Congestion Exposure (ConEx) Working Group   M. Mathis
Internet-Draft   Google, Inc
Intended status: Informational   B. Briscoe
Expires: September 15, 2011   BT
                                                              March 14, 2011

Congestion Exposure (ConEx) Concepts and Abstract Mechanism
draft-ietf-conex-abstract-mech-01

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, the network may signal congestion to the receiver by ECN markings or by dropping packets, and the receiver passes this information back to the sender in transport-layer feedback. The mechanism to be developed by the ConEx WG will enable the sender to also relay this congestion information back into the network in-band at the IP layer, such that the total level of congestion is visible to all IP devices along the path, from where it could, for example, provide input to traffic management.

Table of Contents

1. Introduction

One of the required functions of a transport protocol is controlling congestion in the network. There are three techniques in use today for the network to signal congestion to a transport:

- The most common congestion signal is packet loss. When congested, the network simply discards some packets either as part of an active queue management function [RFC2309] or as the consequence of a queue overflow or other resource starvation. The transport receiver detects that some data is missing and signals such through transport acknowledgments to the transport sender (e.g. TCP SACK options). The sender performs the appropriate congestion control rate reduction (e.g. [RFC5681] for TCP) and, if it is a reliable transport, it retransmits the missing data.
- If the transport supports explicit congestion notification (ECN) [RFC3168] or pre-congestion notification (PCN) [RFC5670] , the transport sender indicates this by setting an ECN-capable transport (ECT) codepoint in every packet. Network devices can then explicitly signal congestion to the receiver by setting ECN bits in the IP header of such packets. The transport receiver communicates these ECN signals back to the sender, which then performs the appropriate congestion control rate reduction.
- Some experimental transport protocols and TCP variants [Vegas] sense queuing delays in the network and reduce their rate before the network has to signal congestion using loss or ECN. A purely delay-sensing transport will tend to be pushed out by other competing transports that do not back off until they have driven the queue into loss. Therefore, modern delay-sensing algorithms use delay in some combination with loss to signal congestion (e.g. LEDBAT [I-D.ietf-ledbat-congestion], Compound [I-D.sridharan-tcpm-ctcp]). In the rest of this document, we will confine the discussion to concrete signals of congestion such as loss and ECN. We will not discuss delay-sensing further, because it can only avoid these more concrete signals of congestion in some circumstances.

In all 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 proposes to extend the capabilities of the Internet protocol suite with the addition of a ConEx Signal that, to a first approximation, relays the congestion information from the transport sender back through the internetwork layer. That signal is shown in Figure 1. It would be visible to all internetwork layer devices along the forward (data) path and is intended to support a number of

new policy-controlled mechanisms that might be used to manage traffic.

There is no expectation that internetwork layer devices will do fine-grained congestion control using ConEx information. That is still probably best done at the transport sender. Rather, the network will be able to use ConEx information to do better bulk traffic management, which in turn should incentivize end-system transports to be more careful about congesting others [I-D.conex-concepts-uses].

```
+---------+                                               +---------+
|Transport|             +-----------+                     |Transport|
| Sender  |>=Data=Path=>|(Congested)|>=====Data=Path=====>| Receiver|
|         |             |  Network  |>-Congestion-Signal->|---.     |
|         |             |   Device  |                     |   |     |
|         |             +-----------+                     |   |     |
|         |                                               |   |     |
|         |<==Feedback=Path=============================<|   |     |
|     ,---|<--Transport Layer returned Congestion Signal-<|<--'     |
|     |   |                                               |         |
|     |   |>==============Data=Path======================>|         |
|     `-->|>---------(new)-IP layer ConEx Signal--------->|         |
|         |        (Carried in Data Packet Headers)       |         |
+---------+                                               +---------+
```

Not shown are policy devices along the data path that observe the ConEx Signal, and use the information to monitor or manage traffic. These are discussed in Section 4.4.

Figure 1

1.1. 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 {ToDo: These are placeholders for whatever words we decide to use}:

Not-ConEx: The transport is not ConEx-capable

ConEx-Capable: The transport is ConEx-Capable. This is the opposite of Not-ConEx and implies one of the following signals

- Re-Echo-Loss: (aka Purple) The transport has experienced a loss
- Re-Echo-ECN: (aka Black) The transport has experienced an ECN mark

Credit: (aka Green) The transport is building up credit to allow for any future delay in expected ConEx signals (see Section 4.3.1)

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.

