

ConEx
Internet-Draft
Intended status: Informational
Expires: January 10, 2013

B. Briscoe
BT
M. Sridharan
Microsoft
July 09, 2012

Network Performance Isolation in Data Centres using Congestion Exposure
(ConEx)
draft-briscoe-conex-data-centre-00

Abstract

This document describes how a multi-tenant data centre operator can isolate tenants from network performance degradation due to each other's usage, but without losing the multiplexing benefits of a LAN-style network where anyone can use any amount of any resource. Zero per-tenant configuration and no implementation change is required on network equipment. Instead the solution is implemented with a simple change to the hypervisor (or container) on each physical server, beneath the tenant's virtual machines. These collectively enforce a very simple distributed contract - a single network allowance that each tenant can allocate among their virtual machines. The solution is simplest and most efficient using layer-3 switches that support explicit congestion notification (ECN) and if the sending operating system supports congestion exposure (ConEx). Nonetheless, an arrangement is described so that the operator can unilaterally deploy a complete solution while operating systems are being incrementally upgraded to support ConEx.

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 10, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Design Features	4
3. Outline Design	7
4. Performance Isolation: Intuition	8
4.1. Simple Boundary Model of Congestion Control	9
4.2. Long-Running Flows	10
4.3. On-Off Flows	12
4.3.1. Numerical Examples Without Policing	14
4.3.2. Congestion Policing of On-Off Flows	17
4.4. Weighted Congestion Controls	18
4.5. A Network of Links	20
4.6. Links of Different Sizes	22
4.7. Diverse Congestion Control Algorithms	23
5. Design	24
6. Parameter Setting	26
7. Incremental Deployment	26
7.1. Migration	26
7.2. Evolution	27
8. Related Approaches	27
9. Security Considerations	28
10. IANA Considerations	28
11. Conclusions	28
12. Acknowledgments	28
13. Informative References	29
Appendix A. Summary of Changes between Drafts	30

1. Introduction

A number of companies offer hosting of virtual machines on their data centre infrastructure--so-called infrastructure as a service (IaaS). A set amount of processing power, memory, storage and network are offered. Although processing power, memory and storage are relatively simple to allocate on the 'pay as you go' basis that has become common, the network is less easy to allocate given it is a naturally distributed system.

This document describes how a data centre infrastructure provider can deploy congestion policing at every ingress to the data centre network, e.g. in all the hypervisors (or containers) in a data centre that provides virtualised 'cloud' computing facilities. These bulk congestion policers pick up congestion information in the data packets traversing the network, using one of two approaches: feedback tunnels or ConEx. Then, these policers at the ingress edge have sufficient information to limit the amount of congestion any tenant can cause anywhere in the data centre. This isolates the network performance experienced by each tenant from the behaviour of all the others, without any tenant-related configuration of any of the switches.

The key to the solution is the use of congestion-bit-rate rather than bit-rate as the policing metric. How this works is very simple and quick to describe (Section 3 outlines the design at the start and Section 5 gives details).

However, it is much more difficult to understand why this approach provides performance isolation. In particular, why it provides performance isolation across a network of links, even though there is apparently no isolation mechanism in each link. Section 4 builds up an intuition for why the approach works, and why other approaches fall down in different ways. The explanation builds as follows:

- o Starting with the simple case of long-running flows focused any one bottleneck link in the network, tenants get weighted shares of the link, much like weighted round robin, but with no mechanism in any of the links;
- o In the more realistic case where flows are not all long-running but a mix of short to very long, it is explained that bit-rate is not a sufficient metric for isolating performance; how often a tenant is not sending is the significant factor for performance isolation, not whether bit-rate is shared equally whenever it is sending;

- o Although it might seem that data volume would be a good measure of how often a tenant does not send, we then show that a tenant can send a large volume of data but hardly affect the performance of others -- by being very responsive to congestion. Using congestion-volume (congestion-bit-rate over time) in a policer encourages large data senders to give other tenants much higher performance, whereas using straight volume as an allocation metric provides no isolation at all from tenants who send the same volume but are oblivious to its effect on others (the widespread behaviour today);
- o We then show that a policer based on the congestion-bit-rate metric works across a network of links treating it as a pool of capacity, whereas other approaches treat each link independently, which is why the proposed approach requires none of the configuration complexity on switches that is involved in other approaches.

