Congestion Exposure (ConEx) Working Group
Internet-Draft
Intended status: Informational
Expires: January 16, 2014

M. Mathis
Google, Inc
B. Briscoe
BT
July 15, 2013

# Congestion Exposure (ConEx) Concepts and Abstract Mechanism draft-ietf-conex-abstract-mech-07

## Abstract

This document describes an abstract mechanism by which senders inform the network about the congestion encountered by packets earlier in the same flow. Today, network elements at any layer may signal congestion to the receiver by dropping packets or by ECN markings, and the receiver passes this information back to the sender in transport-layer feedback. The mechanism described here enables the sender to also relay this congestion information back into the network in-band at the IP layer, such that the total amount of congestion from all elements on the path is revealed to all IP elements along the path, where it could, for example, be used to provide input to traffic management. This mechanism is called congestion exposure or ConEx. The companion document "ConEx Concepts and Use Cases" provides the entry-point to the set of ConEx documentation.

## Status of This Memo

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

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

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

This Internet-Draft will expire on January 16, 2014.

## Copyright Notice

Table of Contents

1. Introduction . . . . . 3
2. Overview . . . . . 3
2.1. Terminology . . . . . 6
3. Requirements for the ConEx Abstract Mechanism . . . . . 7
3.1. Requirements for ConEx Signals . . . . . 7
3.2. Requirements for the Audit Function . . . . . 8
3.3. Requirements for non-abstract ConEx specifications . . . . 8
4. Encoding Congestion Exposure . . . . . 10
4.1. Naive Encoding . . . . . 11
4.2. Null Encoding . . . . . 11
4.3. ECN Based Encoding . . . . . 12
4.4. Independent Bits . . . . . 12
4.5. Codepoint Encoding . . . . . 13
4.6. Units Implied by an Encoding . . . . . 13
5. Congestion Exposure Components . . . . . 14
5.1. Network Devices (Not modified) . . . . . 15
5.2. Modified Senders . . . . . 15
5.3. Receivers (Optionally Modified) . . . . . 15
5.4. Policy Devices . . . . . 16
5.4.1. Congestion Monitoring Devices . . . . . 16
5.4.2. Rest-of-Path Congestion Monitoring . . . . . 16
5.4.3. Congestion Policers . . . . . 17
5.5. Audit . . . . . 17
5.5.1. Using Credit to Simplify Audit . . . . . 20
6. Support for Incremental Deployment . . . . . 21
7. IANA Considerations . . . . . 23
8. Security Considerations . . . . . 23
9. Acknowledgements . . . . . 25
10. Comments Solicited . . . . . 25
11. References . . . . . 25
11.1. Normative References . . . . . 25
11.2. Informative References . . . . . 25

## 1. Introduction

This document describes an abstract mechanism by which, to a first approximation, senders inform the network about the congestion encountered by packets earlier in the same flow. It is not a complete protocol specification, because it is known that designing an encoding (e.g. packet formats, codepoint allocations, etc) is likely to entail compromises that preclude some uses of the protocol. The goal of this document is to provide a framework for developing and testing algorithms to evaluate the benefits of the ConEx protocol and to evaluate the consequences of the compromises in various different encoding designs.

A companion document [RFC6789] provides the entry point to the set of ConEx documentation. It outlines concepts that are pre-requisites to understanding why ConEx is useful, and it outlines various ways that ConEx might be used.

## 2. Overview

As typical end-to-end transport protocols continually seek out more network capacity, network elements signal whenever congestion results, and the transports are responsible for controlling this network congestion [RFC5681]. The more a transport tries to use capacity that others want to use, the more congestion signals will be attributable to that transport. Likewise, the more transport sessions sustained by a user and the longer the user sustains them, the more congestion signals will be attributable to that user. The goal of ConEx is to ensure that the resulting congestion signals are sufficiently visible and robust, because they are an ideal metric for networks to use as the basis of traffic management or other related functions.

Networks indicate congestion by three possible signals: packet loss, ECN marking or queueing delay. ECN marking and some packet loss may be the outcome of Active Queue Management (AQM), which the network uses to warn senders to reduce their rates. Packet loss is also the natural consequence of complete exhaustion of a buffer or other network resource. Some experimental transport protocols and TCP variants infer impending congestion from increasing queuing delay. However, delay is too amorphous to use as a congestion metric. In this and other ConEx documents, the term 'congestion signals' is generally used solely for ECN markings and packet losses, because they are unambiguous signals of congestion.

In both cases the congestion signals follow the route indicated in Figure 1. A congested network device sends a signal in the data stream on the forward path to the transport receiver, the receiver

passes it back to the sender through transport level feedback, and the sender makes some congestion control adjustment.

This document extends the capabilities of the Internet protocol suite with the addition of a new Congestion Exposure signal. To a first approximation this signal, also shown in Figure 1, relays the congestion information from the transport sender back through the internetwork layer where it is visible to any interested internetwork layer devices along the forward path. This document frames the engineering problem of designing the ConEx signal. The requirements are described in Section 3 and some example encoding are presented in Section 4. Section 5 describes all of the protocol components.

This new signal is expressly designed to support a variety of new policy mechanisms that might be used to instrument, monitor or manage traffic. The policy devices are not shown in Figure 1 but might be placed anywhere along the forward data path (see Section 5.4).