2. Requirements for the ConEx Signal

Ideally, all the following requirements would be met by a Congestion Exposure Signal. However it is already known that some compromises will be necessary, therefore all the requirements 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 any of these requirements:

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 accurate. In potentially hostile environments such as the public Internet, it SHOULD be possible for techniques to be deployed to audit the Congestion Exposure Signal by comparing it to the actual congestion signals on the forward data path. The auditing mechanism must have a capability for providing sufficient disincentives against misreported congestion, such as by throttling traffic that reports less congestion than it is actually experiencing.

d. The ConEx Signal SHOULD be timely. There will be a delay between the time when an auditing device sees an actual congestion signal and when it sees the subsequent Congestion Exposure Signal from the sender. The minimum delay will be one round trip, but it may be much longer depending on the transport's choice of feedback delay (consider RTCP [RFC3550] for example). It is not practical to expect auditing devices in the network to make allowance for

such feedback delays. Instead, the sender SHOULD be able to send ConEx signals in advance, as 'credit' for any audit function to hold as a balance against the risk of congestion during the feedback delay. This design choice greatly simplifies auditing (see Section 4.3.1).

It is important to note that the auditing requirement implies a number of additional constraints: The basic auditing technique is to count both actual congestion signals and ConEx Signals someplace along the data path:

- o For congestion signaled by ECN, auditing is most accurate when located near the transport receiver. Within any flow or aggregate of flows, the volume of data tagged with ConEx Signals should never be less than the total volume of ECN marked data seen near the receiver.
- o 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 necessarily see the transport protocol sequence numbers and when the missing packets might simply be taking a different route. But there are common cases where sufficient audit accuracy should be possible:
  - * For non-IPsec traffic conforming to standard TCP sequence numbering on a single path, an auditor could detect losses by observing both the original transmission and the retransmission after the loss. Such auditing would be most accurate near the sender.
  - * For networks designed so that losses predominantly occur under the management of one IP-aware node on the path, the auditor could be located at this bottleneck. It could simply compare ConEx Signals with actual local losses. This is a good model for most consumer access networks where audit accuracy could well be sufficient even if losses occasionally occur at other nodes in the network, such as border gateways (see Section 4.3 for details).

Given that loss-based and ECN-based ConEx might sometimes be best audited at different locations, having distinct encodings would widen the design space for the auditing function.

3. Representing 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 the ConEx Signal is likely to entail some engineering compromises that have the potential to reduce the protocol's usefulness in some settings. Rather than making these engineering choices prematurely,

this document side steps the encoding problem by describing an abstract representation of ConEx Signals. All of the elements of the protocol can be defined in terms of this abstract representation. Most important, the preliminary use cases for the protocol are described in terms of the abstract representation in companion documents [I-D.conex-concepts-uses].

Once we have some example use cases we can evaluate different encoding schemes. Since these schemes are likely to include some conflated code points, some information will be lost resulting in weakening or disabling some of the algorithms and eliminating some use cases.

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.

3.1. Strawman Encoding

As an aid to the reader, it might be helpful to describe a naive strawman encoding of the ConEx protocol described solely in terms of TCP: set the Reserved bit in the IPv4 header (bit 48 counting from zero [RFC0791]--aka the "evil bit" [RFC3514]) on all retransmissions or once per ECN signaled window reduction. Clearly network devices along the forward path can see this bit and act on it. For example they can count marked and unmarked packets to estimate the congestion levels along the path.

However, the IESG has chartered the ConEx working group to establish that there is sufficient demand for an IPv6 ConEx protocol before using the last available bit in the IPv4 header. Furthermore this encoding, by itself, does not sufficiently support partial deployment or strong auditing and might motivate users and/or applications to misrepresent the congestion that they are causing.

Nonetheless, this strawman encoding does present a clear mental model of how the ConEx protocol might function under various uses.

3.2. ECN Based Encoding

Ideally ConEx and ECN are orthogonal signals and SHOULD be entirely independent. However, given the limited number of header bit and/or code points, these signals may have to share code points, at least partially.

The re-ECN specification [I-D.briscoe-tsvwg-re-ecn-tcp] presents an implementation of ConEx that had to be tightly integrated with the

encoding of ECN in order to fit into the IP header. The central theme of the re-ECN work is an audit mechanism that can provide sufficient disincentives against misrepresenting congestion [I-D.briscoe-tsvwg-re-ecn-motiv], which is analyzed extensively in Briscoe's PhD dissertation [Refb-dis].

Re-ECN is a good example of one chosen set of compromises attempting to meet the requirements of Section 2. However, the present document takes a step back, aiming to state the ideal requirements in order to allow the Internet community to assess whether other compromises are possible.

In particular, different incremental deployment choices may be desirable to meet the partial deployment requirement of Section 2. Re-ECN requires the receiver to be at least ECN-capable as well as requiring an update to the sender. Although ConEx will inherently require change at the sender, it would be preferable if it could work, even partially, with any receiver.

