (e.g. a single MPLS network without ECMP routing), it MAY NOT set FNE when it adds a new flow to an active aggregate. And an FNE packet need only be sent if a whole aggregate has been idle for more than 1 second.

4.3.4. Router Forwarding Behaviour

Adding re-ECN works well without modifying the forwarding behaviour of any routers. However, below, two changes are proposed when forwarding packets with a per-hop-behaviour that requires pre-congestion notification:

Preferential drop: When a router cannot avoid dropping ECN-capable packets, preferential dropping of packets with different extended ECN codepoints SHOULD be implemented between packets within a PHB that uses PCN marking. The drop preference order to use is defined in Table 4. Note that to reduce configuration complexity, Re-Echo and FNE MAY be given the same drop preference, but if feasible, FNE should be dropped in preference to Re-Echo.

ECN field	RE flag	Extended ECN codepoint	Drop Pref	Re-ECN meaning
01	0	Re-Echo	5/4	Re-echoed congestion and RECT
00	1	FNE	4	Feedback not established
01	1	RECT	3	Re-ECN capable transport
10	0	AM(0)	3	Admission Marking with Re-Echo
10	1	AM(-1)	3	Admission Marking
11	0	PM(0)	2	Pre-emption Marking with Re-Echo
11	1	PM(-1)	2	Pre-emption Marking
00	0	Not-RECT	1	Not re-ECN-capable transport

Table 4: Drop Preference of Extended ECN Codepoints (1 = drop 1st)

Given this proposal is being advanced at the same time as PCN itself, we strongly RECOMMEND that preferential drop based on extended ECN codepoint is added to router forwarding at the same time as PCN marking. Preferential dropping can be difficult to implement, but we strongly RECOMMEND this security-related re-ECN improvement where feasible as it is an effective defence against flooding attacks.

Marking vs. Drop: We propose that PCN-routers SHOULD inspect the RE flag as well as the ECN field to decide whether to drop or mark PCN DSCPs. They MUST choose drop if the codepoint of this extended ECN field is Not-RECT. Otherwise they SHOULD mark (unless, of course, buffer space is exhausted).

A PCN-capable router MUST NOT ever congestion mark a packet carrying the Not-RECT codepoint because the transport will only understand drop, not congestion marking. But a PCN-capable router can mark rather than drop an FNE packet, even though its ECN field when looked at in isolation is '00' which appears to be a legacy Not-ECT packet. Therefore, if a packet's RE flag is '1', even if its ECN field is '00', a PCN-enabled router SHOULD use congestion marking. This allows the 'feedback not established' (FNE) codepoint to be used for probe packets, in order to pick up PCN marking when bootstrapping an aggregate.

ECN marking rather than dropping of FNE packets MUST only be deployed in controlled environments, such as that in [\[CL-deploy\]](#), where the presence of an egress node that understands ECN marking is assured. Congestion events might otherwise be ignored if the receiver only understands drop, rather than ECN marking. This is because there is no guarantee that ECN capability has been negotiated if feedback is not established (FNE). Also, [\[Re-TCP\]](#) places the strong condition that a router MUST apply drop rather than marking to FNE packets unless it can guarantee that FNE packets are rate limited either locally or upstream.

[4.3.5.](#) Extensions

If a different signalling system, such as NSIS, were used, but it provided admission control in a similar way, using pre-congestion notification (e.g. with RMD [\[NSIS-RMD\]](#)) we believe re-ECN could be used to protect against misbehaving networks in the same way as proposed above.

[5.](#) Emulating Border Policing with Re-ECN

[5.1.](#) Informal Terminology

In the rest of this memo, where the context makes it clear, we will sometimes loosely use the term 'congestion' rather than using the stricter 'downstream pre-congestion'. Also we will loosely talk of positive or negative flows, meaning flows where the moving average of the downstream pre-congestion metric is persistently positive or negative. The notion of a negative metric arises because it is derived by subtracting one metric from another. Of course actual downstream congestion cannot be negative, only the metric can (whether due to time lags or deliberate malice).

Just as we will loosely talk of positive and negative flows, we will also talk of positive or negative packets, meaning packets that contribute positively or negatively to downstream pre-congestion.

Therefore packets can be considered to have a 'worth' of +1, 0 or -1, which, when multiplied by their size, indicates their contribution to downstream congestion. Packets will usually be sent with a worth of 0. Blanking the RE flag increments the worth of a packet to +1. Congestion marking a packet decrements its worth (whether admission marking or pre-emption marking). Congestion marking a previously blanked packet cancel out the positive and negative worth of each marking (a worth of 0). The FNE codepoint is an exception. It has the same positive worth as a packet with the Re-Echo codepoint. The table below specifies unambiguously the worth of each extended ECN

codepoint. Note the order is different from the previous table to emphasise how congestion marking processes decrement the worth.

ECN field	RE flag	Extended ECN codepoint	Worth	Re-ECN meaning
00	0	Not-RECT	n/a	Not re-ECN-capable transport
01	0	Re-Echo	+1	Re-echoed congestion and RECT
10	0	AM(0)	0	Admission Marking with Re-Echo
11	0	PM(0)	0	Pre-emption Marking with Re-Echo
00	1	FNE	+1	Feedback not established
01	1	RECT	0	Re-ECN capable transport
10	1	AM(-1)	-1	Admission Marking
11	1	PM(-1)	-1	Pre-emption Marking

Table 5: 'Worth' of Extended ECN Codepoints

5.2. Policing Overview

It will be recalled that downstream congestion can be found by subtracting upstream congestion from path congestion. Figure 4 displays the difference between the two plots in Figure 3 to show downstream pre-congestion across the same path through the Internet.

To emulate border policing, the general idea is for each domain to apply penalties to its upstream neighbour in proportion to the amount of downstream pre-congestion that the upstream network sends across the border. That is, the penalties should be in proportion to the height of the plot. Downward arrows in the figure show the resulting pressure for each domain to under-declare downstream pre-congestion in traffic they pass to the next domain, because of the penalties.