The solution would also be just as applicable to isolate the network performance of different departments within the data centre of an enterprise, which could be implemented without virtualisation. However, it will be described as a multi-tenant scenario, which is the more difficult case from a security point of view.

{ToDo: Meshed, pref multipath resource pool, not unnecessarily constrained paths.}

2. Design Features

The following goals are met by the design, each of which is explained subsequently:

- o Performance isolation
- o No loss of LAN-like openness and multiplexing benefits
- o Zero tenant-related switch configuration
- o No change to existing switch implementations
- o Weighted performance differentiation
- o Ultra-Simple contract--per-tenant network-wide allowance
- o Sender constraint, but with transferrable allowance
- o Transport-agnostic

- o Extensible to wide-area and inter-data-centre interconnection

Performance Isolation with Openness of a LAN: The primary goal is to ensure that each tenant of a data centre receives a minimum assured performance from the whole network resource pool, but without losing the efficiency savings from multiplexed use of shared infrastructure (work-conserving). There is no need for partitioning or reservation of network resources.

Zero Tenant-Related Switch Configuration: Performance isolation is achieved with no per-tenant configuration of switches. All switch resources are potentially available to all tenants.

Separately, `_forwarding_` isolation may (or may not) be configured to ensure one tenant cannot receive traffic from another's virtual network. However, `_performance_` isolation is kept completely orthogonal, and adds nothing to the configuration complexity of the network.

No New Switch Implementation: Straightforward commodity switches (or routers) are sufficient. Bulk explicit congestion notification (ECN) is recommended, which is available in a large and growing range of layer-3 switches (a layer-3 switch does switching at layer-2, but it can use the Diffserv and ECN fields for traffic control if an IP header can be found). Once the network supports ECN, the performance isolation function is confined to the hypervisor (or container) and the operating systems on the hosts.

Weighted Performance Differentiation: A tenant gets network performance in proportion to their allowance when constrained by others, with no constraint otherwise. Importantly, the assurance is not just instantaneous, but over time. And the assurance is not just localised to each link but network-wide. This will be explained with numerical examples later.

Ultra-Simple Contract: The tenant needs to decide only two things: The peak bit-rate connecting each virtual machine to the network (as today) and an overall 'usage' allowance. This document focuses on the latter. A tenant just decides one number for her contracted allowance that can be shared over all her virtual machines (VMs). The 'usage' allowance is a measure of congestion-bit-rate, which will be explained later, but most tenants will just think of it as a number, where more is better. A tenant has no need to decide in advance which VMs will need more allowance and which less--an automated process allocates the allowance across the VMs, shifting more to those that need it most, as they use it. Therefore, performance cannot be constrained by poor choice of allocations between VMs, removing a whole dimension from

the problem that tenants face when choosing their traffic contract. The allocation process can be operated by the tenant, or provided by the data centre operator as part of an additional platform as a service (PaaS) offer.

Sender Constraint with transferrable allowance: By default, constraints are always placed on data senders, determined by the sending party's traffic contract. Nonetheless, if the receiving party (or any other party) wishes to enhance performance it can arrange this with the sender at the expense of its own allowance.

For instance, when a tenant's VM sends data to a storage facility the tenant that owns the VM consumes her allowance for enhanced sending performance. But by default when she later retrieves data from storage, the storage facility is the sender, so the storage facility consumes its allowance to determine performance in the reverse direction. Nonetheless, during the retrieval request, the storage facility can require that its sending 'costs' are covered by the receiving VM's allowance.

Transport-Agnostic: In a well-provisioned network, enforcement of performance isolation rarely introduces constraints on network behaviour. However, it continually counts how much each tenant is limiting the performance of others, and it will intervene to enforce performance isolation, but against only those customers who most persistently constrain others. This performance isolation is oblivious to flows and to the protocols and algorithms being used above the IP layer.

Interconnection: The solution is designed so that interconnected networks can ensure each is accountable for the performance degradation it contributes to in other networks. If necessary, one network has the information to intervene at its ingress to limit traffic from another network that is degrading performance. Alternatively, with the proposed protocols, networks can see sufficient information in traffic arriving at their borders to give their neighbours financial incentives to limit the traffic themselves.

The present document focuses on a single provider-scenario, but evolution to interconnection with other data centres over wide-area networks, and interconnection with access networks is briefly discussed in Section 7.2.

3. Outline Design

This section outlines the essential features of the design. Design details will be given in Section 5.

