Transport Area Working Group
Internet-Draft
Expires: August 31, 2006

B. Briscoe
BT & UCL

February 27, 2006

Emulating Border Flow Policing using Re-ECN on Bulk Data
draft-briscoe-tsvwg-re-ecn-border-cheat-00

Status of this Memo

Copyright Notice

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 emulates 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.

Table of Contents

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 services it aimed to offer are still very much required.

A recently proposed approach [CL-arch] 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 each other to comply with admission control and rate policing. This memo describes border policing measures to sanction networks that cheat each other. 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 claim that controlled load service can scale 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:

- A framework for admission control over Diffserv using pre-congestion notification [CL-arch] 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;

- 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 ensures upstream nodes honestly declare expected downstream congestion in all forwarded packets, which we then use to emulate border 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 Diffserv admission control scenario in the framework. Also, rather than using a TCP transport for regular congestion feedback, this memo specifies re-ECN using RSVP as the transport. We use the proposed minor extension of RSVP that allows it to carry congestion feedback [RSVP-

ECN], which is much less frequent but more precise than TCP.

Of course, network operators may choose to process per-flow signalling at their borders for their own reasons, such as per-flow accounting. But the goal of this document 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. To be clear, we are absolutely opposed to standardisation of technology that embeds particular business models into the Internet. Our aim here is to provide a new 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.

We specify this protocol solution in detail in Section 4, after specifying the inter-domain policing problem more precisely and briefly recapping the framework for providing admission control using pre-congestion notification in Section 3.

Having described the solution, this memo continues as follows: {ToDo: }

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]. But, even if the source is authorised and admitted at the flow level, it cannot necessarily be

trusted to send packets within the rate profile it requested. 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.

A signalled request refers to a flow of packets by its flow ID tuple (filter spec [RFC2205]) (or its security parameter index (SPI)& nbsp[RFC2207] if port numbers are hidden by IPsec encryption). But merely opening a pin-hole for packets that match an admitted flow ID is an insufficient policing mechanism. The packet rate must also be policed to keep the flow within the requested flow spec [RFC2205].

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 is probably not actually congested, but rather it is denying admission to new flows to protect bandwidth assigned to other lower priority services.

To summarise, in traditional reservation signalling architectures, if a network cannot trust a neighbouring upstream network to rate-police each reservation, it has to check for itself that the data fits within each of the reservations it has admitted.

3.2. Generic Scenario

We will now describe a generic internetworking scenario that we will use to describe and to test our bulk policing proposal. It consists of a number of networks and endpoints that do not fully trust each other to behave. In Section 6 we will tie down exactly what we mean by partial trust, and we will consider the various combinations where some networks do not trust each other and others are colluding together.

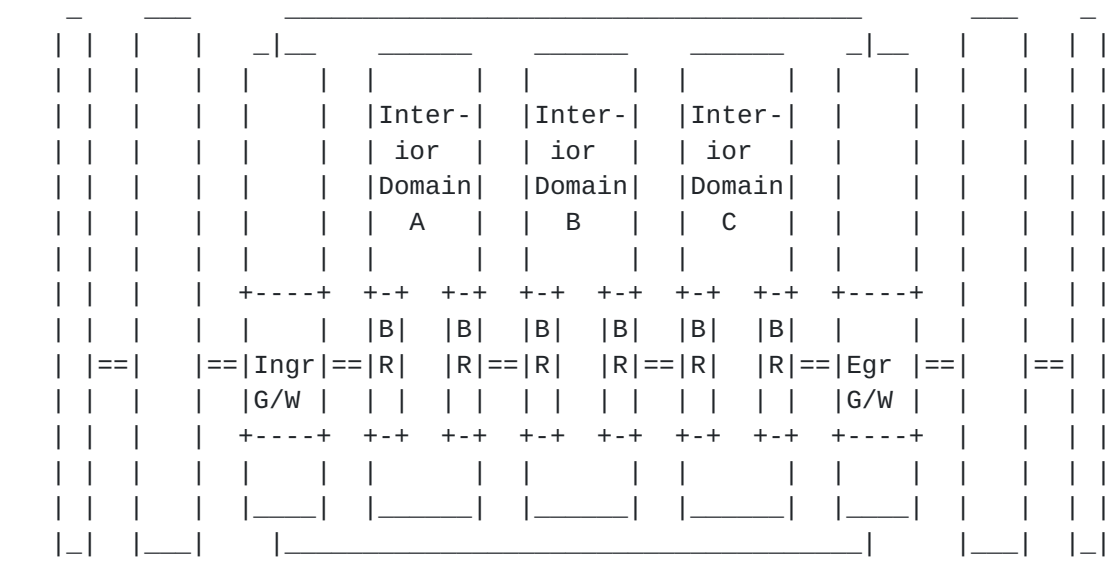

Figure 1: Generic Scenario (see text for explanation of terms)

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-arch] 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 4.4.