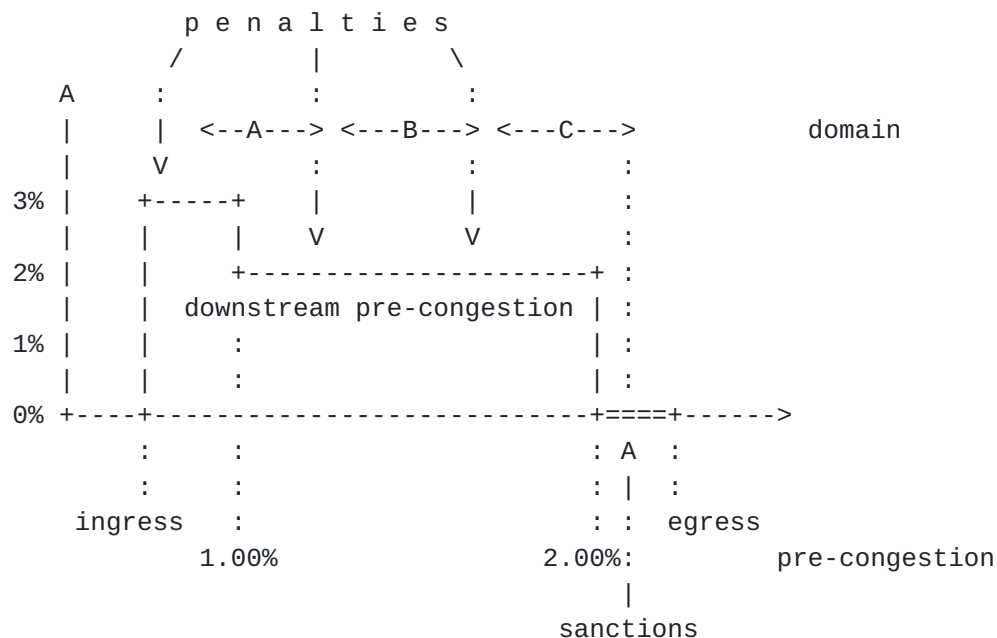


Figure 4: Policing Framework, showing creation of opposing pressures to under-declare and over-declare downstream pre-congestion, using penalties and sanctions

These penalties seem to encourage everyone to understate downstream congestion in order to reduce the penalties they incur. But a balancing pressure is introduced by the last domain, which applies sanctions to flows if downstream congestion goes negative before the egress gateway. The upward arrow at Domain C's border with the egress gateway represents the incentive the sanctions would create to prevent negative traffic. The same upward pressure can be applied at any domain border (arrows not shown).

Any flow that persistently goes negative by the time it leaves a domain must not have been marked correctly in the first place. A domain that discovers such a flow can adopt a range of strategies to protect itself. Which strategy it uses will depend on policy, because it cannot immediately assume malice---there may be an innocent configuration error somewhere in the system.

This memo does not propose to standardise any particular mechanism to detect persistently negative flows, but [Section 5.5](#) does give examples. Note that we have used the term flow, but there will be no need to bury into the transport layer for port numbers; identifiers visible in the network layer will be sufficient (IP address pair, DSCP, protocol ID). The appendix also gives a mechanism to bound the required flow state, preventing state exhaustion attacks.

Of course, some domains may trust other domains to comply with

admission control without applying sanctions or penalties. In these cases, the protocol should still be used but no penalties need be applied. The re-ECN protocol ensures downstream pre-congestion marking is passed on correctly whether or not penalties are applied to it, so the system works just as well with a mixture of some domains trusting each other and others not.

Providers should be free to agree the contractual terms they wish between themselves, so this memo does not propose to standardise how these penalties would be applied. It is sufficient to standardise the re-ECN protocol so the downstream pre-congestion metric is available if providers choose to use it. However, the next section ([Section 5.3](#)) gives some examples of how these penalties might be implemented.

[5.3](#). Pre-requisite Contractual Arrangements

The re-ECN protocol has been chosen to solve the policing problem because it embeds a downstream pre-congestion metric in passing CL traffic that is difficult to lie about and can be measured in bulk. The ability to emulate border policing depends on network operators choosing to use this metric as one of the elements in their contracts with each other.

Already many inter-domain agreements involve a capacity and a usage element. The usage element may be based on volume or various measures of peak demand. We expect that those network operators who choose to use pre-congestion notification for admission control would also be willing to consider using this downstream pre-congestion metric as a usage element in their interconnection contracts for admission controlled (CL) traffic.

Congestion (or pre-congestion) has the dimension of [octet], being the product of volume transferred [octet] and the congestion fraction [dimensionless], which is the fraction of the offered load that the network isn't able to serve (or would rather not serve in the case of pre-congestion). Measuring downstream congestion gives a measure of the volume transferred but modulated by congestion expected downstream. So volume transferred during off-peak periods counts as nearly nothing, while volume transferred at peak times counts very highly. The re-ECN protocol allows one network to measure how much pre-congestion has been 'dumped' into it by another network. And then in turn how much of that pre-congestion it dumped into the next downstream network.

[Section 5.6](#) describes mechanisms for calculating border penalties referring to [Appendix A.2](#) for suggested metering algorithms for downstream congestion at a border router. Conceptually, it could

hardly be simpler. It broadly involves accumulating the volume of packets with the RE flag blanked and the volume of those with congestion marking then subtracting the two.

Once this downstream pre-congestion metric is available, operators are free to choose how they incorporate it into their interconnection contracts [[IXQoS](#)]. Some may include a threshold volume of pre-congestion as a quality measure in their service level agreement, perhaps with a penalty clause if the upstream network exceeds this threshold over, say, a month. Others may agree a set of tiered monthly thresholds, with increasing penalties as each threshold is exceeded. But, it would be just as easy, and more resistant to gaming, to do away with discrete thresholds, and instead make the penalty rise smoothly with the volume of pre-congestion by applying a price to pre-congestion itself. Then the usage element of the interconnection contract would directly relate to the volume of pre-congestion caused by the upstream network.

The direction of penalties and charges relative to the direction of traffic flow is a constant source of confusion. Typically, where capacity charges are concerned, lower tier customer networks pay higher tier provider networks. So money flows from the edges to the middle of the internetwork, towards greater connectivity, irrespective of the flow of data. But we advise that penalties or charges for usage should follow the same direction as the data flow---the direction of control at the network layer. Otherwise a network lays itself open to 'denial of funds' attacks. So, where a tier 2 provider sends data into a tier 3 customer network, we would expect the penalty clauses for sending too much pre-congestion to be against the tier 2 network, even though it is the provider.

It may help to remember that data will be flowing in the other direction too. So the provider network has as much opportunity to levy usage penalties as its customer, and it can set the price or strength of its own penalties higher if it chooses. Usage charges in both directions tend to cancel each other out, which confirms that usage-charging is less to do with revenue raising and more to do with encouraging load control discipline in order to smooth peaks and troughs, improving utilisation and quality.

Further, when operators agree penalties in their interconnection contracts for sending downstream congestion, they should make sure that any level of negative marking only equates to zero penalty. In other words, penalties are always paid in the same direction as the data, and never against the data flow, even if downstream congestion seems to be negative. This is consistent with the definition of physical congestion; when a resource is underutilised, it is not negatively congested. Its congestion is just zero. So, although

short periods of negative marking can be tolerated to correct temporary over-declarations due to lags in the feedback system, persistent downstream negative congestion can have no physical meaning and therefore must signify a problem. The incentive for domains not to tolerate persistently negative traffic depends on this principle that penalties must never be paid against the data flow.

Also note that at the last egress of the Diffserv region, domain C should not agree to pay any penalties to the egress gateway for pre-congestion passed to the egress gateway. Downstream pre-congestion to the egress gateway should have reached zero here. If domain C were to agree to pay for any remaining downstream pre-congestion, it would give the egress gateway an incentive to over-declare pre-congestion feedback and take the resulting profit from domain C.