Edge policing: Traffic policing is located at the policy enforcement point where each sending host connects to the network, typically beneath the tenant's operating system in the hypervisor controlled by the infrastructure operator. In this respect, the approach has a similar arrangement to the Diffserv architecture with traffic policers forming a ring around the network [RFC2475].

Congestion policing: However, unlike Diffserv, traffic policing limits congestion-bit-rate, not bit-rate. Congestion bit-rate is the product of congestion probability and bit-rate. For instance, if the instantaneous congestion probability (cf. loss probability) across a network path were 0.02% and a tenant's maximum contracted congestion-bit-rate was 600kb/s, then the policer would allow the tenant to send at a bit-rate of up to 3Gb/s (because $3\text{Gb/s} \times 0.02\% = 600\text{kb/s}$). The detail design section describes how congestion policers at the network ingress know the congestion that each packet will encounter in the network. {ToDo: rewrite this section to describe how a congestion policer works, not to focus just on units.}

Hose model: The congestion policer controls all traffic from a particular sender without regard to destination, similar to the Diffserv 'hose' model. {ToDo: dual policer, and multiple hoses for long-term average.}

Flow policing unnecessary: A congestion policer could be designed to focus policing on the particular data flow(s) contributing most to the excess congestion-bit-rate. However we will explain why bulk policing should be sufficient.

FIFO forwarding: Each network queue only needs a first-in first-out discipline, with no need for any priority scheduling. If scheduling by traffic class is used (for whatever reason), congestion policing can be used to isolate tenants from each other within each class. {ToDo: Say this the other way round.}

ECN marking recommended: All queues that might become congested should support bulk ECN marking, but packets that do not support ECN marking can be accommodated.

In the proposed approach, the network operator deploys capacity as usual--using previous experience to determine a reasonable contention ratio at every tier of the network. Then, the tenant contracts with

the operator for an allowance that determines the rate at which the congestion policer allows each tenant to contribute to congestion {ToDo: Dual policer}. Section 6 discusses how the operator would determine this allowance. Each VM's congestion policer limits its peak congestion-bit-rate as well as limiting the overall average per tenant.

4. Performance Isolation: Intuition

Network performance isolation traditionally meant that each user could be sure of a minimum guaranteed bit-rate. Such assurances are useful if traffic from each tenant follows relatively predictable paths and is fairly constant. If traffic demand is more dynamic and unpredictable (both over time and across paths), minimum bit-rate assurances can still be given, but they have to be very small relative to the available capacity.

This either means the shared capacity has to be greatly overprovided so that the assured level is large enough, or the assured level has to be small. The former is unnecessarily expensive; the latter doesn't really give a sufficiently useful assurance.

Another form of isolation is to guarantee that each user will get $1/N$ of the capacity of each link, where N is the number of active users at each link. This is fine if the number of active users (N) sharing a link is fairly predictable. However, if large numbers of tenants do not typically share any one link but at any time they all could (as in a data centre), a $1/N$ assurance is fairly worthless. Again, given N is typically small but could be very large, either the shared capacity has to be expensively overprovided, or the assured bit-rate has to be worthlessly small.

Both these traditional forms of isolation try to give the tenant an assurance about instantaneous bit-rate by constraining the instantaneous bit-rate of everyone else. However, there are two mistakes in this approach. The amount of capacity left for a tenant to transfer data as quickly as possible depends on:

1. the load over time of everyone else
2. how much everyone else yields to the increase in congestion when someone else tries to transfer data

This is why limiting congestion-bit-rate over time is the key to network performance isolation. It focuses policing only on those tenants who go fast over congested path(s) excessively and persistently over time. This keeps congestion below a design threshold everywhere so that everyone else can go fast.

Congestion policing can and will enforce a congestion response if a particular tenant sends traffic that is completely unresponsive to congestion. However, the purpose of congestion policing is not to intervene in everyone's rate control all the time. Rather it is encourage each tenant to avoid being policed -- to keep the aggregate of all their flows' responses to congestion within an overall envelope. Nonetheless, the upper bound set by the congestion policer still ensures that each tenant's minimum performance is isolated from the combined effect of everyone else.

It has not been easy to find a way to give the intuition on why congestion policing isolates performance, particularly across a networks of links not just on a single link. The approach used in this section, is to describe the system as if everyone is using the congestion response they would be forced to use if congestion policing had to intervene. We therefore call this the boundary model of congestion control. It is a very simple congestion response, so it is much easier to understand than if we introduced all the square root terms and other complexity of New Reno TCP's response. And it means we don't have to try to describe a mix of responses.