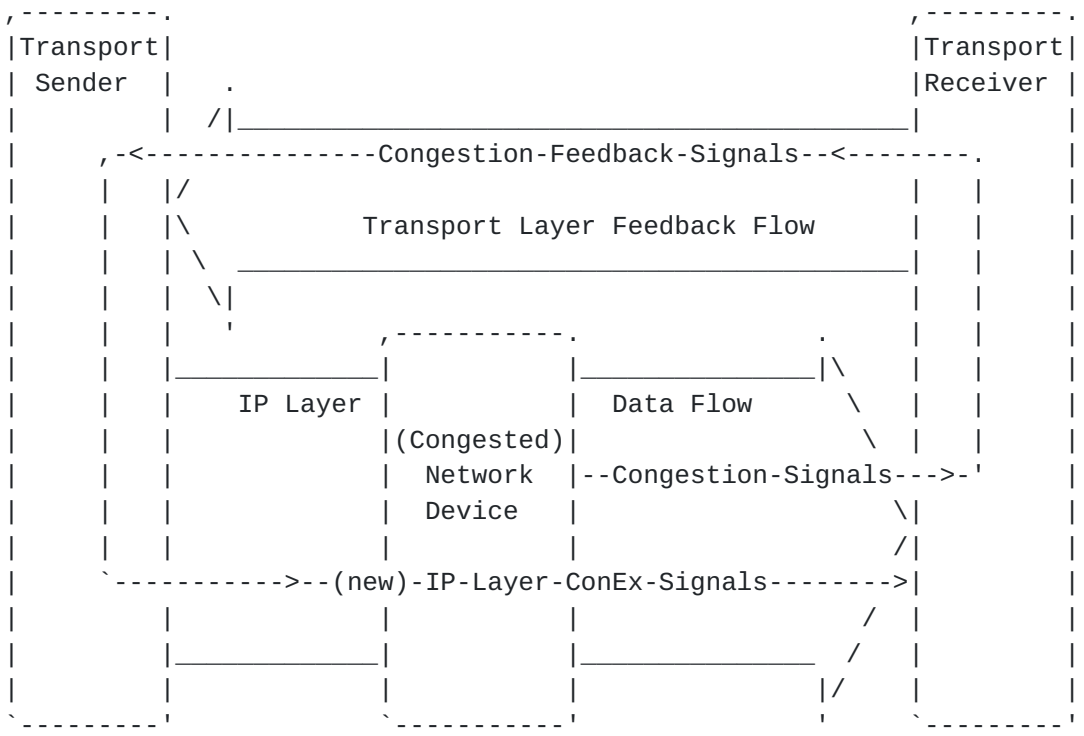


Figure 1: The Flow of Congestion and ConEx Signals

Since the policy devices can affect how traffic is treated it is assumed that there is an intrinsic motivation for users, applications or operating systems to understate the congestion that they are causing. Therefore, it is important to be able to audit ConEx signals, and to be able apply sufficient sanction to discourage cheating of congestion policies. The general approach to auditing is

to count signals on the forward path to confirm that there are never fewer ConEx signals than congestion signals. Many ConEx design constraints come from the need to assure that the audit function is sufficiently robust. The audit function is described in Section 5.5, however significant portions of this document (and prior research [Refb-dis]) is motivated by issues relating to the audit function and making it robust.

The congestion and ConEx signals shown in Figure 1 represent a series of discrete events: ECN marks or lost packets, carried by the forward data stream and fed back into the Internetwork layer. The policy and audit functions are most likely to act on the accumulated values of these signals, for which we use the term "volume". For example traffic volume is the total number of bytes delivered, optionally over a specified time interval and over some aggregate of traffic (e.g. all traffic from a site). While loss-volume is the total amount of bytes discarded from some aggregate over an interval. The term congestion-volume is defined precisely in [RFC6789]. Note that volume per unit time is (average) rate.

A design goal of the ConEx protocol is that the important policy mechanisms can be implemented per logical link without per flow state (see Section 5.4). However, the price to pay can be flow state to audit ConEx signals (Section 5.5). This is justified in that i) auditing at the edges, with limited per flow state, enables policy elsewhere, including in the core, without any per flow state; ii) auditing can use soft flow state, which does not require route pinning.

There is a long standing argument over units of congestion: bytes vs packets (see [I-D.ietf-tsvwg-byte-pkt-congest] and its references). Section 4.6 explains why this problem must be addressed carefully. However, this document does not take a strong position on this issue. Nonetheless, it does require that the units of congestion must be an explicitly stated property of any proposed encoding, and the consequences of that design decision must be evaluated along with other aspects of the design.

To be successful the ConEx protocol must have the property that the relevant stakeholders each have the incentive to unilaterally start on each stage of partial deployment, which in turn creates incentives for further deployment. Furthermore, legacy systems that will never be upgraded do not become a barrier to deploying ConEx. Issues relating to partial deployment are described in Section 6.

Note that ConEx signals are not intended to be used for fine-grained congestion control. They are anticipated to be most useful at longer time scales, for example the total congestion caused by a user might

serve as an input to higher level policy or accountability functions, designed to create incentives for improving user behavior, such as choosing to send large quantities of data at off-peak times, at lower data rates or with less aggressive protocols such as LEDBAT [I-D.ietf-ledbat-congestion] (see [RFC6789]).

Ultimately ConEx signals have the potential to provide a mechanism to regulate global Internet congestion. From the earliest days of congestion control research there has been a concern that there is no mechanism to prevent transport designers from incrementally making protocols more aggressive without bound and spiraling to a "tragedy of the commons" Internet congestion collapse. The "TCP friendly" paradigm was created in part to forestall this failure. However, it no longer commands any authority because it has little to say about the Internet of today, which has moved beyond the scaling range of standard TCP. As a consequence, many transports and applications are opening arbitrarily large numbers of connections or using arbitrary levels of aggressiveness. ConEx represents a recognition that the IETF cannot regulate this space directly because it concerns the behaviour of users and applications, not individual transport protocols. Instead the IETF can give network operators the protocol tools to arbitrate the space themselves, with better bulk traffic management. This in turn should create incentives for users, and designers of application and of transport protocols to be more mindful about contributing to congesting.

## 2.1. Terminology

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

ConEx signals in IP packet headers from the sender to the network:

Not-ConEx: The transport (or at least this packet) is not ConEx-capable.

ConEx-Capable: The transport is ConEx-Capable. This is the opposite of Not-ConEx.

ConEx Signal: A packet sent by a ConEx Capable transport. It carries at least one of the following signals:

- Re-Echo-Loss: The transport has experienced a loss.
- Re-Echo-ECN: The transport has experienced an ECN mark.
- Credit: The transport is building up credit to allow for any future delay in expected ConEx signals (see Section 5.5.1)
- ConEx-Not-Marked: The transport is ConEx-capable but is signaling none of Re-Echo-Loss, Re-Echo-ECN or Credit.

ConEx-Marked: At least one of Re-Echo-Loss, Re-Echo-ECN or Credit.

## 3. Requirements for the ConEx Abstract Mechanism

First time readers may wish to skim this section, since it is more understandable having read the entire document.