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. 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.

4. Re-ECN Protocol for an RSVP 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 virtual queue at its egress link [PCN]. 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 to be able to measure expected downstream pre-congestion on the paths between a border router and the egress gateways that packets are headed 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 pre-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 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.

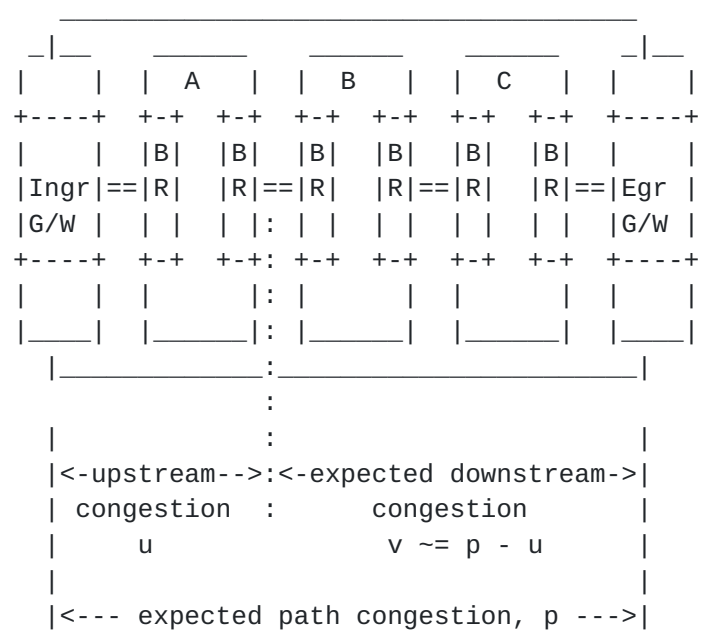

Figure 2: Re-ECN concept

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, deferring description of their semantics to the following sections. First we recap the re-ECN wire protocol proposed in [Re-TCP]. It uses the two bit ECN field broadly as in RFC3168 [RFC3168]. It also uses a new re-ECN extension (RE) bit. The actual position of the RE bit 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 bit. [Re-TCP] proposes using bit 48 (currently unused) in the IPv4 header for the RE bit, while it proposes an ECN extension header for IPv6.

Unlike the ECN field, the RE bit 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, an ingress gateway changes the setting of the RE bit, acting as a proxy for the sender, as permitted in the specification of re-ECN.

Although the RE bit 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 bit 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 bit	re-ECN codepoint	re-ECN meaning
00	Not-ECT	0	NRECT	Not re-ECN-capable transport
00	Not-ECT	1	NF	No feedback
01	ECT(1)	0	Re-Echo	Re-echoed congestion and RECT
01	ECT(1)	1	RECT	re-ECN capable transport
10	ECT(0)	0	--CU--	Currently unused
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

As permitted by RFC3168, [PCN] proposes new semantics for the ECN codepoints when combined with a Diffserv codepoint (DSCP) that uses pre-congestion notification. It also proposes various alternative encodings for these semantics, attempting to fit five states into the four available ECN codepoints by making various compromises. The five states are Not-ECT, ECT (ECN-capable transport), the ECN Nonce, Admission Marking (AM) and Pre-emption Marking (PM).

One of the five states was for the ECN Nonce [RFC3540], but the capability we describe in this memo supercedes 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 by a cheating receiver. 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 compromises that [PCN] explores ("Alternative 5") leaves out support for the ECN Nonce. Therefore we use that one. Then, with the addition of the RE bit, the 8 encodings of the extended ECN (EECN) field become those defined in the table below. Note that these codepoints only take on the semantics in the table below when combined with a Diffserv codepoint that the operator has defined as supporting pre-congestion notification.