We cannot emphasise enough that the intention is not to make individual flows conform to this boundary response to congestion. Indeed the intention is to allow a diverse evolving mix of congestion responses, but constrained in total within a simple overall envelope.

After describing and further justifying using the a simple boundary model of congestion control, we start by considering long-running flows sharing one link. Then we will consider on-off traffic, before widening the scope from one link to a network of links and to links of different sizes. Then we will depart from the initial simplified model of congestion control and consider diverse congestion control algorithms, including no end-system response at all.

Formal analysis to back-up the intuition provided by this section will be made available in a more extensive companion technical report [conex-dc_tr].

4.1. Simple Boundary Model of Congestion Control

The boundary model of congestion control ensures a flow's bit-rate is inversely proportional to the congestion level that it detects. For instance, if congestion probability doubles, the flow's bit-rate halves. This is called a scalable congestion control because it maintains the same rate of congestion signals (marked or dropped packets) no matter how fast it goes. Examples are Relentless TCP and Scalable TCP [ToDo: add refs].

New Reno-like TCP algorithms [RFC5681] have been widely replaced by alternatives closer to this scalable ideal (e.g. Cubic TCP, Compound TCP [ToDo: add refs]), because at high rates New Reno generated congestion signals too infrequently to track available capacity fast enough [RFC3649]. More recent TCP updates (e.g. data centre TCP) are becoming closer still to the scalable ideal.

It is necessary to carefully distinguish congestion bit-rate, which is an absolute measure of the rate of congested bits vs. congestion probability, which is a relative measure of the proportion of congested bits to all bits. For instance, consider a scenario where a flow with scalable congestion control is alone in a 1Gb/s link, then another similar flow from another tenant joins it. Both will push up the congestion probability, which will push down their rates until they together fit into the link. Because the flow's rate has to halve to accomodate the new flow, congestion probability will double (lets say from 0.002% to 0.004%), by our initial assumption of a scalable congestion control. When it is alone on the link, the congestion-bit-rate of the flow is 20kb/s ($= 1\text{Gb/s} * 0.002\%$), and when it shares the link it is still 20kb/s ($= 500\text{Mb/s} * 0.04\%$).

In summary, a congestion control can be considered scalable if the bit-rate of packets carrying congestion signals (the congestion-bit-rate) always stays the same no matter how much capacity it finds available. This ensures there will always be enough signals in a round trip time to keep the dynamics under control.

Reminder: Making individual flows conform to this boundary or scalable response to congestion is a non-goal. Although we start this explanation with this specific simple end-system congestion response, this is just to aid intuition.

4.2. Long-Running Flows

Table 1 shows various scenarios where each of five tenants has contracted for 400kb/s of congestion-bit-rate in order to share a 1Gb/s link. In order to help intuition, we start with the (unlikely) scenario where all their flows are long-running. Long-running flows will try to use all the link capacity, so for simplicity we take utilisation as a round 100%.

In the case we have just described (scenario A) neither tenant's policer is intervening at all, because both their congestion allowances are 40kb/s and each sends only one flow that contributes 20kb/s of congestion -- half the allowance.

Tenant	contracted congestion- bit-rate kb/s	scenario A # : Mb/s	scenario B # : Mb/s	scenario C # : Mb/s	scenario D # : Mb/s
(a)	40	1 : 500	5 : 250	5 : 200	5 : 250
(b)	40	1 : 500	3 : 250	3 : 200	2 : 250
(c)	40	- : ---	3 : 250	3 : 200	2 : 250
(d)	40	- : ---	2 : 250	2 : 200	1 : 125
(e)	40	- : ---	- : ---	2 : 200	1 : 125
	Congestion probability	0.004%	0.016%	0.02%	0.016%

Table 1: Bit-rates that a congestion policer allocates to five tenants sharing a 1Gb/s link with various numbers (#) of long-running flows all using 'scalable congestion control'

Scenario B shows a case where four of the tenants all send 2 or more long-running flows. Recall that each flow always contributes 20kb/s no matter how fast it goes. Therefore the policers of tenants (a-c) limit them to two flows-worth of congestion ($2 \times 20\text{kb/s} = 40\text{kb/s}$). Tenant (d) is only asking for 2 flows, so it gets them without being policed, and all four get the same quarter share of the link.

Scenario C is similar, except the fifth tenant (e) joins in, so they all get equal $1/5$ shares of the link.