### 3.1. Requirements for ConEx Signals

Ideally, all the following requirements would be met by a Congestion Exposure Signal:

a. The ConEx Signal SHOULD be visible to internetwork layer devices along the entire path from the transport sender to the transport receiver. Equivalently, it SHOULD be present in the IPv4 or IPv6 header, and in the outermost IP header if using IP in IP tunneling. The ConEx Signal SHOULD be immutable once set by the transport sender. A corollary of these requirements is that the chosen ConEx encoding SHOULD pass silently without modification through pre-existing networking gear.

b. The ConEx Signal SHOULD be useful under only partial deployment. A minimal deployment SHOULD only require changes to transport senders. Furthermore, partial deployment SHOULD create incentives for additional deployment, both in terms of enabling ConEx on more devices and adding richer features to existing devices. Nonetheless, ConEx deployment need never be universal, and it is anticipated that some hosts and some transports may never support the ConEx Protocol and some networks may never use the ConEx Signals.

c. The ConEx signal SHOULD be timely. There will be a minimum delay of one RTT, and often longer if the transport protocol sends infrequent feedback (consider RTCP [RFC3550] for example).

d. The ConEx signal SHOULD be accurate and auditable. The general approach is to observe the volume of congestion signals and ConEx signals on the forward data path and verify that the ConEx signals do not under-represent the congestion signals (see Section 5.5). The simplest mechanism to compensate for the round trip delay between the signals is for the sender to include a "credit" signal to cover the yet to be observed congestion that might occur during this delay. (see Section 5.5.1 for details). Furthermore, the ConEx signals for packet loss and ECN marking SHOULD have distinct encodings because they are likely to require different auditing techniques.

It is already known that implementing ConEx signals is likely to entail some compromises, and therefore all the requirements above are expressed with the keyword 'SHOULD' rather than 'MUST'. The only mandatory requirement is that a concrete protocol description MUST give sound reasoning if it chooses not to meet some requirement.

### 3.2. Requirements for the Audit Function

The role and constraints on the audit function are described in Section 5.5. There is no intention to standardise the audit function. However, it is necessary to lay down the following normative constraints on audit behaviour so that transport designers will know what to design against and implementers of audit devices will know what pitfalls to avoid:

Minimal False Hits: Audit SHOULD introduce minimal false hits for honest flows;

Minimal False Misses: Audit SHOULD quickly detect and sanction dishonest flows, ideally on the first dishonest packet;

Transport Oblivious: Audit SHOULD NOT be designed around one particular rate response, such as any particular TCP congestion control algorithm or one particular resource sharing regime such as TCP-friendliness [RFC5348]. An important goal is to give ingress networks the freedom to unilaterally allow different rate responses to congestion and different resource sharing regimes [Evol_cc], without having to coordinate with other networks over details of individual flow behaviour;

Sufficient Sanction: Audit SHOULD introduce sufficient sanction (e.g. loss in goodput) such that senders cannot gain from understating congestion;

Proportionate Sanction: To the extent that the audit might be subject to false hits, the sanction SHOULD be proportionate to the degree to which congestion is understated. If audit over-punishes, attackers will find ways to harness it into amplifying attacks on others. Ideally audit should, in the long-run, cause the user to get no better performance than they would get by being accurate.

Manage Memory Exhaustion: Audit SHOULD be able to counter state exhaustion attacks. For instance, if the audit function uses flow-state, it should not be possible for senders to exhaust its memory capacity by gratuitously sending numerous packets, each with a different flow ID.

Identifier Accountability: Audit SHOULD NOT be vulnerable to `identity whitewashing', where a transport can label a flow with a new ID more cheaply than paying the cost of continuing to use its current ID [CheapPseud];

### 3.3. Requirements for non-abstract ConEx specifications

An experimental ConEx specification SHOULD describe the following protocol details:

Network Layer:

A. The specific ConEx signal encodings with packet formats, bit fields and/or code points;
B. An inventory of invalid combinations of flags or invalid codepoints in the encoding. Whether security gateways should normalise, discard or ignore such invalid encodings, and what values they should be considered equivalent to by ConEx-aware elements;
C. An inventory of any conflated signals or any other effects that are known to compromise signal integrity;
D. Whether the source is responsible for allowing for the round trip delay in ConEx signals (e.g. using a Credit marking), and if so whether Credit is maintained for the duration of a flow or degrades over time, and what defines the end of the duration of a flow;
E. A specification for signal units (bytes vs packets, etc), any approximations allowed and algorithms to do any implied conversions or accounting;
F. If the units are bytes a definition of which headers are included in the size of the packet;
G. How tunnels should propagate the ConEx encoding;
H. Whether the encoding fields are mutable or not, to ensure that header authentication, checksum calculation, etc. process them correctly. A ConEx encoding field SHOULD be immutable end-to-end, then end points can detect if it has been tampered with in transit;
I. if a specific encoding allows mutability (e.g. at proxies), an inventory of invalid transitions between codepoints. In all encodings, transitions from any ConEx marking to Not-ConEx MUST be invalid;
J. A statement that the ConEx encoding is only applicable to unicast and anycast, and that forwarding elements should silently ignore any ConEx signalling on multicast packets (they should be forwarded unchanged)
K. Definition of any extensibility;
L. Backward and forward compatibility and potential migration strategies. In all cases, a ConEx encoding MUST be arranged so that legacy transport senders implicitly send Not-ConEx;
M. Any (optional) modification to data-plane forwarding dependent on the encoding (e.g. preferential discard, interaction with Diffserv, ECN etc.);
N. Any warning or error messages relevant to the encoding.

Note regarding item J on multicast: A multicast tree may involve different levels of congestion on each leg. Any traffic management can only monitor or control multicast congestion at or near each receiver. It would make no sense for the sender to try

to expose "whole path congestion" in sent packets, because it cannot hope to describe all the differing congestion levels on every leg of the tree.

Transport Layer:

A. A specification of any required changes to congestion feedback in particular transport protocols.

B. A specification (or minimally a recommendation) for how a transport should estimate credits at the beginning of a new connection.

C. A specification of whether any other protocol options should (or must) be enabled along with an implementation of ConEx (e.g. at least attempting to negotiate ECN and SACK capability);

D. A specification of any configuration that a ConEx stack may require (or preferably confirmation that it requires no configuration);

E. A specification of the statistics that a protocol stack should log for each type of marking on a per-flow or aggregate basis.

Security:

A. An example of a strong audit algorithm suitable for detecting if a single flow is misstating congestion. This algorithm should present minimal false results, but need not have optimal scaling properties (e.g. may need per flow state).

B. An example of an audit algorithm suitable for detecting misstated congestion in a large aggregate (e.g. no per-flow state).

The possibility exists that these specifications over constrain the ConEx design, and can not be fully satisfied. An important part of the evaluation of any particular design will be a thorough inventory of all ways in which it might fail to satisfy these specifications.