ECN field	PCN codepoint	RE bit	re-ECN codepoint	re-ECN meaning
00	Not-ECT	0	NRECT	Not re-ECN-capable transport
00	Not-ECT	1	NF	No feedback
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)

For the rest of this memo, we will not distinguish between Admission Marking and Pre-emption Marking (unless stated otherwise). 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.

All but the "not re-ECN-capable transport" (NRECT) field imply the presence of an ECN-capable transport. Congested PCN-capable routers must drop rather than mark packets carrying the NRECT codepoint. Note that adding PCN-capability to a router will involve checking the RE bit as well as the ECN field and DSCP before deciding whether to drop or to mark a packet during congestion. Router implementations might well append the RE bit to their internal representation of the ECN field, treating them internally as one 3-bit extended ECN value.

4.3. Protocol Operation

In this section we will give an overview of the operation of the re-ECN protocol for an RSVP transport, deferring a detailed specification to the following sections.

The re-ECN protocol involves a simple tweak to the action of the gateway at the ingress edge of the CL region. In the framework just described [CL-arch], 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 bit 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 bit is set and cleared this way round for incremental deployent reasons (see [Re-TCP]). To avoid confusion we will use the term `blanking' (rather than marking) when the RE bit is cleared to "0", so we will talk of the `RE blanking fraction' as the fraction of octets with the RE bit cleared to "0".

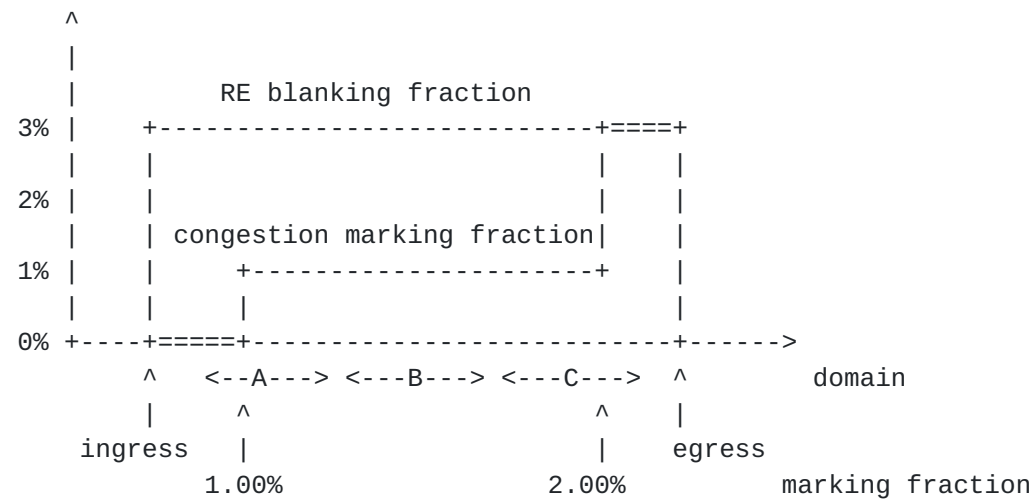

Figure 3: Example Re-ECN Codepoint Marking fractions (Imprecise)

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 two congested routers somewhere within domans A and C. And the table below shows the downstream pre-congestion measured at various border observation points along the path. These figures are actually reasonable approximations derived from more precise formulae given in Appendix A of [Re-TCP]. The RE bit 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%

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 arrive relatively infrequently compared to the speed with which congestion changes. Although this feedback will always be out of date, on average positive errors will 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, then the ingress gateway re-echoes it into the forward data path by blanking the RE bit. 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.

4.4. 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 framework [CL-arch].

However, with our new re-ECN scheme, the ingress does not know what proportion of the data probes should have the RE bit 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 NF codepoint of the extended ECN header in all probe packets (Table 2). As per the framework, 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 NF codepoint to be set on probes, given legacy firewalls etc. might discard such packets (because this flag had no prevous legitimate use). However, in the deployment scenarios envisaged for this admission control framework, 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 bit set.

Note that we have said SHOULD rather than MUST for the NF 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.

4.5. 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 aggregate bootstrap behaviour 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 NF 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 NF 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 NF packet, it will be seen later that sanctions may be incorrectly applied 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 NF 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 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 NF. This REQUIREMENT allows the design of network policers to be deterministic.

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 NF when it adds a new flow to an active aggregate and an NF packet need only be sent if a whole aggregate has been idle for more than 1 second.