In Scenario D, only tenant (a) asks for more than two flows, so (a)'s policer limits it to two flows-worth of congestion, and everyone else gets the number of flows-worth that they ask for. This means that tenants (d&e) get less than everyone else, because they asked for less than they would have been allowed. (Similarly, in Scenarios A & B, some of the tenants are inactive, so they get zero, which is also less than they could have had if they had wanted.)

With lots of long-running flows, as in scenarios B & C, congestion policing seems to emulate round robin scheduling, equalising the bit-rate of each tenant, no matter how many flows they run. By configuring different contracted allowances for each tenant, it can easily be seen that congestion policing could emulate weighted round robin (WRR), with the relative sizes of the allowances acting as the weights.

Scenario D departs from round-robin. This is deliberate, the idea being that tenants are free to take less than their share in the short term, which allows them to take more at other times, as we will

see in Section 4.4. In Scenario D, policing focuses only on the tenant (a) that is continually exceeding its contract. This policer focuses discard solely on tenant a's traffic so that it cannot cause any more congestion at the shared link (shown as 0.016% in the last row).

To summarise so far, ingress congestion policers control congestion-bit-rate in order to indirectly assure a minimum bit-rate per tenant. With lots of long-running flows, the outcome is somewhat similar to WRR, but without the need for any mechanism in each queue.

4.3. On-Off Flows

Aiming to behave like round-robin (or weighted round-robin) is only useful when all flows are infinitely long. For transfers of finite size, congestion policing isolates one tenant's performance from the behaviour of others -- unlike WRR would, as will now be explained.

Figure 1 compares two example scenarios where tenant 'b' regularly sends small files in the top chart and the same size files but more often in the bottom chart (a higher 'on-off ratio'). This is the typical behaviour of a Web server when more clients request more files at peak time. Meanwhile, in this example, tenant c's behaviour doesn't change between the two scenarios -- it sends a couple of large files, each starting at the same time in both cases.

The capacity of the link that 'b' and 'c' share is shown as the full height of the plot. The files sent by 'b' are shown as little rectangles. 'b' can go at the full bit-rate of the link when 'c' is not sending, which is represented by the tall thin rectangles labelled 'b' near the middle. We assume for simplicity that 'b' and 'c' divide up the bit-rate equally. So, when both 'b' and 'c' are sending, the 'b' rectangles are half the height (bit-rate) and twice the duration relative to when 'b' sends alone. The area of a file to be transferred stays the same, whether tall and thin or short and fat, because the area represents the size of the file (bit-rate x duration = file size). The files from 'c' look like inverted castellations, because 'c' uses half the link rate while each file from 'b' completes, then 'c' can fill the link until 'b' starts the next file. The cross-hatched areas represent idle times when no-one is sending.

For this simple scenario we ignore start-up dynamics and just focus on the rate and duration of flows that are long enough to stabilise, which is why they can be represented as simple rectangles. We will introduce the effect of flow startups later.

In the bottom case, where 'b' sends more often, the gaps between b's

transfers are smaller, so 'c' has less opportunity to use the whole line rate. This squeezes out the time it takes for 'c' to complete its file transfers (recall a file will always have the same area which represents its size). Although 'c' finishes later, it still starts the next flow at the same time. In turn, this means 'c' is sending during a greater proportion of b's transfers, which extends b's average completion time too.