To focus the discussion, from now on, unless otherwise stated, we will assume a downstream network charges its upstream neighbour in proportion to the pre-congestion it sends (V_b in the notation of [Appendix A.2](#)). Effectively tiered thresholds would be just more coarse-grained approximations of the fine-grained case we choose to examine. If these neighbours had previously agreed that the (fixed) price per octet of pre-congestion would be L , then the bill at the end of the month would simply be the product $L \cdot V_b$, plus any fixed charges they may also have agreed.

We are well aware that the IETF tries to avoid standardising technology that depends on a particular business model. Indeed, this principle is at the heart of all our own work. Our aim here is to make a new metric available that we believe is superior to all existing metrics. Then, our aim is to show that border policing can at least work with the one model we have just outlined. We assume that operators might then experiment with the metric in other models. Of course, operators are free to complement this pre-congestion-based usage element of their charges with traditional capacity charging, and we expect they will.

Also note well that everything we discuss in this memo only concerns interconnection within the Diffserv region. ISPs are free to sell or give away reservations however they want on the retail market. But of course, interconnection charges will have a bearing on that. Indeed, in the present scenario, the ingress gateway effectively sells reservations on one side and buys congestion penalties on the other. As congestion rises, one can imagine the gateway discovering that congestion penalties have risen higher than the (probably fixed) revenue it will earn from selling the next flow reservation. This encourages the gateway to cut its losses by blocking new calls, which is why we believe downstream congestion penalties can emulate per-flow rate policing at borders, as the next section explains.

5.4. Emulation of Per-Flow Rate Policing: Rationale and Limits

The important feature of charging in proportion to congestion volume is that the penalty aggregates and disaggregates correctly along with packet flows. This is because the penalty rises linearly with bit rate (unless congestion is absolutely zero) and linearly with congestion, because it is the product of them both. So if the packets crossing a border belong to a thousand flows, and one of those flows doubles its rate, the ingress gateway forwarding that flow will have to put twice as much congestion marking into the packets of that flow. And this extra congestion marking will add proportionately to the penalties levied at every border the flow crosses in proportion to the amount of pre-congestion remaining on the path.

Effectively, usage charges will continuously flow from ingress gateways to the places generating pre-congestion marking, in proportion to the pre-congestion marking introduced and to the data rates from those gateways.

As importantly, pre-congestion itself rises super-linearly with utilisation of a particular resource. So if someone tries to push another flow into a path that is already signalling enough pre-congestion to warrant admission control, the penalty will be a lot greater than it would have been to add the same flow to a less congested path. This makes the incentive system fairly insensitive to the actual level of pre-congestion for triggering admission control that each ingress chooses. The deterrent against exceeding whatever threshold is chosen rises very quickly with a small amount of cheating.

These are the properties that allow re-ECN to emulate per-flow border policing of both rate and admission control. It is not a perfect emulation of per-flow border policing, but we claim it is sufficient to at least ensure the cost to others of a cheat is borne by the cheater, because the penalties are at least proportionate to the level of the cheat. If an edge network operator is selling reservations at a large profit over the congestion cost, these pre-congestion penalties will not be sufficient to ensure networks in the middle get a share of those profits, but at least they can cover their costs.

We will now explain with an example. When a whole inter-network is operating at normal (typically very low) congestion, the pre-congestion marking from virtual queues will be a little higher than if the real queues had been used---still low, but more noticeable. But low congestion levels do not imply that usage /charges/ must also be low. Usage charges will depend on the /price/ L as well.

If the metric of the usage element of an interconnection agreement was changed from pure volume to pre-congested volume, one would expect the price of pre-congestion to be arranged so that the total usage charge remained about the same. So, if an average pre-congestion fraction turned out to be $1/1000$, one would expect that the price L (per octet) of pre-congestion would be about 1000 times the previously used (per octet) price for volume. We should add that a switch to pre-congestion is unlikely to exactly maintain the same overall level of usage charges, but this argument will be approximately true, because usage charge will rise to at least the level the market finds necessary to push back against usage.

From the above example it can be seen why a 1000x higher price will make operators become acutely sensitive to the congestion they cause in other networks, which is of course the desired effect; to encourage networks to /control/ the congestion they allow their users to cause to others.

If any network sends even one flow at higher rate, they will immediately have to pay proportionately more usage charges. Because there is no knowledge of reservations within the Diffserv region, no interior router can police whether the rate of each flow is greater than each reservation. So the system doesn't truly emulate rate-policing of each flow. But there is no incentive to pack a higher rate into a reservation, because the charges are directly proportional to rate, irrespective of the reservations.

However, if virtual queues start to fill on any path, even though real queues will still be able to provide low latency service, pre-congestion marking will rise fairly quickly. It may eventually reach the threshold where the ingress gateway would deny admission to new flows. If the ingress gateway cheats and continues to admit new flows, the affected virtual queues will rapidly fill, even though the real queues will still be little worse than they were when admission control should have been invoked. The ingress gateway will have to pay the penalty for such an extremely high pre-congestion level, so the pressure to invoke admission control should become unbearable.

The above mechanisms protect against rational operators. In [Section 5.6.3](#) we discuss how networks can protect themselves from accidental or deliberate misconfiguration in neighbouring networks.

5.5. Sanctioning Dishonest Marking

As CL traffic leaves the last network before the egress gateway (domain C) the RE blanking fraction should match the congestion marking fraction, when averaged over a sufficiently long duration (perhaps ~10s to allow a few rounds of feedback through regular

signalling of new and refreshed reservations).

To protect itself, domain C should install a monitor at its egress. It aims to detect flows of CL packets that are persistently negative. If flows are positive, domain C need take no action---this simply means an upstream network must be paying more penalties than it needs to. [Appendix A.3](#) gives a suggested algorithm for the monitor, meeting the criteria below.

- o It SHOULD introduce minimal false positives for honest flows;
- o It SHOULD quickly detect and sanction dishonest flows (minimal false negatives);
- o It MUST be invulnerable to state exhaustion attacks from malicious sources. For instance, if the dropper uses flow-state, it should not be possible for a source to send numerous packets, each with a different flow ID, to force the dropper to exhaust its memory capacity;
- o It MUST introduce sufficient loss in goodput so that malicious sources cannot play off losses in the egress dropper against higher allowed throughput. Salvatori [[CLoop_pol](#)] describes this attack, which involves the source understating path congestion then inserting forward error correction (FEC) packets to compensate expected losses.

Note that the monitor operates on flows but with careful design we can avoid per-flow state. This is why we have been careful to ensure that all flows MUST start with a packet marked with the FNE codepoint. If a flow does not start with the FNE codepoint, a monitor is likely to treat it unfavourably. This risk makes it worth setting the FNE codepoint at the start of a flow, even though there is a cost to setting FNE (positive 'worth').

Starting flows with an FNE packet also means that a monitor will be resistant to state exhaustion attacks from other networks, as the monitor can then be designed to never create state unless an FNE packet arrives. And an FNE packet counts positive, so it will cost a lot for a network to send many of them.