The chosen ConEx protocol certainly must not require ECN to be deployed in any network. In this respect re-ECN is already a good example--it acts perfectly well as a loss-based ConEx protocol it the loss-based audit techniques in Section 4.3 are used. However, it would still be desirable to avoid the dependence on an ECN receiver.

For a tutorial background on re-ECN techniques, see [Re-fb, FairerFaster].

3.2.1. ECN Changes

Although the re-ECN protocol requires no changes to the network part of the ECN protocol, it is important to note that it does propose some relatively minor modifications to the host-to-host aspects of the ECN protocol specified in RFC 3168. They include: redefining the ECT(1) code point (the change is consistent with RFC3168 but requires deprecating the experimental ECN nonce [RFC3540]); modifications to the ECN negotiations carried on the SYN and SYN-ACK; and using a different state machine to carry ECN signals in the transport acknowledgments from a modified Receiver to the Sender. This last change is optional, but it permits the transport protocol to carry multiple congestion signals per round trip. It greatly simplifies accurate auditing, and is likely to be useful in other transports, e.g. DCTCP [DCTCP].

All of these adjustments to RFC 3168 may also be needed in a future standardized ConEx protocol. There will need to be very careful consideration of any proposed changes to ECN or other existing protocols, because any such changes increase the cost of deployment.

3.3. Abstract Encoding

The ConEx protocol could take one of two different encodings: independently settable bits or an enumerated set of mutually exclusive codepoints.

In both cases, the amount of congestion is signaled by the volume of marked data--just as the volume of lost data or ECN marked data signals the amount of congestion experienced. Thus the size of each packet carrying a ConEx Signal is significant.

3.3.1. 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)

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 4.3 on the audit function)

3.3.2. 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 (Section 3.3.1). 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.

4. Congestion Exposure Components

{ToDo: Picture of the components, similar to that in the last slideset about conex-concepts-uses?}

4.1. Modified Senders

The sending transport needs to be modified to send Congestion Exposure Signals in response to congestion feedback signals.

4.2. Receivers (Optionally Modified)

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

However, a TCP receiver feeds back ECN congestion signals no more than once within a round trip. The sender may require more precise feedback from the receiver otherwise it will appear to be understating its ConEx Signals (see Section 3.2.1).

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. This was the approach taken when adding re-ECN to TCP [I-D.briscoe-tsvwg-re-ecn-tcp].

4.3. Audit

To audit ConEx Signals against actual losses (as opposed to ECN) an auditor could use one of the following techniques:

TCP-specific approach: The auditor could monitor TCP flows or aggregates of flows, only holding state on a flow if it first sends a Credit or a Re-Echo-Loss marking. The auditor could detect retransmissions by monitoring sequence numbers. It would assure that (volume of retransmitted data) <= (volume of data marked Re-Echo-Loss). Traffic would only be auditable in this way if it conformed to the standard TCP protocol and the IP payload was not encrypted (e.g. with IPsec).

Predominant bottleneck approach: Unlike the above TCP-specific solution, this technique would work for IP packets carrying any transport layer protocol, and whether encrypted or not. But it only works well for networks designed so that losses predominantly occur under the management of one IP-aware node on the path. The auditor could then be located at this bottleneck. It could simply compare ConEx Signals with actual local losses. Most consumer access networks are design to this model, e.g. the radio network controller (RNC) in a cellular network or the broadband remote access server (BRAS) in a digital subscriber line (DSL) network.

The accuracy of an auditor at one predominant bottleneck might still be sufficient, even if losses occasionally occurred at other nodes in the network (e.g. border gateways). Although the auditor at the predominant bottleneck would not always be able to detect losses at other nodes, transports would not know where losses were occurring either. Therefore a transport would not know which losses it could cheat on without getting caught, and which ones it couldn't.

To audit ConEx Signals against actual ECN markings or losses, the auditor could work as follows: monitor flows or aggregates of flows, only holding state on a flow if it first sends a ConEx-Marked packet (Credit or either Re-Echo marking). Count the number of bytes marked with Credit or Re-Echo-ECN. Separately count the number of bytes marked with ECN. Use Credits to assure that {#ECN} <= {#Re-Echo-ECN} + {#Credit}, even though the Re-Echo-ECN markings are delayed by at least one RTT.

4.3.1. 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 it makes the two passes of the feedback loop 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 it is hard for a network device to know the RTT of each transport.

Instead, it considerably simplifies the audit function if the source transport is made responsible for removing the round trip delay in ConEx signals. The transport SHOULD signal sufficient credit in advance to cover any reasonably expected congestion 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.

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 ever the balance does go negative, the audit function can immediately start punishing a flow, without any grace period.