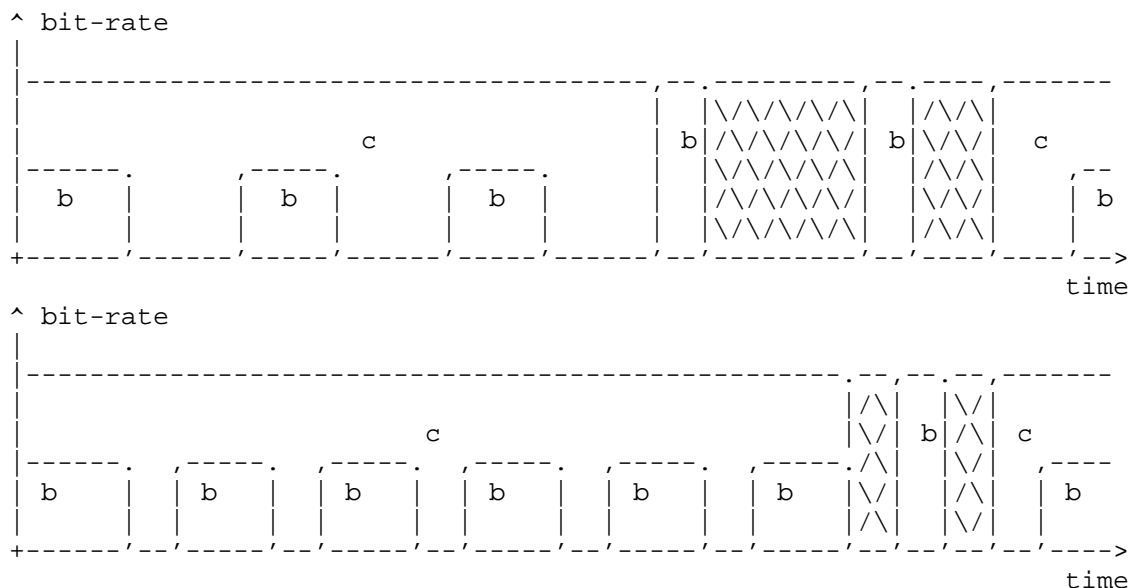


Figure 1: In the lower case, the on-off ratio of 'b' has increased, which extends all the completion times of 'c' and 'b'

Round-robin would do little if anything to isolate 'c' from the effect of 'b' sending files more often. Round-robin is designed to force 'b' and 'c' to share the capacity equally when they are both active. But in both scenarios they already share capacity equally when they are both active. The difference is in how often they are active. Round-robin and other traditional fair queuing techniques don't have any memory to sense that 'b' has been active more of the time.

In contrast, a congestion policer can tell when one tenant is sending files more frequently, by measuring the rate at which the tenant is contributing to congestion. Our aim is to show that policers will be able to isolate performance properly by using the right metric (congestion bit-rate), rather than using the wrong metric (bit-rate), which doesn't sense whether the load over time is large or small.

4.3.1. Numerical Examples Without Policing

The usefulness of the congestion bit-rate metric will now be illustrated with the numerical examples in Table 2. The scenarios illustrate what the congestion bit-rate would be without any policing or scheduling action in the network. Then this metric can be monitored and limited by a policer, to prevent one tenant from harming the performance of others.

The 2nd & 3rd columns (file-size and inter-arrival time) fully represent the behaviour of each tenant in each scenario. All the other columns merely characterise the outcome in various ways. The inter-arrival time (T) is the average time between starting one file and the next. For instance, tenant 'b' sends a 16Mb file every 200ms on average. The formula in the heading of some columns shows how the column was derived from other columns.

Scenario E is contrived so that the three tenants all offer the same load to the network, even though they send files of very different size (S). The files sent by tenant 'a' are 100 times smaller than those of tenant 'b', but 'a' sends them 100 times more often. In turn, b's files are 100 times smaller than c's, but 'b' in turn sends them 100 times more often. Graphically, the scenario would look similar to Figure 1, except with three sizes of file, not just two. Scenarios E-G are designed to roughly represent various distributions of file sizes found in data centres, but still to be simple enough to facilitate intuition, even though each tenant would not normally send just one size file.

The average completion time (t) and the maximum were calculated from a fairly simple analytical model (documented in a companion technical report [conex-dc_tr]). Using one data point as an example, it can be seen that a 1600Mb (200MB) file from tenant 'c' completes in 1905ms (about 1.9s). The files that are 100 times smaller complete 100 times more quickly on average. In fact, in this scenario with equal loads, each tenant perceives that their files are being transferred at the same rate of 840Mb/s on average (file-size divided by completion time, as shown in the apparent bit-rate column). Thus on average all three tenants perceive they are getting 84% of the 1Gb/s link on average (due to the benefit of multiplexing and utilisation being low at 240Mb/s / 1Gb/s = 24% in this case).

The completion times of the smaller files vary significantly, depending on whether a larger file transfer is proceeding at the same time. We have already seen this effect in Figure 1, where, when tenant b's files share with 'c', they take twice as long to complete as when they don't. This is why the maximum completion time is greater than the average for the small files, whereas there is

imperceptible variance for the largest files.

The final column shows how congestion bit-rate will be a useful metric to enforce performance isolation (the figures illustrate the situation before any enforcement mechanism is added). In the case of equal loads (scenario E), average congestion bit-rates are all equal. In scenarios F and G average congestion bit-rates are higher, because all tenants are placing much more load on the network over time, even though each still sends at equal rates to others when they are active together. Figure 1 illustrated a similar effect in the difference between the top and bottom scenarios.