Monitor algorithms will often maintain a moving average across flows of the fraction of RE blanked packets. When maintaining an average across flows, a monitor MUST ignore packets with the FNE codepoint set. An ingress gateway sets the FNE codepoint when it does not have the benefit of feedback from the egress. So counting packets with FNE cleared would be likely to make the average unnecessarily positive, providing headroom (or should we say footroom?) for

dishonest (negative) traffic.

If the monitor detects a persistently negative flow, it could drop sufficient negative and neutral packets to force the flow to not be negative. This is the approach taken for the 'egress dropper' in [\[Re-TCP\]](#), but for the scenario in this memo, where everyone would expect everyone else to keep to the protocol, a management alarm SHOULD be raised on detecting persistently negative traffic and any automatic sanctions taken SHOULD be logged. Even if the chosen policy is to take no automatic action, the cause can then be investigated manually.

Then all ingresses cannot understate downstream pre-congestion without their action being logged. So network operators can deal with offending networks at the human level, out of band. As a last resort, perhaps where the ingress gateway address seems to have been spoofed in the signalling, packets can be dropped. Drops could be focused on just sufficient packets in misbehaving flows to remove the negative bias while doing minimal harm.

A future version of this memo may define a control message that could be used to notify an offending ingress gateway (possibly via the egress gateway) that it is sending persistently negative flows. However, we are aware that such messages could be used to test the sensitivity of the detection system, so currently we prefer silent sanctions.

An extreme scenario would be where an ingress gateway (or set of gateways) mounted a DoS attack against another network. If their traffic caused sufficient congestion to lead to drop but they understated path congestion to avoid penalties for causing high congestion, the preferential drop recommendations in [Section 4.3.4](#) would at least ensure that these flows would always be dropped before honest flows..

[5.6.](#) Border Mechanisms

[5.6.1.](#) Border Accounting Mechanisms

One of the main design goals of re-ECN was for border security mechanisms to be as simple as possible, otherwise they would become the pinch-points that limit scalability of the whole internetwork. As the title of this memo suggests, we want to avoid per-flow processing at borders. We also want to keep to passive mechanisms that can monitor traffic in parallel to forwarding, rather than having to filter traffic inline---in series with forwarding. As data rates continue to rise, we suspect that all-optical interconnection between networks will soon be a requirement. So we want to avoid any

new need for buffering (even though border filtering is current practice for other reasons, we don't want to make it even less likely that we will ever get rid of it).

So far, we have been able to keep the border mechanisms simple, despite having had to harden them against some subtle attacks on the re-ECN design. The mechanisms are still passive and avoid per-flow processing, although we do use filtering as a fail-safe to temporarily shield against extreme events in other networks, such as accidental misconfigurations ([Section 5.6.3](#)).

The basic accounting mechanism at each border interface simply involves accumulating the volume of packets with positive worth (Re-Echo and FNE), and subtracting the volume of those with negative worth: AM(-1) and PM(-1). Even though this mechanism takes no regard of flows, over an accounting period (say a month) this subtraction will account for the downstream congestion caused by all the flows traversing the interface, wherever they come from, and wherever they go to. The two networks can agree to use this metric however they wish to determine some congestion-related penalty against the upstream network (see [Section 5.3](#) for examples). Although the algorithm could hardly be simpler, it is spelled out using pseudo-code in [Appendix A.2.1](#).

Various attempts to subvert the re-ECN design have been made. In all cases their root cause is persistently negative flows. But, after describing these attacks we will show that we don't actually have to get rid of all persistently negative flows in order to thwart the attacks.

In honest flows, downstream congestion is measured as positive minus negative volume. So if all flows are honest (i.e. not persistently negative), adding all positive volume and all negative volume without regard to flows will give an aggregate measure of downstream congestion. But such simple aggregation is only possible if no flows are persistently negative. Unless persistently negative flows are completely removed, they will reduce the aggregate measure of congestion. The aggregate may still be positive overall, but not as positive as it would have been had the negative flows been removed.

In [Section 5.5](#) we discussed how to sanction traffic to remove, or at least to identify, persistently negative flows. But, even if the sanction for negative traffic is to discard it, unless it is discarded at the exact point it goes negative, it will wrongly subtract from aggregate downstream congestion, at least at any borders it crosses after it has gone negative but before it is discarded.

We rely on sanctions to deter dishonest understatement of congestion. But even the ultimate sanction of discard can only be effective if the sender is bothered about the data getting through to its destination. A number of attacks have been identified where a sender gains from sending dummy traffic or it can attack someone or something using dummy traffic even though it isn't communicating any information to anyone:

- o A network can simply create its own dummy traffic to congest another network, perhaps causing it to lose business at no cost to the attacking network. This is a form of denial of service perpetrated by one network on another. The preferential drop measures in [Section 4.3.4](#) provide crude protection against such attacks, but we are not overly worried about more accurate prevention measures, because it is already possible for networks to DoS other networks on the general Internet, but they generally don't because of the grave consequences of being found out. We are only concerned if re-ECN increases the motivation for such an attack, as in the next example.
- o A network can just generate negative traffic and send it over its border with a neighbour to reduce the overall penalties that it should pay to that neighbour. It could even initialise the TTL so it expired shortly after entering the neighbouring network, reducing the chance of detection further downstream. This attack need not be motivated by a desire to deny service and indeed need not cause denial of service. A network's main motivator would most likely be to reduce the penalties it pays to a neighbour. But, the prospect of financial gain might tempt the network into mounting a DoS attack on the other network as well, given the gain would offset some of the risk of being detected.

Note that we have not included DoS by Internet hosts in the above list of attacks, because we have restricted ourselves to a scenario with edge-to-edge admission control across a Diffserv region. In this case, the edge ingress gateways insulate the Diffserv region from DoS by Internet hosts. Re-ECN resists more general DoS attacks, but this is discussed in [\[Re-TCP\]](#).

The first step towards a solution to all these problems with negative flows is to be able to estimate the contribution they make to downstream congestion at a border and to correct the measure accordingly. Although ideally we want to remove negative flows themselves, perhaps surprisingly, the most effective first step is to cancel out the polluting effect negative flows have on the measure of downstream congestion at a border. It is more important to get an unbiased estimate of their effect, than to try to remove them all. A suggested algorithm to give an unbiased estimate of the contribution

from negative flows to the downstream congestion measure is given in [Appendix A.2.2](#).

Although making an accurate assessment of the contribution from negative flows may not be easy, just the single step of neutralising their polluting effect on congestion metrics removes all the gains networks could otherwise make from mounting dummy traffic attacks on each other. This puts all networks on the same side (only with respect to negative flows of course), rather than being pitched against each other. The network where this flow goes negative as well as all the networks downstream lose out from not being reimbursed for any congestion this flow causes. So they all have an interest in getting rid of these negative flows. Networks forwarding a flow before it goes negative aren't strictly on the same side, but they are disinterested bystanders---they don't care that the flow goes negative downstream, but at least they can't actively gain from making it go negative. The problem becomes localised so that once a flow goes negative, all the networks from where it happens and beyond downstream each have a small problem, each can detect it has a problem and each can get rid of the problem if it chooses to. But negative flows can no longer be used for any new attacks.