5. Emulating Border Policing with Re-ECN

Note: In the rest of this memo, where the context makes it clear, we will loosely use the term 'congestion' rather than using the stricter 'downstream pre-congestion'. Also we will loosely talk of positive or negative traffic, meaning traffic where the moving average of the downstream pre-congestion metric is persistently positive or negative respectively.

The notion of positive and negative downstream pre-congestion is because downstream pre-congestion is calculated by subtracting the congestion marking fraction from the RE blanking fraction. Therefore packets can be considered to have a 'value multiplier' of +1, 0 or -1. Blanking the RE bit increments the 'value multiplier' of a packet. Congestion marking a packet decrements 'the value multiplier' (whether admission marking or pre-emption marking). Both together cancel each other out (a neutral or zero 'value-multiplier'). The NF codepoint is an exception. It has the same positive 'value multiplier' as a re-echoed packet. The table below specifies unambiguously the value multipliers of each extended ECN codepoint.

ECN field	RE bit	re-ECN codepoint	'Value multiplier'	re-ECN meaning
00	0	NRECT	n/a	Not re-ECN-capable transport
00	1	NF	+1	No feedback
01	0	Re-Echo	+1	Re-echoed congestion and RECT
01	1	RECT	0	re-ECN capable transport
10	0	AM(0)	0	Admission Marking with Re-Echo
10	1	AM(-1)	-1	Admission Marking
11	0	PM(0)	0	Pre-emption Marking with Re-Echo
11	1	PM(-1)	-1	Pre-emption Marking

Table 4: 'Sign' of Extended ECN Codepoints

Just as we will loosely talk of positive and negative traffic when we mean the level of downstream pre-congestion in the stream of traffic, we will also talk of positive or negative packets, meaning whether a packet contributes positively or negatively to downstream pre-congestion.

5.1. Policing Overview

To emulate border policing, the general idea is for each domain to apply financial penalties to its upstream neighbour in proportion to the amount of downstream pre-congestion that the upstream network sends across the border. This seems to encourage everyone to understate downstream pre-congestion to reduce the penalties they incur. But it is in the last domain's interest to create a balancing upward pressure by applying sanctions to flows where the marking fraction goes negative before the egress gateway.

Of course, some domains may trust other domains to comply without applying sanctions or penalties. In these cases, 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.

Figure 4 uses the same example as in previous sections to show the downstream pre-congestion marking fraction, v, across a path through the Internet. Downward arrows show the pressure for each domain to

underdeclare downstream pre-congestion in traffic they pass to the next domain, because of the penalties. 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, so if domain C agreed to pay for any downstream pre-congestion, it would give the egress gateway an incentive to overdeclare pre-congestion feedback and take the resulting profit from domain C.

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, Section 5.2 gives some examples of how these penalties might be implemented.