## 4. Encoding Congestion Exposure

Most protocol specifications start with a description of packet formats and codepoints with their associated meanings. This document does not: It is already known that choosing the encoding for ConEx is likely to entail some engineering compromises that have the potential to reduce the protocol's usefulness in some settings. For instance the experimental ConEx encoding chosen for IPv6 [I-D.ietf-conex-destopt] had to make compromises on tunnelling. Rather than making these engineering choices prematurely, this document side steps the encoding problem by making it abstract. It describes several different representations of ConEx Signals, none of which are specified to the level of specific bits or code points.

The goal of this approach is to be as complete as possible for discovering the potential usage and capabilities of the ConEx

protocol, so we have some hope of making optimal design decisions when choosing the encoding. Even if experiments reveal particular problems due to the encoding, then this document will still serve as a reference model.

### 4.1. Naive Encoding

For tutorial purposes, it is helpful to describe a naive encoding of the ConEx protocol for TCP and similar protocols: set a bit (not specified here) in the IP header on each retransmission and on each ECN signaled window reduction. Network devices along the forward path can see this bit and act on it. For example any device along the path might limit the rate of all traffic if the rate of marked (congested) packets exceeds a threshold.

This simple encoding is sufficient to illustrate many of the benefits envisioned for ConEx. At first glance it looks like it might motivate people to deploy and use it. It is a one line code change that a small number of OS developers and content providers could unilaterally deploy across a significant fraction of all Internet traffic. However, this encoding does not support auditing so it would also motivate users and/or applications to misrepresent the congestion that they are causing [RFC3514]. As a consequence the naive encoding is not likely to be trusted and thus creates its own disincentives for deployment.

Nonetheless, this Naive encoding does present a clear mental model of how the ConEx protocol might function under various uses. It is useful for thought experiments where it can be stipulated that all participants are honest and it does illustrate some of the incentives that might be introduced by ConEx.

### 4.2. Null Encoding

In limited contexts it is possible to implement ConEx-like functions without any signals at all by measuring rest-of-path congestion directly from TCP headers. The algorithm is to keep at least one RTT of past TCP headers and matching each new header against the history to count duplicate data.

This could implement many ConEx policies, without any explicit protocol. It is fairly easy to implement, at least at low rate (e.g. in a software based edge router). However, it would only be useful in cases where the network operator can see the TCP headers. This is currently (2012) the vast majority of traffic because UDP, IPSEC and VPN tunnels are used far less than SSL or TLS over TCP/IP, which do not hide TCP sequence numbers from network devices. However, anyone specifically intending to avoid the attention of a congestion policy

device would only have to hide their TCP headers from the network operator (e.g. by using a VPN tunnel).

### 4.3. ECN Based Encoding

The re-ECN specification [I-D.briscoe-conex-re-ecn-tcp] presents an encoding of ConEx in IPv4 and IPv6 that was tightly integrated with ECN encoding in order to fit into the IPv4 header. ConEx and ECN are orthogonal signals in the sense that any individual packet may need to represent any one of the 4 possible combinations of signal values. Ideally their encoding should be entirely independent. However, given the limited number of header bits and/or code points, re-ECN chooses to partially share code points and to re-echo both losses and ECN with just one codepoint.

The central theme of the re-ECN work is an audit mechanism that provides sufficient disincentives against misrepresenting congestion [I-D.briscoe-conex-re-ecn-motiv]. It is analyzed extensively in Briscoe's PhD dissertation [Refb-dis]. For a tutorial background on re-ECN motivation and techniques, see [Re-fb, FairerFaster].

Re-ECN is an example of one chosen set of compromises attempting to meet the requirements of Section 3. The present document takes a step back, aiming to state the ideal requirements in order to allow the Internet community to assess whether different compromises might be better.

The problem with Re-ECN is that it requires that receivers be ECN enabled in addition to sender changes. Newer encodings [I-D.ietf-conex-destopt] overcome this problem by being able to represent loss and ECN based congestion separately.

### 4.4. Independent Bits

This encoding involves flag bits, each of which the sender can set independently to indicate to the network one of the following four signals:

ConEx (Not-ConEx) The transport is (or is not) using ConEx with this packet (the protocol must be arranged so that legacy transport senders implicitly send Not-ConEx; see network layer encoding requirement L in Section 3.3)

Re-Echo-Loss (Not-Re-Echo-Loss) The transport has (or has not) experienced a loss

Re-Echo-ECN (Not-Re-Echo-ECN) The transport has (or has not) experienced ECN-signaled congestion

Credit (Not-Credit)  The transport is (or is not) building up congestion credit (see Section 5.5 on the audit function)

A packet with ConEx set combined with all the three other flags cleared implies ConEx-Not-Marked

This encoding does not imply any exclusion property among the signals. Multiple types of congestion (ECN, loss) can be signalled on the same ACK. However, there will be many invalid combinations of flags (e.g. Not-ConEx combined with any of the ConEx-marked flags), which could be used to advantage against naive policy devices that only check each flag separately.

As long as the packets in a flow have uniform sizes, it does not matter whether the units of congestion are packets or bytes. However, if an application sends very irregular packet sizes, it may be necessary for the sender to mark multiple packets to avoid being in technical violation of the audit function.

### 4.5. Codepoint Encoding

This encoding involves signaling one of the following five codepoints:

ENUM {Not-ConEx, ConEx-Not-Marked, Re-Echo-Loss, Re-Echo-ECN, Credit}

Each named codepoint has the same meaning as in the encoding using independent bits in the previous section. The use of any one codepoint implies the negative of all the others.

Inherently, the semantics of most of the enumerated codepoints are mutually exclusive. 'Credit' is the only one that might need to be used in combination with either Re-Echo-Loss or Re-Echo-ECN, but even that requirement is questionable. It must not be forgotten that the enumerated encoding loses the flexibility to signal these two combinations, whereas the encoding with four independent bits is not so limited. Alternatively two extra codepoints could be assigned to these two combinations of semantics. The comment in the previous section about units also applies.

### 4.6. Units Implied by an Encoding

The following comments apply generally to all the other encodings.

Congestion can be due to exhaustion of bit-carry capacity, or exhaustion of packet processing power. When a packet is discarded or marked to indicate congestion, there is no easy way to know whether the lost or marked packet signifies bit-congestion or packet-

congestion. The above ConEx encodings that rely on marking packets suffer from the same ambiguity.