Once an unbiased estimate of the effect of negative flows can be made, the problem reduces to detecting and preferably removing flows that have gone negative as soon as possible. But importantly, complete eradication of negative flows is no longer critical---best endeavours will be sufficient.

Note that the guiding principle behind all the above discussion is that any gain from subverting the protocol should be precisely neutralised, rather than punished. If a gain is punished to a greater extent than is sufficient to neutralise it, it will most likely open up a new vulnerability, where the amplifying effect of the punishment mechanism can be turned on others.

For instance, if possible, flows should be removed as soon as they go negative, but we do NOT RECOMMEND any attempts to discard such flows further upstream while they are still positive. Such over-zealous push-back is unnecessary and potentially dangerous. These flows have paid their 'fare' up to the point they go negative, so there is no harm in delivering them that far. If someone downstream asks for a flow to be dropped as near to the source as possible, because they say it is going to become negative later, an upstream node cannot test the truth of this assertion. Rather than have to authenticate such messages, re-ECN has been designed so that flows can be dropped solely based on locally measurable evidence. A message hinting that a flow should be watched closely to test for negativity is fine. But not a message that claims that a positive flow will go negative

later, so it should be dropped. .

5.6.2. Competitive Routing

With the above penalty system, each domain seems to have a perverse incentive to fake pre-congestion. For instance domain B profits from the difference between penalties it receives at its ingress (its revenue) and those it pays at its egress (its cost). So if B overstates internal pre-congestion it seems to increase its profit. However, we can assume that domain A could bypass B, routing through other domains to reach the egress. So the competitive discipline of least-cost routing can ensure that any domain tempted to fake pre-congestion for profit risks losing /all/ its incoming traffic. The least congested route would eventually be able to win this competitive game, only as long as it didn't declare more fake pre-congestion than the next most competitive route.

This memo does not need to standardise any particular mechanism for routing based on re-ECN. Goldenberg et al [[Smart_rtg](#)] refers to various commercial products and presents its own algorithms for moving traffic between multi-homed routes based on usage charges. None of these systems require any changes to standards protocols because the choice between the available border gateway protocol (BGP) routes is based on a combination of local knowledge of the charging regime and local measurement of traffic levels. If, as we propose, charges or penalties were based on the level of re-ECN measured in passing traffic, a similar optimisation could be achieved without requiring any changes to standard routing protocols.

We must be clear that applying pre-congestion-based routing to this admission control system remains an open research issue. Traffic engineering based on congestion requires careful damping to avoid oscillations, and should not be attempted without adult supervision :) Mortier & Pratt [[ECN-BGP](#)] have analysed traffic engineering based on congestion. But without the benefit of re-ECN, they had to add a path attribute to BGP to advertise a route's downstream congestion (actually they proposed that BGP should advertise the charge for congestion, which we believe wrongly embeds an assumption into BGP that the only thing to do with congestion is charge for it).

5.6.3. Fail-safes

The mechanisms described so far create incentives for rational operators to behave. That is, one operator aims to make another behave responsibly by applying penalties and expects a rational response (i.e. one that trades off costs against benefits). It is usually reasonable to assume that other network operators will behave rationally (policy routing can avoid those that might not). But this

approach does not protect against the misconfigurations and accidents of other operators.

Therefore, we propose the following two mechanisms at a network's borders to provide "defence in depth". Both are similar:

Highly positive flows: A small sample of positive packets should be picked randomly as they cross a border interface. Then subsequent packets matching the same source and destination address and DSCP should be monitored. If the fraction of positive marking is well above a threshold (to be determined by operational practice), a management alarm SHOULD be raised, and the flow MAY be automatically subject to focused drop.

Persistently negative flows: A small sample of congestion marked packets should be picked randomly as they cross a border interface. Then subsequent packets matching the same source and destination address and DSCP should be monitored. If the RE blanking fraction minus the congestion marking fraction is persistently negative, a management alarm SHOULD be raised, and the flow MAY be automatically subject to focused drop.

Both these mechanisms rely on the fact that highly positive (or negative) flows will appear more quickly in the sample by selecting randomly solely from positive (or negative) packets.

Note that there is no assumption that /users/ behave rationally. The system is protected from the vagaries of irrational user behaviour by the ingress gateways, which transform internal penalties into a deterministic, admission control mechanism that prevents users from misbehaving, by directly engineered means.

6. Analysis

The domains in Figure 1 are not expected to be completely malicious towards each other. After all, we can assume that they are all co-operating to provide an internetworking service to the benefit of each of them and their customers. Otherwise their routing policies would not interconnect them in the first place. However, we assume that they are also competitors of each other. So a network may try to contravene our proposed protocol if it would gain or make a competitor lose, or both, but only if it can do so without being caught. Therefore we do not have to consider every possible random attack one network could launch on the traffic of another, given anyway one network can always drop or corrupt packets that it forwards on behalf of another.

Therefore, we only consider new opportunities for /gainful/ attack that our proposal introduces. But to a certain extent we can also rely on the in depth defences we have described ([Section 5.6.3](#)) intended to mitigate the potential impact if one network accidentally misconfiguring the workings of this protocol.

The ingress and egress gateways are shown in the most generic arrangement possible in Figure 1, without any surrounding network. This allows us to consider more specific cases where these gateways and a neighbouring network are operated by the same player. As well as cases where the same player operates neighbouring networks, we will also consider cases where the two gateways collude as one player and where the sender and receiver collude as one. Collusion of other sets of domains is less likely, but we will consider such cases. In the general case, we will assume none of the nine trust domains across the figure fully trust any of the others.

As we only propose to change routers within the Diffserv region, we assume the operators of networks outside the region will be doing per-flow policing. That is, we assume the networks outside the Diffserv region and the gateways around its edges can protect themselves. So given we are proposing to remove flow policing from some networks, our primary concern must be to protect networks that don't do per-flow policing (the potential 'victims') from those that do (the 'enemy'). The ingress and egress gateways are the only way the outer enemy can get at the middle victim, so we can consider the gateways as the representatives of the enemy as far as domains A, B and C are concerned. We will call this trust scenario 'edges against middles'.

Earlier in this memo, we outlined the classic border rate policing problem ([Section 3](#)). It will now be useful to reiterate the motivations that are the root cause of the problem. The more reservations a gateway can allow, the more revenue it receives. The middle networks want the edges to comply with the admission control protocol when they become so congested that their service to others might suffer. The middle networks also want to ensure the edges cannot steal more service from them than they are entitled to.

In the context of this 'edges against middles' scenario, the re-ECN protocol has two main effects:

- o The more pre-congestion there is on a path across the Diffserv region, the higher the ingress gateway must declare downstream pre-congestion.
- o If the ingress gateway does not declare downstream pre-congestion high enough on average, it will 'hit the ground before the

runway', going negative and triggering sanctions, either directly against the traffic or against the ingress gateway at a management level

An executive summary of our security analysis can be stated in three parts, distinguished by the type of collusion considered.