The one-way nature of packet forwarding probably makes per-flow state unavoidable for the audit function. This was a necessary sacrifice

to avoid per-flow state elsewhere in the wider ConEx architecture. Nonetheless, care was taken to ensure that packets could bring soft-state to the audit function, so that it would continue to work if a flow shifted to a different audit device, perhaps after a reroute or an audit device failure. Therefore, although the audit function is likely to need flow state memory, at least it complies with the 'fate-sharing' design principle of the Internet [IntDesPrinciples], and at least per-flow audit is only required at the outer edges of the internetwork, where it is less of a scalability concern.

Note also that ConEx does not intend to embed rules in the network on how individual flows _behave_. The audit function only does per-flow processing to check the integrity of ConEx _information_.

4.3.2. Behaviour Constraints for the Audit Function

There is no intention to standardise how to design or implement 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, preferably at the first dishonest packet;

Transport Oblivious: Audit MUST 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 [RFC3448]. 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 downstream networks;

Sufficient Sanction: Audit MUST introduce sufficient sanction (e.g. loss in goodput) so that sources cannot understate congestion and play off losses at the audit function against higher allowed throughput at a congestion policer [Salvatori05];

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 sources to exhaust its memory capacity by gratuitously sending numerous packets, each with a different flow ID.

Identifier Accountability: Audit MUST 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];

4.4. 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, the auditing devices referred to in the previous section primarily enforce compliance with the ConEx protocol and do not need to be configured with any client-specific policy.

4.4.1. Policy 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 next section), 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 between networks based on congestion, or merely send warnings to clients who are causing excessive congestion.

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

4.4.2. 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 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, it measures the amount of congestion that this traffic is contributing to somewhere downstream. If this exceeds a policy-configured 'congestion-bit-rate' the congestion policer will limit all the monitored ConEx traffic.

A congestion policer can be implemented by a simple token bucket. But unlike a bit-rate policer, it removes a token only when it forwards a packet that is 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, which should be

sufficient traffic management without having to additionally constrain the straight bit-rate. See [CongPol] for details.

5. IANA Considerations

This memo includes no request to IANA.

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

6. Security Considerations

Significant parts of this whole document are about auditability of ConEx Signals, in particular Section 4.3.

7. Conclusions

{ToDo:}

8. Acknowledgements

This document was improved by review comments from Toby Moncaster, Nandita Dukkipati, Mirja Kuehlewind and Caitlin Bestler.

9. 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.

10. References

10.1. Normative References

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

10.2. 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.

[CongPol] Jacquet, A., Briscoe, B., and T. Moncaster, "Policing Freedom to Use

the Internet Resource Pool", Proc ACM Workshop on Re-Architecting the Internet (ReArch'08) , December 2008, <http://bobbriscoe.net/projects/refb/#polfree>.

[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-tsvwg-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-tsvwg-re-ecn-tcp-motivation-02 (work in progress), October 2010.

[I-D.briscoe-tsvwg-re-ecn-tcp] Briscoe, B., Jacquet, A., Moncaster, T., and A. Smith, "Re-ECN: Adding Accountability for Causing Congestion to TCP/IP", draft-briscoe-tsvwg-re-ecn-tcp-09 (work in progress), October 2010.

[I-D.conex-concepts-uses] Briscoe, B., Woundy, R., Moncaster, T., and J. Leslie, "ConEx Concepts

and Use Cases", draft-ietf-conex-concepts-uses-01 (work in progress), March 2011.

[I-D.ietf-ledbat-congestion] Shalunov, S., Hazel, G., and J. Iyengar, "Low Extra Delay Background Transport (LEDBAT)", draft-ietf-ledbat-congestion-03 (work in progress), October 2010.

[I-D.sridharan-tcpm-ctcp] Sridharan, M., Tan, K., Bansal, D., and D. Thaler, "Compound TCP: A New TCP Congestion Control for High-Speed and Long Distance Networks", draft-sridharan-tcpm-ctcp-02 (work in progress), November 2008.

[IntDesPrinciples] Clark, D., "The Design Philosophy of the DARPA Internet Protocols", ACM SIGCOMM CCR 18(4)106--114, August 1988, <http://www.acm.org/sigcomm/ccr/archive/1995/jan95/ccr-9501-clark.pdf>.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

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

[RFC3448] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003.

[RFC3514] Bellovin, S., "The Security Flag in the IPv4 Header", RFC 3514, April 1 2003.

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

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

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

[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.

[Vegas] Brakmo, L. and L. Peterson, "TCP Vegas: End-to-End Congestion

Avoidance on a Global Internet",
IEEE Journal on Selected Areas in
Communications 13(8)1465--80,
October 1995, <http://
ieeexplore.ieee.org/iel1/49/9740/
00464716.pdf?arnumber=464716>.

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/