The maximum instantaneous congestion bit-rate is nearly always 20kb/s. That is because, by definition, all the tenants are using scalable congestion controls with a constant congestion rate of 20kb/s. As we saw in Section 4.1, the congestion rate of a particular scalable congestion control is always the same, no matter how many other flows it competes with.

Once it is understood that the congestion bit-rate of one scalable flow is always 'w' and doesn't change whenever a flow is active, it becomes clear what the congestion bit-rate will be when averaged over time; it will simply be 'w' multiplied by the proportion of time that the tenant's file transfers are active. That is, $w \cdot t / T$. For instance, in scenario E, on average tenant b's flows start 200ms apart, but they complete in 19ms. So they are active for $19/200 = 10\%$ of the time (rounded). A tenant that causes a congestion bit-rate of 20kb/s for 10% of the time will have an average congestion-bit-rate of 2kb/s, as shown.

To summarise so far, no matter how many more files transfer at the same time, each scalable flow still contributes to congestion at the same rate, but it contributes for more of the time, because it squeezes out into the gap before its next flow starts.

Tenant	File size	Ave. inter-arrival	Ave. load	Completion time	Apparent bit-rate	Congestion bit-rate
	S	T	S/T	ave : max t	ave : min S/t	ave : max w*t/T
	Mb	ms	Mb/s	ms	Mb/s	kb/s
Scenario E						
a	0.16	2	80	0.19 : 0.48	840 : 333	2 : 20
b	16	200	80	19 : 35	840 : 460	2 : 20
c	1600	20000	80	1905 : 1905	840 : 840	2 : 20
			240			
Scenario F						
a	0.16	0.67	240	0.31 : 0.48	516 : 333	9 : 20
b	16	50	320	29 : 42	557 : 380	11 : 20
c	1600	10000	160	3636 : 3636	440 : 440	7 : 20
			720			
Scenario G						
a	0.16	0.67	240	0.33 : 0.64	481 : 250	10 : 20
b	16	40	400	32 : 46	505 : 345	16 : 40
c	1600	10000	160	4543 : 4543	352 : 352	9 : 20
			800			

Single link of capacity 1Gb/s. Each tenant uses a scalable congestion control which contributes a congestion-bit-rate for each flow of $w = 20\text{kb/s}$.

Table 2: How the effect on others of various file-transfer behaviours can be measured by the resulting congestion-bit-rate

In scenario F, clients have increased the rate they request files from tenants a, b and c respectively by 3x, 4x and 2x relative to scenario E. The tenants send the same size files but 3x, 4x and 2x more often. For instance tenant 'b' is sending 16Mb files four times as often as before, and they now take longer as well -- nearly 29ms rather than 19ms -- because the other tenants are active more often too, so completion gets squeezed to later. Consequently, tenant 'b' is now sending 57% of the time, so its congestion-bit-rate is $20\text{kb/s} * 57\% = 11\text{kb/s}$. This is nearly 6x higher than in scenario E, reflecting both b's own increase by 4x and that this increase coincides with everyone else increasing their load.

In scenario G, tenant 'b' increases even more, to 5x the load it offered in scenario E. This results in average utilisation of 800Mb/s / 1Gb/s = 80%, compared to 72% in scenario F and only 24% in scenario E. 'b' sends the same files but 5x more often, so its load rises 5x.

Completion times rise for everyone due to the overall rise in load, but the congestion rates of 'a' and 'c' don't rise anything like as much as that of 'b', because they still leave large gaps between files. For instance, tenant 'c' completes each large file transfer in 4.5s (compared to 1.9s in scenario E), but it still only sends files every 10s. So 'c' only sends 45% of the time, which is reflected in its congestion bit-rate of 20kb/s * 45% = 9kb/s.

In contrast, on average tenant 'b' can only complete each medium-sized file transfer in 32ms (compared to 19ms in scenario E), but on average it starts sending another file after 40ms. So 'b' sends 79% of the time, which is reflected in its congestion bit-rate of 20kb/s * 79% = 16kb/s (rounded).

However, during the 45% of the time that 'c' sends a large file, b's completion time is higher than average (as shown in Figure 1). In fact, as shown in the maximum completion time column, 'b' completes in 46ms, but it starts sending a new file after 40ms, which is before the previous one has completed. Therefore, during each of c's large files, 'b' sends 46/40 = 116% of the time on average.