Neighbour-only Middle-Middle Collusion: Here there is no collusion or collusion is limited to neighbours in the feedback loop. In other words, two neighbouring networks can be assumed to act as one. Or the egress gateway might collude with domain C. Or the ingress gateway might collude with domain A. Or ingress and egress gateways might collude with each other.

In these cases where only neighbours in the feedback loop collude, we conclude that all parties have a positive incentive to declare downstream pre-congestion truthfully, and the ingress gateway has a positive incentive to invoke admission control when congestion rises above the admission threshold in any network in the region (including its own). No party has an incentive to send more traffic than declared in reservation signalling (even though only the gateways read this signalling). In short, no party can gain at the expense of another.

Non-neighbour Middle-Middle Collusion: In the case of other forms of collusion between middle networks (e.g. between domain A and C) it would be possible for say A & C to create a tunnel between themselves so that A would gain at the expense of B. But C would then lose the gain that A had made. Therefore the value to A & C of colluding to mount this attack seems questionable. It is made more questionable, because the attack can be statistically detected by B using the second 'defence in depth' mechanism mentioned already. Note that C can defend itself from being attacked through a tunnel by treating the tunnel end point as a direct link to a neighbouring network (e.g. as if A were a neighbour of C, via the tunnel), which falls back to the safety of the neighbour-only scenario.

Middle-Edge Collusion: Collusion between networks or gateways within the Diffserv region and networks or users outside the region has not yet been fully analysed. The presence of full per-flow policing at the ingress gateway seems to make this a less likely source of a successful attack.

{ToDo: Due to lack of time, the full write up of the security analysis is deferred to the next version of this memo.}

Finally, it is well known that the best person to analyse the

security of a system is not the designer. Therefore, our confident claims must be hedged with doubt until others with perhaps a greater incentive to break it have mounted a full analysis.

7. Incremental Deployment

We believe ECN has so far not been widely deployed because it requires widespread end system and network deployment just to achieve a marginal improvement in performance. The ability to offer a new service (admission control) would be a much stronger driver for ECN deployment.

As stated in the introduction, the aim of this memo is to "build in security from the start" when admission control is based on pre-congestion notification. However, the proposal has been designed so that security can be added some time after first deployment. Given admission control based on pre-congestion notification requires few changes to standards, it should be deployable fairly soon. However, re-ECN requires a change to IP, which may take a little longer.

We expect that initial deployments of PCN-based admission control will be confined to single networks, or to clubs of networks that trust each other. The proposal in this memo will only become relevant once networks with conflicting interests wish to interconnect their admission controlled services, but without the scalability constraints of per-flow border policing. It will not be possible to use re-ECN, even in a controlled environment between consenting operators, unless it is standardised into IP. Given the IPv4 header has limited space for further changes, current IESG policy [{ToDo: ref?}] is not to allow experimental use of codepoints in the IPv4 header, as whenever an experiment isn't taken up, the space it used tends to be impossible to reclaim.

If PCN-based admission control is deployed before re-ECN is standardised into IP, wherever a networks (or club of networks) connects to another network (or club of networks) with conflicting interests, they will place a gateway between the two regions that does per-flow rate policing and admission control. If re-ECN is eventually standardised into IP, it will be possible for these separate regions to upgrade all their gateways to use re-ECN before removing the per-flow policing gateways between them. Given the edge-to-edge deployment model of PCN-based admission control, it is reasonable to imagine this incremental deployment model without needing to cater for partial deployment of re-ECN in just some of the gateways around one Diffserv region.

Only the edge gateways around a Diffserv region have to be upgraded

to add re-ECN support, not interior routers. It is also necessary to add the mechanisms that use re-ECN to secure a network against misbehaving gateways and networks. Specifically, these are the border mechanisms ([Section 5.6](#)) and the mechanisms to sanction dishonest marking ([Section 5.5](#)).

We also RECOMMEND adding improvements to forwarding on interior routers ([Section 4.3.4](#)). But the system works whether all, some or none are upgraded, so interior routers may be upgraded in a piecemeal fashion at any time.

8. Design Choices and Rationale

The primary insight of this work is that downstream congestion is the metric that would be most useful to control an internetwork, and particularly to police how one network responds to the congestion it causes in a remote network. This is the problem that has previously made it so hard to provide scalable admission control.

The case for using re-feedback (a generalisation of re-ECN) to police congestion response and provide QoS is made in [[Re-fb](#)]. Essentially, the insight is that congestion is a factor that crosses layers from the physical upwards. Therefore re-feedback polices congestion where it emerges from a physical interface between networks. This is achieved by bringing the congestion information to the interface, rather than examining packet addressing where there is congestion. Then congestion crossing the physical interface at a border can be policed at the interface, rather than policing the congestion on packets that claim to come from an address (which may be spoofed). Also, re-feedback works in the network layer independently of other layers---despite its name re-feedback does not actually require feedback. It requires a source to act conservatively before it gets feedback.

On the subject of lack of feedback, the feedback not established (FNE) codepoint is motivated by arguments for a state set-up bit in IP to prevent state exhaustion attacks. This idea was first put forward informally by David Clark and documented by Handley and Greenhalgh in [[Steps DoS](#)]. The idea is that network layer datagrams should signal explicitly when they require state to be created in the network layer or the layer above (e.g. at flow start). Then a node can refuse to create any state unless a datagram declares this intent. We believe the proposed FNE codepoint serves the same purpose as the proposed state-set-up bit, but it has been overloaded with a more specific purpose, using it on more packets than just the first in a flow, but never less (i.e. it is idempotent). In effect the FNE codepoint serves the purpose of a `soft-state set-up

codepoint'.

The re-feedback paper [[Re-fb](#)] also makes the case for converting the economic interpretation of congestion into hard engineering mechanism, which is the basis of the approach used in this memo. The admission control gateways around the Diffserv region use hard engineering, not incentives, to prevent end users from sending more traffic than they have reserved. Incentive-based mechanisms are only used between networks, because they are expected to respond to incentives more rationally than end-users can be expected to. However, even then, a network can use fail-safes to protect itself from excessively unusual behaviour by neighbouring networks, whether due to an accidental misconfiguration or malicious intent.

The guiding principle behind the incentive-based approach used between networks is that any gain from subverting the protocol should be precisely neutralised, rather than punished. If a gain is punished to a greater extent than is sufficient to neutralise it, it will most likely open up a new vulnerability, where the amplifying effect of the punishment mechanism can be turned on others.

The re-feedback paper also makes the case against the use of congestion charging to police congestion if it is based on classic feedback (where only upstream congestion is visible to network elements). It argues this would open up receiving networks to 'denial of funds' attacks and would require end users to accept dynamic pricing (which few would).

Re-ECN has been deliberately designed to simplify policing at the borders between networks. These trust boundaries are the critical pinch-points that will limit the scalability of the whole internetwork unless the overall design minimises the complexity of security functions at these borders. The border mechanisms described in this memo run passively in parallel to data forwarding and they do not require per-flow processing.