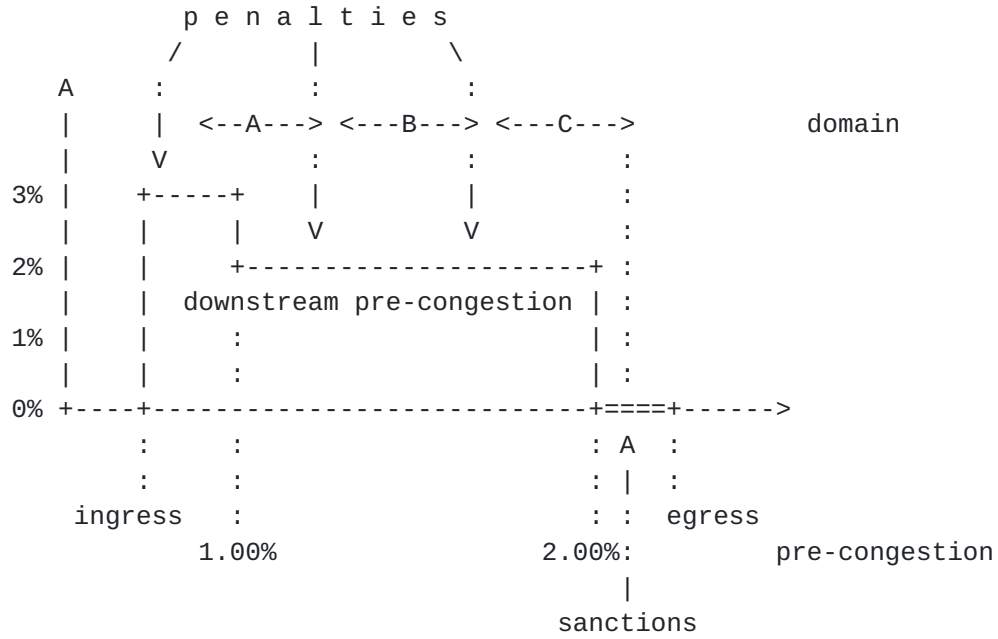

Figure 4: Policing Framework, showing creation of opposing pressures to underdeclare and overdeclare downstream pre-congestion, using penalties and sanctions

Any traffic 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 traffic 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. So this memo also does not propose to standardise any particular mechanism, but Section 5.4 does give examples of how the underlying re-ECN protocol

could be used to apply sanctions to persistently negative traffic. The ultimate sanction would be to drop such negative traffic indiscriminately, without regard to flows. A less drastic sanction might be to focus drop on specific packets in specific flows to remove the negative bias while doing minimal harm.

In all cases 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.

The incentive for domains not to tolerate negatively marked traffic depends on financial penalties never being negative. That is, 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. 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 overdeclarations due to lags in the feedback system, persistent downstream negative congestion can have no physical meaning and therefore must signify a problem.

The upward arrow at the egress of domain C at its border with the egress gateway in Figure 4 represents this incentive not to allow negative traffic. But the same upward pressure applies at every domain border (arrows not shown).

With the above penalty system, each domain seems to have a perverse incentive to fake pre-congestion. For instance domain B's profit depends on the difference between pre-congestion at its ingress (its revenue) and 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 usage revenue. 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.

Again, this memo does need to standardise any particular mechanism for routing based on re-ECN. Section 5.5 explains why no new standards would be needed for congestion routing as long as re-ECN marking had been standardised. That section also points to papers concerning optimising routing in the presence of usage charging.

5.2. 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 that 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 traffic.

Appendix A.2 gives a suggested algorithm for metering downstream congestion at a border router. It could hardly be simpler. It involves accumulating the volume of packets with the RE bit blanked and the volume of those with congestion marking and subtracting the two. In order to discard a persistent negative balance (see above), time is slotted into periods of say 10secs (or a time sufficient for a few rounds of feedback depending on the level of aggregation). Every timeslot, a positive balance between the two counters is accumulated into a long-term counter and reset. Whereas, if the balance during any timeslot is negative, it is discarded and a management alarm SHOULD also be raised. Over an accounting period (say a month) the single metric in the long term counter represents all the downstream congestion caused by traffic passing the border meter.

Congestion has the dimension of [byte], being the product of volume transferred [byte] and percentage pre-congestion [dimensionless]. The above algorithm effectively gives a measure of the volume transferred, but modulated by pre-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.

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 precise 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.

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 where there is greater connectivity. But penalties or charges for usage normally follow the same direction as the data flow---the direction of control at the network layer. 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 3 network, even though it is the provider.

The relative direction of penalties and charges is a constant source of confusion. 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.

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, B_v, using the notation of Appendix A.2. If they previously agreed the (fixed) price per byte of pre-congestion would be L, then the bill at the end of the month will simply be the product $L.B_v$, 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. But our aim is merely to show that border policing can at least work with this one model, then we can assume that operators might experiment with the metric in other models. Effectively tiered thresholds are just more coarse-grained approximations of the fine-grained case we choose to examine. 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.

5.3. 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 deaggregates correctly along with packet flows. This is because the penalty rises linearly with bit rate and linearly with congestion, because it is the product of them both. So if the packets crossing a border consist of 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 charges levied at every border the flow crosses in proportion to the amount of pre-congestion remaining on the path.

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. So, the system as a whole is fairly insensitive to the actual level of pre-congestion that each ingress chooses for triggering admission control. 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. 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---still low, but more noticeable. But this does not imply that usage /charges/ must also be low. That depends on the /price/ L.