This problem is most acute when audit needs to check that one count of markings matches another. For example if there are ConEx markings on three large (1500B) packets, is that sufficient to match the loss of 5 small (60B) packets? If a packet-marking is defined to mean all the bytes in the packet are marked, then we have 4500B of Conex marked data against 300B of lost data, which is easily sufficient. If instead we are counting packets, then we have 3 ConEx packets against 5 lost packets, which is not sufficient. This problem will not arise when all the packets in a flow are the same size, but a choice needs to be made for flows in which packet sizes vary, such as BGP, SPDY and some variable rate video encoding schemes.

Whether to use bytes or packets is not obvious. For instance, the most expensive links in the Internet, in terms of cost per bit, are all at lower data rates, where transmission times are large and packet sizes are important. In order for a policy to consider wire time, it needs to know the number of congested bytes. However, high speed networking equipment and the transport protocols themselves sometimes gauge resource consumption and congestion in terms of packets.

This document does not take a strong position on this issue. However, a ConEx encoding will need to explicitly specify whether it assumes units of bytes or packets consistently for both congestion indications and ConEx markings (see network layer requirement E in Section 3.3). It may help to refer to the guidance in [I-D.ietf-tsvwg-byte-pkt-congest].

[I-D.ietf-tsvwg-byte-pkt-congest] advises that congestion indications should be interpreted in units of bytes when responding to congestion, at least on today's Internet. In any TCP implementation this is simple to achieve for varying size packets, given TCP SACK tracks losses in bytes. If an encoding is specified in units of bytes, the encoding should also specify which headers to include in the size of a packet (see network layer requirement F in Section 3.3).

## 5. Congestion Exposure Components

The components shown in Figure 1 as well as policy and audit are described in more detail.

### 5.1. Network Devices (Not modified)

Congestion signals originate from network devices as they do today. A congested router, switch or other network device can discard or ECN mark packets when it is congested.

### 5.2. Modified Senders

The sending transport needs to be modified to send Congestion Exposure Signals in response to congestion feedback signals (e.g. for the case of a TCP transport see [I-D.ietf-conex-tcp-modifications]). We want to permit ConEx without ECN (e.g. if the receiver does not support ECN). However, we want to encourage a ConEx sender to at least attempt to negotiate ECN (a ConEx transport protocol spec may require this), because it is believed that ConEx without ECN is harder to audit, and thus potentially exposed to cheating. Since honest users have the potential to benefit from stronger mechanisms to manage traffic they have an incentive to deploy ConEx and ECN together. This incentive is not sufficient to prevent a dishonest user from constructing (or configuring) a sender that enables ConEx after choosing not to negotiate ECN, but is should be sufficient to prevent this from being the sustained default case for any significant pool of users.

Permitting ConEx without ECN is necessary to facilitate bootstrapping other parts of ConEx deployment.

### 5.3. Receivers (Optionally Modified)

Any receiving transport may already feedback sufficiently useful signals to the sender so that it does not need to be altered.

If the transport receiver does not support ECN, then it's native loss signaling mechanism (required for compliance with existing congestion control standards) will be sufficient for the Sender to generate ConEx signals.

A traditional ECN implementation (RFC 3168 for TCP) signals congestion no more than once per round trip. The sender may require more precise feedback from the receiver otherwise it is at risk of appearing to be understating its ConEx Signals.

Ideally, ConEx should be added to a transport like TCP without mandatory modifications to the receiver. But an optional modification to the receiver could be recommended for precision (see [I-D.kuehlewind-tcpm-accurate-ecn]). This was the approach taken when adding re-ECN to TCP [I-D.briscoe-conex-re-ecn-tcp].

### 5.4. Policy Devices

Policy devices are characterised by a need to be configured with a policy related to the users or neighboring networks being served. In contrast, auditing devices solely enforce compliance with the ConEx protocol and do not need to be configured with any client-specific policy.

One of the design goals of the ConEx protocol is that none of the important policy mechanisms requires per flow state, and that policy mechanisms can even be implemented for heavily aggregated traffic in the core of the Internet with complexity akin to accumulating marking volumes per logical link. Of course, policy mechanisms may sometimes choose to focus down on individual flows, but ConEx aims to make aggregate policy devices feasible.

#### 5.4.1. Congestion Monitoring Devices

Policy devices can typically be decomposed into two functions i) monitoring the ConEx signal to compare it with a policy then ii) acting in some way on the result. Various actions might be invoked against 'out of contract' traffic, such as policing (see Section 5.4.3), re-routing, or downgrading the class of service.

Alternatively a policy device might not act directly on the traffic, but instead report to management systems that are designed to control congestion indirectly. For instance the reports might trigger capacity upgrades, penalty clauses in contracts, levy charges based on congestion, or merely send warnings to clients who are causing excessive congestion.

Nonetheless, whatever action is invoked, the congestion monitoring function will always be a necessary part of any policy device.

#### 5.4.2. Rest-of-Path Congestion Monitoring

ConEx signals indicate the level of congestion along a whole path from source to destination. In contrast, ECN signals monitored in the middle of a network indicate the level of congestion experienced so far on the path (of course, only in ECN-capable traffic.

If a monitor in the middle of a network (e.g. at a network border) measures both of these signals, it can subtract the level of ECN (path so far) from the level of ConEx (whole path) to derive a measure of the congestion that packets are likely to experience between the monitoring point and their destination (rest-of-path congestion).

It will often be preferable for policy devices to monitor rest-of-path congestion if they can, because it is a measure of the downstream congestion that the policy device can directly influence by controlling the traffic passing through it.

### 5.4.3. Congestion Policers

A congestion policer can be implemented in a very similar way to a bit-rate policer, but its effect can be focused solely on traffic of users causing congestion downstream, which ConEx signals make visible. Without ConEx signals, the only way to mitigate congestion is to blindly limit traffic bit-rate, on the assumption that high bit-rate is more likely to cause congestion.

A congestion policer monitors all ConEx traffic entering a network, or some identifiable subset. Using ConEx signals (and preferably subtracting ECN signals to yield rest-of-path congestion), it measures the amount of congestion that this traffic is contributing somewhere downstream. If this persistently exceeds a policy-configured 'congestion-bit-rate' the congestion policer can limit all the monitored ConEx traffic.