9. Security Considerations

This whole memo concerns the security of a scalable admission control system. In particular the analysis section. Below some specific security issues are mentioned that did not belong elsewhere or which comment on the overall robustness of the security provided by the design.

Firstly, we must repeat the statement of applicability in the analysis: that we only consider new opportunities for /gainful/ attack that our proposal introduces, particularly if the attacker can

avoid being identified. Despite only involving a few bits, there is sufficient complexity in the whole system that there are probably numerous possibilities for other attacks. However, as far as we are aware, none reap any benefit to the attacker. For instance, it would be possible for a downstream network to remove the congestion markings introduced by an upstream network, but it would only lose out on the penalties it could apply to a downstream network.

When one network forwards a neighbouring network's traffic it will always be possible to cause damage by dropping or corrupting it. Therefore we do not believe networks would set their routing policies to interconnect in the first place if they didn't trust the other networks not to arbitrarily damage their traffic.

Having said this, we do want to highlight some of the weaker parts of our argument. We have argued that networks will be dissuaded from faking congestion marking by the possibility that upstream networks will route round them. As we have said, these arguments are based on fairly delicate assumptions and will remain fairly tenuous until proved in practice, particularly close to the egress where less competitive routing is likely.

We should also point out that the approach in this memo was only designed to be robust for admission control. We do not claim the incentives will always be strong enough to force correct flow pre-emption behaviour. This is because a user will tend to perceive much greater loss in value if a flow is pre-empted than if admission is denied at the start. However, in general the incentives for correct flow pre-emption are similar to those for admission control.

Finally, it may seem that the 8 codepoints that have been made available by extending the ECN field with the RE flag have been used rather wastefully. In effect the RE flag has been used as an orthogonal single bit in nearly all cases. The only exception being when the ECN field is cleared to "00". The mapping of the codepoints in an earlier version of this proposal used the codepoint space more efficiently, but the scheme became vulnerable to a network operator focusing its congestion marking to mark more positive than neutral packets in order to reduce its penalties.

With the scheme as now proposed, once the RE flag is set or cleared by the sender or its proxy, it should not be written by the network, only read. So the gateways can detect if any network maliciously alters the RE flag. IPSec AH integrity checking does not cover the IPv4 option flags (they were considered mutable---even the one we propose using for the RE flag that was 'currently unused' when IPSec was defined). But it would be sufficient for a pair of gateways to make random checks on whether the RE flag was the same when it

reached the egress gateway as when it left the ingress. Indeed, if IPSec AH had covered the RE flag, any network intending to alter sufficient RE flags to make a gain would have focused its alterations on packets without authenticating headers (AHs).

No cryptographic algorithms have been harmed in the making of this proposal.

10. IANA Considerations

This memo includes no request to IANA.

11. Conclusions

This memo builds on a promising technique to solve the classic problem of making flow admission control scale to any size network. It involves the use of Diffserv in a deployment model that uses pre-congestion notification feedback to control admission into a network path [[CL-deploy](#)]. However as it stands, that deployment model depends on all network domains trusting each other to comply with the protocols, invoking admission control and flow pre-emption when requested.

We propose that the congestion feedback used in that deployment model should be re-echoed into the forward data path, by making a trivial modification to the ingress gateway. We then explain how the resulting downstream pre-congestion metric in packets can be monitored in bulk at borders to sufficiently emulate flow rate policing.

We claim the result of combining these two approaches is an admission control system that scales to any size network /and/ any number of interconnected networks, even if they all act in their own interests.

This proposal aims to convince its readers to "Design in Security from the start," by building modified ingress gateways from day one, even if border policing is not needed at first. This way, we will not build ourselves tomorrow's legacy problem.

Re-echoing congestion feedback is based on a principled technique called Re-ECN [[Re-TCP](#)], designed to add accountability for causing congestion to the general-purpose IP datagram service. Re-ECN proposes to consume the last completely unused bit in the basic IPv4 header.

12. Acknowledgements

All the following have given helpful comments and some may become co-authors of later drafts: Arnaud Jacquet, Alessandro Salvatori, Steve Rudkin, David Songhurst, John Davey, Ian Self, Anthony Sheppard, Carla Di Cairano-Gilfedder (BT), Mark Handley (who identified the excess canceled packets attack), Stephen Hailes, Adam Greenhalgh (UCL), Francois Le Faucheur, Anna Charny (Cisco), Jozef Babiarz, Kwok-Ho Chan, Corey Alexander (Nortel), David Clark, Bill Lehr, Sharon Gillett, Steve Bauer (MIT) (who publicised various dummy traffic attacks), Sally Floyd (ICIR) and comments from participants in the CFP/CRN inter-provider QoS and broadband working groups.

13. Comments Solicited

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

14. References

14.1. Normative References

- [PCN] Briscoe, B., Eardley, P., Songhurst, D., Le Faucheur, F., Charny, A., Liatsos, V., Babiarz, J., Chan, K., Dudley, S., Westberg, L., Bader, A., and G. Karagiannis, "Pre-Congestion Notification Marking", [draft-briscoe-tsvwg-cl-phb-02](#) (work in progress), June 2006.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2211] Wroclawski, J., "Specification of the Controlled-Load Network Element Service", [RFC 2211](#), September 1997.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", [RFC 3168](#), September 2001.
- [RFC3246] Davie, B., Charny, A., Bennet, J., Benson, K., Le Boudec, J., Courtney, W., Davari, S., Firoiu, V., and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", [RFC 3246](#), March 2002.
- [RSVP-ECN]

Le Faucheur, F., Charny, A., Briscoe, B., Eardley, P., Babiarz, J., and K. Chan, "RSVP Extensions for Admission Control over Diffserv using Pre-congestion Notification", [draft-lefaucheur-rsvp-ecn-01](#) (work in progress), June 2006.

[Re-TCP] Briscoe, B., Jacquet, A., and A. Salvatori, "Re-ECN: Adding Accountability for Causing Congestion to TCP/IP", [draft-briscoe-tsvwg-re-ecn-tcp-02](#) (work in progress), June 2006.

14.2. Informative References

[CL-deploy]

Briscoe, B., Eardley, P., Songhurst, D., Le Faucheur, F., Charny, A., Babiarz, J., Chan, K., Westberg, L., Bader, A., and G. Karagiannis, "A Deployment Model for Admission Control over DiffServ using Pre-Congestion Notification", [draft-briscoe-tsvwg-cl-architecture-03](#) (work in progress), June 2006.

[CLoop_pol]

Salvatori, A., "Closed Loop Traffic Policing", Politecnico Torino and Institut Eurecom Masters Thesis , September 2005.

[ECN-BGP] Mortier, R. and I. Pratt, "Incentive Based Inter-Domain Routeing", Proc Internet Charging and QoS Technology Workshop (ICQT'03) pp308--317, September 2003, <<http://research.microsoft.com/users/mort/publications.aspx>>.