For instance, combining capacity and volume charges is quite a common feature of interconnection agreements in today's Internet, particularly since p2p file-sharing became popular. Imagine that the monthly payment between two networks is made up of a volume charge and a capacity charge, and they usually turn out to be in a ratio of about 1:2 (not atypical). If charging for volume were replaced with charging for congested volume, one would expect the price of congestion to be arranged so that the total charge for usage remained about the same---still about one third of the total settlement. Because that is obviously the charge that the market has found is necessary to push back against usage. So, if an average pre-congestion fraction turned out to be 0.1%, one would expect that the price L per byte of pre-congestion would be about 1000 times the previously used per byte price for volume (before congestion metrics were available).

From the above example it can be seen why operators will 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.

Effectively, usage charges will continuously flow from ingress gateways to the places where there is mild pre-congestion, in proportion to the data rates from those gateways and to the path pre-congestion.

If anyone 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 reservation.

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 we discuss how networks can protect themselves from accidental or deliberate misconfiguration in neighbouring networks.

5.4. Policing 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).

If domain C doesn't trust the networks around it to behave honestly, it should install a monitor at its egress. This monitor 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.

Note that the monitor operates on flows but we would like it not to require per-flow state. This is why we have been careful to ensure that all flows MUST start with a packet marked with the NF codepoint. If a flow does not start with the NF codepoint, a monitor is likely to treat it unfavourably. This incentivises setting of the NF codepoint.

This also means that a monitor will be resistant to state exhaustion attacks from other networks, as the monitor never creates state unless an NF packet arrives. And an NF packet counts positive, so it will cost a lot for a network to send many of them.

Monitor algorithms will often maintain an average fraction of RE blanked packets across flows. When maintaining an average across flows, a monitor MUST ignore packets with the NF codepoint set. An ingress gateway sets the NF codepoint when it does not have the benefit of feedback from the ingress. So counting packets with FE 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 it is probably more advisable to raise a management alarm. So all ingresses cannot understate downstream pre-congestion without getting logged. Then the network operator can deal with the offending network at the human level, out of band.

5.5. Competitive Routing

Goldenberg et al [Smart_rtg] refers to various commercial product and presents its own algorithms for moving traffic between multihomed 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. Without the benefit of re-ECN, they 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 congestion will be charged for).

5.6. 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 expecting a rational response that trades off costs against benefits. It is usually reasonable to assume that other network operators 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 similar mechanisms at a network's borders to provide "defence in depth":

Highly positive flows RE blanked packets should be sampled and a small regular sample 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 rate 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 congestion marked packets should be sampled and a small regular sample 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 rate minus the congestion marking rate 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 postive (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 vagiaries 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 polices 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) intended to mitigate the potential impact if one network accidentally misconfiguring the workings of this protocol.

In the generic scenario we introduced in Figure 1 the ingress and egress gateways are shown in the most generic arrangement, 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 are 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.

Taking the generic scenario in Figure 1, 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 our primary concern is to be able to protect networks that don't do per-flow policing from those that do. 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 spell out the motivations that would create the lack of trust as 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 pay for.

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

- The more pre-congestion there is on a path across the Diffserv region, the higher the ingress gateway has to declare downstream pre-congestion v_0.

- because downstream pre-congestion should on average be zero at the egress

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

In the case of other forms of collusion (e.g. between domain A and C) it would be possible for say A & B to create a tunnel between theselves 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 effectively prevent A attacking it through a tunnel, by treating the tunnel end point as a direct link to a neighbouring network, which falls back to the regular scenario without collusion.

{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 an incentive to break it have mounted a full analysis.

7. Extensions

If a different signalling system, such as NSIS, were used, but providing admission control in a similar way using pre-congestion notification (e.g. with RMD [NSIS-RMD]) a similar approach to re-ECN could be used.

8. Design Choices and Rationale

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 crosses layers from the physical upwards. Therefore re-feedback polices congestion response based on physical interfaces not addresses. That is, the congestion leaving a physical interface can be policed at the interface, rather than the congestion on packets that claim to come from an address, which may be spoofed. Also, re-feedback does not actually require feedback. A source must act conservatively before it gets feedback.

On the subject of lack of feedback, the no feedback (NF) codepoint is motivated by arguments for a state set-up bit in IP to prevent state exhaustion attacks. This idea was first put forward by David Clark and documented in [Handley_Steps_DoS]. The idea is that network layer datagrams should signal explicitly when they require state to be created in the layer above (e.g. at flow start). Then the higher layer can refuse to create any state unless a datagram declares this intent. We believe the NF codepoint can be used to serve the same purpose as the proposed more generic state-set-up bit.