A congestion policer can be implemented by a simple token bucket applied to an aggregate. But unlike a bit-rate policer, it removes tokens only when it forwards packets that are ConEx-Marked, effectively treating Not-ConEx-Marked packets as invisible. Consequently, because tokens give the right to send congested bits, the fill-rate of the token bucket will represent the allowed congestion-bit-rate. This should provide sufficient traffic management without having to additionally constrain the straight bit-rate at all. See [I-D.briscoe-conex-policing] for details.

Note that the policing action is to introduce a throttle (delay through traffic) immediately upstream of the congestion policer. This throttle could include a queue with its own AQM, which potentially increases the whole path congestion. In effect the congestion policer has moved the congestion earlier in the path, and focused it on one user to protect downstream resources by reducing the congestion in the rest of the path.

## 5.5. Audit

The most critical aspect of ConEx is the capability to support robust auditing. It can be assumed that there will be an intrinsic motivation for users to understate the congestion that they are causing. Without strong audit functions the ConEx signal is likely to become inaccurate to the point of being useless. The most important feature of an encoding design is likely to be the

robustness of the auditing it supports.

The general approach is to compare the volume of ConEx signals to direct measures of actual congestion volume. The credit approach described in Section 5.5.1 can be used to guarantee that this is a strict bound: if the actual congestion exceeds the ConEx signal, then some congestion was understated and some sanction should be applied to the traffic. Although sanctions are beyond the scope of this document, an example sanction might be to throttle the traffic immediately upstream of the auditor to prevent the user from getting any advantage by understating congestion. Such a throttle would likely include some combination of delaying or dropping traffic.

A ConEx auditor might use one of the following techniques:

Generic loss auditing: For congestion signaled by loss, totally accurate auditing is not believed to be possible in the general case, because it involves a network node detecting the absence of some packets, when it cannot always necessarily identify retransmissions or missing packets. The missing packet might simply be taking a different route, or the IP payload may be encrypted.

It is for this reason that it is desirable to motivate the deploying of ECN, even though ECN is not strictly required for ConEx.

ECN auditing: Directly observe and compare the volume of ECN and ConEx marks. Since the volume of ECN marks rises monotonically along a path, ECN auditing is most accurate when located near the transport receiver. For this reason ECN should be monitored downstream of the predominant bottleneck.

TCP-specific loss auditing: For non-encrypted standard TCP traffic on a single path, an auditor could measure losses by detecting retransmissions, which appear as duplicate sequence numbers upstream of the loss and out of order data downstream of the loss. Since some reordering is present in the Internet, such a loss estimator would be most accurate near the sender. Such an audit device should treat non-ECN-capable packets with encrypted IP payload as Not-ConEx, even if they claim to be ConEx-capable, unless the operator knows it is also using one of the other two techniques below that can audit such packets against losses.

Predominant bottleneck loss auditing: For networks designed so that losses predominantly occur under the control of one IP-aware bottleneck node on the path, the auditor could be located at this bottleneck. It could simply compare ConEx Signals with actual local packet discards (and ECN marks). This is a good model for most consumer access networks where audit accuracy could well be sufficient even if losses occasionally occur elsewhere in the network.

Although the auditor at the predominant bottleneck would not be able to count losses at other nodes, transports would not know where losses were occurring either. Therefore a transport would not know which losses it could cheat and which ones it couldn't without getting caught.

ECN tunnel loss auditing: A network operator can arrange IP-in-IP tunnels (or IP-in-MPLS etc.) so that any losses within the tunnels are deferred until the tunnel egress. Then the audit function can be deployed at the egress and be aware of all losses. This is possible by enabling ECN marking on switches and routers within a tunnel, irrespective of whether end-systems support ECN, by exploiting a side-effect of the way tunnels handle the ECN field. After encapsulation at the tunnel ingress, the network should arrange for any non-ECN packets (with '00' in ECN field of the outer) to be set to the ECN-capable transport (ECT(0)) codepoint. Then, if they experience congestion at one of the ECN-capable switches or routers within the tunnel, some will be ECN-marked rather than immediately dropped. However, when the tunnel decapsulator strips the outer from such an ECN-marked packet, if it finds the inner header has '00' in the ECN field (meaning that the endpoints do not support ECN) it will automatically drop the packet, assuming it complies with [RFC6040]. Thus, an audit function at the decapsulator can know which packets would have been dropped within the tunnel (and even which are genuinely ECN-marked for the end-to-end protocol). Non-ECN end-systems outside the tunnel see no sign of the use of ECN internally.

In addition, other audit techniques may be identified in the future.

[Refb-dis] gives a comprehensive inventory of attacks against audit proposed by various people. It includes pseudocode for both deterministic and statistical audit functions designed to thwart these attacks and analyses the effectiveness of an implementation. Although this work is specific to the re-ECN protocol, most of the material is useful for designing and assessing audit of other specific ConEx encodings, against both ECN and loss.

The auditing function should be able to trigger sufficient sanction to discourage understating congestion [Salvatori05]. This seems to require designing the sanction in concert with the policy functions, even though they might be implemented in different parts of the network. However, [Refb-dis] proves audit and policy functions can be independent as long as audit drops sufficient traffic to 'normalise' actual congestion signals to be no greater than ConEx signals. Note that in the future it might prove to be desirable to provide advice on uniformly implementing sanctions, because otherwise insufficient sanctions impairs the ability to implement policy elsewhere in the network.

Some of the audit algorithms require per flow state. This cost is expected to be tolerable, because these techniques are most apropos near the edges of the network, where traffic is generally much less aggregated, so the state need not overwhelm any one device. The flow-state required for audit creates itself as it detects new flows. Therefore a flow will not fail if it is re-routed away from the audit box currently holding its flow-state, so auditing does not require route pinning and works fine with multipath flows.

Holding flow-state seems to create a vulnerability to attacks that exhaust the auditor's memory by opening numerous new short flows. The audit function can protect itself from this attack by not allocating new flow-state unless a ConEx-marked packet arrives (e.g. credit at the start of a flow). Because policy devices rate limit ConEx-marked packets, this sets a natural limit to the rate at which a source can create flow-state in audit devices.

Auditing can be distributed and redundant. One flow may be audited in multiple places, using multiple techniques. Some audit techniques do not require any per flow state and can be applied to aggregate traffic. These might be able to detect the presence of understated congestion at large scale and support recursively hunting for individual flows that are understating their congestion. Even at large scales, flows can be randomly selected for individual auditing.

Sampling techniques can also be used to bound the total auditing memory footprint, although the implementer must be wary of "identifier white washing when caught" tactics where a source cheats until caught by sampling, then simply discards that flow ID and starts cheating with a new one.