[ECN-MPLS]

Bruce, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", [draft-davie-ecn-mpls-00](#) (work in progress), June 2006.

[IXQoS]

Briscoe, B. and S. Rudkin, "Commercial Models for IP Quality of Service Interconnect", BT Technology Journal (BTTJ) 23(2)171--195, April 2005, <<http://www.cs.ucl.ac.uk/staff/B.Briscoe/pubs.html#ixqos>>.

[NSIS-RMD]

Bader, A., Westberg, L., Karagiannis, G., Kappler, C., and T. Phelan, "RMD-QOSM - The Resource Management in Diffserv QOS Model", [draft-ietf-nsis-rmd-06](#) (work in progress), February 2006.

[RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S.

- Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", [RFC 2205](#), September 1997.
- [RFC2207] Berger, L. and T. O'Malley, "RSVP Extensions for IPSEC Data Flows", [RFC 2207](#), September 1997.
- [RFC2208] Mankin, A., Baker, F., Braden, B., Bradner, S., O'Dell, M., Romanow, A., Weinrib, A., and L. Zhang, "Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement Some Guidelines on Deployment", [RFC 2208](#), September 1997.
- [RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", [RFC 2747](#), January 2000.
- [RFC2998] Bernet, Y., Ford, P., Yavatkar, R., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, B., Wroclawski, J., and E. Felstaine, "A Framework for Integrated Services Operation over Diffserv Networks", [RFC 2998](#), November 2000.
- [RFC3540] Spring, N., Wetherall, D., and D. Ely, "Robust Explicit Congestion Notification (ECN) Signaling with Nonces", [RFC 3540](#), June 2003.
- [Re-fb] Briscoe, B., Jacquet, A., Di Cairano-Gilfedder, C., Salvatori, A., Soppera, A., and M. Koyabe, "Policing Congestion Response in an Internetwork Using Re-Feedback", ACM SIGCOMM CCR 35(4)277--288, August 2005, <<http://www.acm.org/sigs/sigcomm/sigcomm2005/techprog.html#session8>>.
- [Smart_rtg] Goldenberg, D., Qiu, L., Xie, H., Yang, Y., and Y. Zhang, "Optimizing Cost and Performance for Multihoming", ACM SIGCOMM CCR 34(4)79--92, October 2004, <<http://citeseer.ist.psu.edu/698472.html>>.
- [Steps_DoS] Handley, M. and A. Greenhalgh, "Steps towards a DoS-resistant Internet Architecture", Proc. ACM SIGCOMM workshop on Future directions in network architecture (FDNA'04) pp 49--56, August 2004.

[Appendix A](#). Implementation

A.1. Ingress Gateway Algorithm for Blanking the RE flag

The ingress gateway receives regular feedback reporting the fraction of congestion marked octets for each aggregate arriving at the egress. So for each aggregate it should blank the RE flag on the same fraction of octets. It is more efficient to calculate the reciprocal of this fraction when the signalling arrives, $Z_0 = (1 / \text{Congestion-Level-Estimate})$. Z_0 will be the number of octets of packets the ingress should send with the RE flag set between those it sends with the RE flag blanked. Z_0 will also take account of the sustainable rate reported during the flow pre-emption process, if necessary.

A suitable pseudo-code algorithm for the ingress gateway is as follows:

```
=====
B_i = 0                /* interblank volume                */
for each PCN-capable packet {
    b = readLength()    /* set b to packet size                */
    B_i += b            /* accumulate interblank volume        */
    if B_i < b * Z_0 { /* test whether interblank volume...  */
        writeRE(1)
    } else {            /* ...exceeds blank RE spacing * pkt size*/
        writeRE(0)      /* ...and if so, clear RE              */
        B_i = 0         /* ...and re-set interblank volume     */
    }
}
=====
```

A.2. Downstream Congestion Metering Algorithms

A.2.1. Bulk Downstream Congestion Metering Algorithm

To meter the bulk amount of downstream pre-congestion in traffic crossing an inter-domain border, an algorithm is needed that accumulates the size of positive packets and subtracts the size of negative packets. We maintain two counters:

V_b: accumulated pre-congestion volume

B: total data volume (in case it is needed)

A suitable pseudo-code algorithm for a border router is as follows:


```

=====
V_b = 0
B   = 0
for each PCN-capable packet {
    b = readLength(packet)      /* set b to packet size      */
    B += b                      /* accumulate total volume */
    if readEECN(packet) == (Re-Echo || FNE) {
        V_b += b              /* increment...          */
    } elseif readEECN(packet) == ( AM(-1) || PM(-1) ) {
        V_b -= b              /* ...or decrement V_b... */
    }                          /* ...depending on EECN field */
}
=====

```

At the end of an accounting period this counter V_b represents the pre-congestion volume that penalties could be applied to, as described in [Section 5.3](#).

For instance, accumulated volume of pre-congestion through a border interface over a month might be V_b = 5PB (petabyte = 10^{15} byte). This might have resulted from an average downstream pre-congestion level of 1% on an accumulated total data volume of B = 500PB.

A.2.2. Inflation Factor for Persistently Negative Flows

The following process is suggested to complement the simple algorithm above in order to protect against the various attacks from persistently negative flows described in [Section 5.6.1](#). As explained in that section, the most important and first step is to estimate the contribution of persistently negative flows to the bulk volume of downstream pre-congestion and to inflate this bulk volume as if these flows weren't there. The process below has been designed to give an unboased estimate, but it may be possible to define other processes that achieve similar ends.

While the above simple metering algorithm is counting the bulk of traffic over an accounting period, the meter should also select a subset of the whole flow ID space that is small enough to be able to realistically measure but large enough to give a realistic sample. Many different samples of different subsets of the ID space should be taken at different times during the accounting period, preferably covering the whole ID space. During each sample, the meter should count the volume of positive packets and subtract the volume of negative, maintaining a separate account for each flow in the sample. It should run a lot longer than the large majority of flows, to avoid a bias from missing the starts and ends of flows, which tend to be positive and negative respectively.

Once the accounting period finishes, the meter should calculate the total of the accounts $V_{\{bI\}}$ for the subset of flows I in the sample, and the total of the accounts $V_{\{fI\}}$ excluding flows with a negative account from the subset I . Then the weighted mean of all these samples should be taken $a_S = \sum_{\text{forall } I} V_{\{fI\}} / \sum_{\text{forall } I} V_{\{bI\}}$.

If V_b is the result of the bulk accounting algorithm over the accounting period (Appendix A.2.1) it can be inflated by this factor a_S to get a good unbiased estimate of the volume of downstream congestion over the accounting period $a_S.V_b$, without being polluted by the effect of persistently negative flows.

[A.3.](#) Algorithm for Sanctioning Negative Traffic

{ToDo: Write up algorithms similar to [Appendix D](#) of [\[Re-TCP\]](#) for the negative flow monitor with flow management algorithm and the variant with bounded flow state.}

Author's Address

Bob Briscoe
BT & UCL
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/>

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