The re-feedback paper [Re-fb] also makes the case for using an economic interpretation of congestion, which is the basis of the incentives-based approach used in this memo. That paper also makes the case against the use of classic feedback if the economic interpretation of congestion is to be realised. The problem with using classic feedback for policing congestion is that it opens up receiving networks to `denial of funds' attacks.

{ToDo: Further Design Rationale will be included in future versions of this memo}

9. IANA Considerations

{ToDo:}This memo includes no request to IANA (yet).

10. 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 fit elsewhere in the memo or which comment on the 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. Despite only involving a few bits, there is sufficient complexity in the whole system that there are numerous possibilities for attacks not catered for. But as far as we are aware, none reap any benefit to the attacker. It will always be possible for one network to cause damage to another neighbouring network's traffic by dropping or corrupting it as it forwards 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 damage their traffic without any /direct/ gain to themselves.

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 intuitive 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 pre-emption of flows tends to be associated with much higher damage to an operator's reputation for robust quality than denying admission. 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 bit have been used rather wastefully. In effect the RE bit 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.

{ToDo: More security considerations will undoubtedly be added in future versions of this memo.}

11. Conclusions

Using pre-congestion is a promising technique to control flow admissions that will scale to any size network. However, it requires a mechanism to ensure that networks can interconnect even if they do not trust each to keep to the admission control protocols. We claim that the re-ECN protocol provides such a mechanism, so that one network can detect and prevent another network in the system fro cheating for its own gain.

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 (BT), Stephen Hailes (UCL), Francois Le Faucheur, Anna Charny (Cisco), Jozef Babiarz, Kwok-Ho Chan, Corey Alexander (Nortel), David Clark, Bill Lehr, Sharon Gillett (MIT) 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., and S. Dudley, "Pre-Congestion Notification", draft-briscoe-tsvwg-cl-phb-01 (work in progress), March 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-00 (work in progress), October 2005.

[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-01 (work in progress), March 2006.

14.2. Informative References

[CL-arch] Briscoe, B., Eardley, P., Songhurst, D., Le Faucheur, F., Charny, A., Babiarz, J., and K. Chan, "A Framework for Admission Control over DiffServ using Pre-Congestion Notification", draft-briscoe-tsvwg-cl-architecture-02 (work in progress), March 2006.

[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>.

[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>.

Appendix A. Implementation

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

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 bit on the same fraction of octets. It is more efficient to calculate the reciprocal of this fraction when the signalling arrives, Z_0 = 1 / Congestion-Level-Estimate, which will be the number of bytes of packets the ingress should send with the RE bit set between those it sends with the RE bit 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 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. Bulk Downstream Congestion Metering Algorithm

To meter the bulk amount of downstream pre-congestion in passing traffic an algorithm is needed that accumulates the size of packets with RE blanked (or NF set) and subtracts the size of congestion marked packets, but ignores a persistently negative balance over a duration of T ~ 10secs, say. Three counters need to be maintained:

B_v: accumulated pre-congestion volume

B_s: pre-congestion volume in timeslot

B_t: total data volume

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

```
=====================================================================
B_v = 0
B_s = 0
B_t = 0
t = timeNow() + T           /* divide into timeslots of few secs */
for each packet {
    b = readLength()            /* set b to packet size          */
    B_t += b                    /* accumulate total volume       */
    if readRE() == 0 || readEECN() == NF {
        B_s += b                /* increment...                  */
    } elseif readECN() == 1X {
        B_s -= b                /* ...or decrement B_s...        */
    }                           /*...depending on EECN field     */
    if timeNow() > t {      /* every timeslot...                 */
        if B_v > 0 {        /* count a negative balance as zero  */
            B_v += B_s      /* otherwise accumulate the balance  */
        }
        B_s = 0                 /* re-set the temp counter...    */
        t += T                  /* ...for the next timeslot      */
    }
}
=====================================================================
```

At the end of an accounting period this counter B_v represents the pre-congestion volume that penalties could be applied to, as described in Section 5.2.

For instance, accumulated volume of pre-congestion through a border interface over a month might be B_v = 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_t = 500PB.

A.3. Algorithm for Sanctioning Negative Traffic

{ToDo: Write up dropper with flow management algorithm and 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

Disclaimer of Validity

Copyright Statement

Acknowledgment

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