### 5.5.1. Using Credit to Simplify Audit

At the audit function, there will be an inherent delay of at least one round trip between a congestion signal and the subsequent ConEx signal it triggers, as shown in Figure 1. However, the audit function cannot be expected to wait for a round trip to check that one signal balances the other, because that requires excessive state and the auditor cannot easily determine the RTT of each flow.

The simplest mechanism to compensate for the round trip delay between the signals is to have the sender include a "credit" signal to cover the yet to be observed congestion that might occur during this delay. The transport signals sufficient credit in advance to cover congestion expected during its feedback delay. Then, the audit function does not need to make allowance for round trip delays that it cannot quantify. This design choice correctly makes the transport responsible for both minimizing feedback delay and for the risk that

packets in flight will cause congestion to others before the source can react.

Making the source responsible for allowing for the round trip delay in ConEx signals is a design choice that needs to be consistently applied, as is the question of whether Credit markings continue to maintain their value for the duration of a flow or expire or degrade over time. Any such requirements should be defined in a particular ConEx encoding specification (see network layer encoding requirement D in Section 3.3).

For example, imagine the audit function keeps a running account of the balance between actual congestion signals (loss or ECN), which it counts as negative, and ConEx signals, which it counts as positive. Having made the transport responsible for round trip delays, it will be expected to have pre-loaded the audit function with some credit at the start. Therefore, if the balance ever goes negative, the audit function can immediately start punishing a flow, without any grace period.

## 6. Support for Incremental Deployment

The ConEx abstract protocol described so far is intended to support incremental deployment in every possible respect. For convenience, the following list collects together all the features of ConEx that support incremental deployment, and points to further information on each:

Packets: The wire protocol encoding allows each packet to indicate whether it is using ConEx or not (see Section 4 on Encoding Congestion Exposure).

Senders: ConEx requires a modification to the source in order to send ConEx packet markings (see Section 5.2). Although ConEx support can be indicated on a packet-by-packet basis, it is likely that all the packets in a flow will either consistently support ConEx or consistently not. It is also likely that, if the implementation of a transport protocol supports ConEx, all the packets sent from that host using that protocol will be ConEx marked.

The implementations of some of the transport protocols on a host might not support ConEx (e.g. the implementation of DNS over UDP might not support ConEx, while perhaps RTP over UDP and TCP will). Any non-upgraded transports and non-upgraded hosts will simply continue to send regular Not-ConEx packets as always.

A network operator can create incentives for senders to voluntarily reveal ConEx information (see the item on incremental deployment by 'Networks' below).

Receivers: A ConEx source should be able to work without a modified receiver. However, without sufficiently precise congestion feedback from the receiver, the source may have to conservatively send extra ConEx markings in order to avoid understating congestion. The need for more precise receiver feedback is not exclusive to ConEx, for instance Data Centre TCP (DCTCP [DCTCP]) uses precise feedback to good effect. Nonetheless, if a receiver offers precise feedback, [I-D.kuehlewind-tcpm-accurate-ecn] it will be best if ConEx uses it (see Section 5.3).

Proxies: Although it was stated above that ConEx requires a modification to the source, ConEx signals could theoretically be introduced by a proxy for the source, as long as it can intercept feedback from the receiver. Similarly, more precise feedback could thoretically be provided by a proxy for the receiver rather than modifying the receiver itself.

Forwarding: No modification to forwarding or queuing is needed for ConEx.

However, once ConEx is deployed, it is possible that a queue implementation could optionally take advantage of the ConEx information in packets. For instance, it has been suggested [I-D.briscoe-conex-re-ecn-tcp] that a queue would be more robust against flooding if it preferentially discarded Not-ConEx packets then Not-Marked ConEx packets.

A ConEx sender re-echoes congestion whether the queues signaling congestion are ECN-enabled or not. Nonetheless, auditing works best if most congestion is indicated by ECN rather than loss (see Section 3). Also, monitoring rest-of-path congestion is not accurate if there are congested non-ECN queues upstream of the monitoring point (Section 5.4.2).

Networks: If a subset of traffic sources (or proxies) use ConEx signals to reveal congestion in the internetwork layer, a network operator can choose (or not) to use this information for traffic management. As long as the end-to-end ConEx signals are present, each network can unilaterally choose to use them--independently of whether other networks do.

ConEx marked packets may safely traverse a network that ignores them. ConEx signals are defined to remain unchanged once set by the sender, but some encodings may allow changes in transit (e.g. by proxies). In no circumstances will a network node change ConEx marked packets to Not-ConEx (network layer encoding requirement I in Section 3.3). If necessary, endpoints should be able to detect if a network is removing ConEx signals (network layer encoding requirement H in Section 3.3).

An operator can deploy policy devices (Section 5.4) wherever

traffic enters its network, in order to monitor the downstream congestion that incoming traffic contributes to, and control it if necessary. A network operator can create incentives for the developers of sending applications and transports to voluntarily reveal ConEx information. Without ConEx information, a network operator tends to have to limit the bit-rate or volume from a site more than is necessary, just in case it might congest others. With ConEx information, the operator can solely limit congestion-causing traffic, and otherwise allow complete freedom. This greater freedom acts as an inducement for the source to volunteer ConEx information. An operator may also monitor whether a source transport has sent ConEx packets, and treat the same transport with greater suspicion (e.g. a more stringent rate-limit) whenever it selectively sends packets without ConEx support. See [RFC6789] for further discussion of deployment incentives for networks and references to scenarios where some networks use ConEx-based policy devices and others don't.

An operator can deploy audit devices (Section 5.5) unilaterally within its own network to verify that traffic sources are not understating ConEx information. From the viewpoint of one network operator (say N_a), it only cares that the level of ConEx signaling is sufficient to cover congestion in its own network. If traffic continues into a congested downstream network (say N_b), it is of no concern to the first network (N_a) if the end-to-end ConEx signaling is insufficient to cover the congestion in N_b as well. This is N_b's concern, and N_b can both detect such anomalous traffic and deal with it using ConEx-based policy devices (Section 5.4).

## 7. IANA Considerations

This memo includes no request to IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

## 8. Security Considerations

The only known risk associated with ConEx is that users and applications are very likely to be motivated to under-represent the congestion that they are causing. Significant portions of this document are about mechanisms to audit the ConEx signals and create sufficient sanction to inhibit such under-representation. In particular see Section 5.5.

Security attacks and their defences are best discussed against a concrete protocol specification, not the abstract mechanism of this

document. A concrete ConEx protocol will need to be accompanied by a document describing how the protocol and its audit mechanisms defend against likely attacks. [Refb-dis] will be a useful source for such a document. It gives a comprehensive inventory of attacks against audit that have been proposed by various parties. It includes pseudocode for both deterministic and statistical audit functions designed to thwart these attacks and analyses the effectiveness of an implementation.

However, [Refb-dis] is specific to the re-ECN protocol, which signalled ECN & loss together, whereas ConEx signals them separately. Therefore, although likely attacks will be similar, there will be more combinations of attacks to worry about, and defences and their analysis are likely to be a little different for ConEx.

The main known attacks that a security document for a concrete ConEx protocol will need to address are listed below, and [Refb-dis] should be referred to for how re-ECN was designed to defend against similar attacks:

- Attacks on the audit function (see Section 7.5 of [Refb-dis]):
  - Flow ID Whitewashing: Designing the audit function so that a source cannot gain from starting a new flow once audit has detected cheating in a previous flow.
  - Dragging Down an Aggregate: Avoiding audit discarding packets from all flows within an aggregate, which would allow one flow to pull down the average so that the audit function would discard packets from all flows, not just the offending flow.
  - Dragging Down a Spoofed Flow ID: An attacker understates ConEx markings in packets that spoof another flow, which fools the audit function into dropping the genuine user's packets.
- Attacks by networks on other networks (see Section 8.2 of [Refb-dis]):
  - Dummy Traffic: Sending dummy traffic across a border with understated ConEx markings to bring down the average ConEx markings in the aggregate of border traffic. This attack can be combined with a TTL that expires before the packets reach an audit function.
  - Signal Poisoning with 'Cancelled' Marking: Sending high volumes of valid packets that are both ConEx-Marked and ECN-Marked, which seems to represent congestion upstream, but it makes these packets immune to being further ECN-Marked downstream.

It is planned to document all known attacks and their defences (including all the above) in the RFC series against a concrete ConEx protocol specification. In the interim, [Refb-dis] and its references should be referred to for details and ways to address these attacks in the case of re-ECN.

## 9. Acknowledgements

This document was improved by review comments from Toby Moncaster, Nandita Dukkipati, Mirja Kuehlewind, Caitlin Bestler, John Leslie, Ingemar Johansson and David Wagner.

## 10. Comments Solicited

Comments and questions are encouraged and very welcome. They can be addressed to the IETF Congestion Exposure (ConEx) working group mailing list <conex@ietf.org>, and/or to the authors.

## 11. References

### 11.1. Normative References

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

### 11.2. Informative References

[CheapPseud] Friedman, E. and P. Resnick, "The Social Cost of Cheap Pseudonyms", Journal of Economics and Management Strategy 10(2)173--199, 1998.

[DCTCP] Alizadeh, M., Greenberg, A., Maltz, D., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., and M. Sridharan, "Data Center TCP (DCTCP)", ACM SIGCOMM CCR 40(4)63--74, October 2010, <http://portal.acm.org/citation.cfm?id=1851192>.

[Evol_cc] Gibbens, R. and F. Kelly, "Resource pricing and the evolution of congestion control", Automatica 35(12)1969--1985, December 1999, <http://www.statslab.cam.ac.uk/~frank/evol.html>.

[FairerFaster] Briscoe, B., "A Fairer, Faster Internet Protocol", IEEE

Spectrum Dec 2008:38--43, December 2008, <http://bobbriscoe.net/projects/refb/#fairfastip>.

[I-D.briscoe-conex-policing] Briscoe, B., "Network Performance Isolation using Congestion Policing", draft-briscoe-conex-policing-00 (work in progress), February 2013.

[I-D.briscoe-conex-re-ecn-motiv] Briscoe, B., Jacquet, A., Moncaster, T., and A. Smith, "Re-ECN: A Framework for adding Congestion Accountability to TCP/IP", draft-briscoe-conex-re-ecn-motiv-00 (work in progress), April 2012.

[I-D.briscoe-conex-re-ecn-tcp] Briscoe, B., Jacquet, A., Moncaster, T., and A. Smith, "Re-ECN: Adding Accountability for Causing Congestion to TCP/IP", draft-briscoe-conex-re-ecn-tcp-00 (work in progress), April 2012.

[I-D.ietf-conex-destopt] Krishnan, S., Kuehlewind, M., and C. Ucendo, "IPv6 Destination Option for ConEx", draft-ietf-conex-destopt-03 (work in progress), September 2012.

[I-D.ietf-conex-tcp-modifications] Kuehlewind, M. and R. Scheffenegger, "TCP modifications for Congestion Exposure", draft-ietf-conex-tcp-modifications-02 (work in progress), May 2012.

[I-D.ietf-ledbat-congestion] Shalunov, S., Hazel, G., Iyengar, J., and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)", draft-ietf-ledbat-congestion-10 (work in progress), September 2012.

[I-D.ietf-tsvwg-byte-pkt-congest] Briscoe, B. and J. Manner, "Byte

and Packet Congestion Notification", draft-ietf-tsvwg-byte-pkt-congest-08 (work in progress), August 2012.

[I-D.kuehlewind-tcpm-accurate-ecn] Kuehlewind, M. and R. Scheffenegger, "Accurate ECN Feedback Option in TCP", draft-kuehlewind-tcpm-accurate-ecn-option-01 (work in progress), July 2012.

[RFC3514] Bellovin, S., "The Security Flag in the IPv4 Header", RFC 3514, April 2003.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC5348] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, September 2008.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, September 2009.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, November 2010.

[RFC6789] Briscoe, B., Woundy, R., and A. Cooper, "Congestion Exposure (ConEx) Concepts and Use Cases", RFC 6789, December 2012.

[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>.

[Refb-dis] Briscoe, B., "Re-feedback: Freedom with Accountability for Causing Congestion in a Connectionless Internetwork", UCL PhD Dissertation , 2009, <http://bobbriscoe.net/projects/refb/#refb-dis>.

[Salvatori05] Salvatori, A., "Closed Loop Traffic Policing", Politecnico Torino and Institut Eurecom Masters Thesis , September 2005.

## Authors' Addresses

Matt Mathis
Google, Inc
1600 Amphitheater Parkway
Mountain View, California 93117
USA

EMail: mattmathis at google.com

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

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