This actually means 'b' is overlapping two files for 16% of the time on average and sending one file for the remaining 84%. Whenever two file transfers overlap, 'b' will be causing 2 x 20kb/s = 40kb/s of congestion, which explains why tenant b in scenario G is the only case with a maximum congestion rate of 40kb/s rather than 20kb/s as in every other case. Over the duration of c's large files, 'b' would therefore cause congestion at an average rate of 20kb/s * 84% + 40kb/s * 16% = 23kb/s (or more simply 10kb/s * 116% = 23kb/s). Of course, when 'c' is not sending a large file, 'b' will contribute less to congestion, which is why its average congestion rate is 16kb/s overall, as discussed earlier.

4.3.2. Congestion Policing of On-Off Flows

Still referring to the numerical examples in Table 2, we will now discuss the effect of limiting each tenant with a congestion policer.

The network operator might have deployed congestion policers to cap each tenant's average congestion rate to 16kb/s. None of the tenants are exceeding this limit in any of the scenarios, but tenant 'b' is just shy of it in scenario G. Therefore all the tenants would be free to behave in all sorts of ways like those of scenarios E-G, but they

would be prevented from degrading the performance of the other tenants beyond the point reached by tenant 'b' in scenario G. If tenant 'b' added more load, the policer would prevent the extra load entering the network by focusing drop solely on tenant 'b', preventing the other tenants from experiencing any more congestion due to tenant 'b'. Then tenants 'a' and 'c' would be assured the (average) apparent bit-rates shown, whatever the behaviour of 'b'.

If 'a' added more load, 'c' would not suffer. Instead 'b' would go over limit and its rate would be trimmed during congestion peaks, sacrificing some of its lead to 'a'. Similarly, if 'c' added more load, 'b' would be made to sacrifice some of its performance, so that 'a' would not suffer. Further, if more tenants arrived to share the same link, the policer would force 'b' to sacrifice performance in favour of the additional tenants.

There is nothing special about a policer limit of 16kb/s. The example when discussing infinite flows used a limit of 40kb/s per tenant. And some tenants can be given higher limits than others (e.g. at an additional charge). If the operator gives out congestion limits that together add up to a higher amount but it doesn't increase the link capacity, it merely allows the tenants to apply more load (e.g. more files of the same size in the same time), but each with lower bit-rate.

{ToDo: Discuss min bit-rates}

{ToDo: discuss instantaneous limits and how they protect the minimum bit-rate of other tenants}

4.4. Weighted Congestion Controls

At high speed, congestion controls such as Cubic TCP, Data Centre TCP, Compound TCP etc all contribute to congestion at widely differing rates, which is called their 'aggressiveness' or 'weight'. So far, we have made the simplifying assumption of a scalable congestion control algorithm that contributes to congestion at a constant rate of $w = 20\text{kb/s}$. We now assume tenant 'c' uses a similar congestion control to before, but with different parameters in the algorithm so that its weight is still constant, but at $w = 2.2\text{kb/s}$.

Tenant 'b' still uses $w = 20\text{kb/s}$ for its smaller files, so when the two compete for the 1Gb/s link, they will share it in proportion to their weights, 20:2.2 (or 90%:10%). That is, 'b' and 'c' will respectively get $(20/22.2)*1\text{Gb/s} = 900\text{Mb/s}$ and $(2.2/22.2)*1\text{Gb/s} = 100\text{Mb/s}$ of the 1Gb/s link. Figure 2 shows the situation before (upper) and after (lower) this change.

When the two compete, 'b' transfers each file $9/5$ faster than before (900Mb/s rather than 500Mb/s), so it completes them in $5/9$ of the time. 'b' still contributes congestion at the same rate of 20kb/s, but for $5/9$ less time than before. Therefore, relative to before, 'b' uses up its allowance $5/9$ as quickly.

Tenant 'c' contributes congestion at $2.2/22.2$ of its previous rate, that is 2kb/s rather than 20kb/s. Although tenant 'b' goes faster, as each file finishes, it gets out of the way sooner, so 'c' can catch up to where it got to before after each 'b' file and should complete hardly any later than before. Tenant 'c' will probably lose some completion time because it has to accelerate and decelerate more. But, whenever it is sending a file, 'c' gains $(20\text{kb/s} - 2\text{kb/s}) = 18\text{kb}$ of allowance every second, which it can use for other transfers.

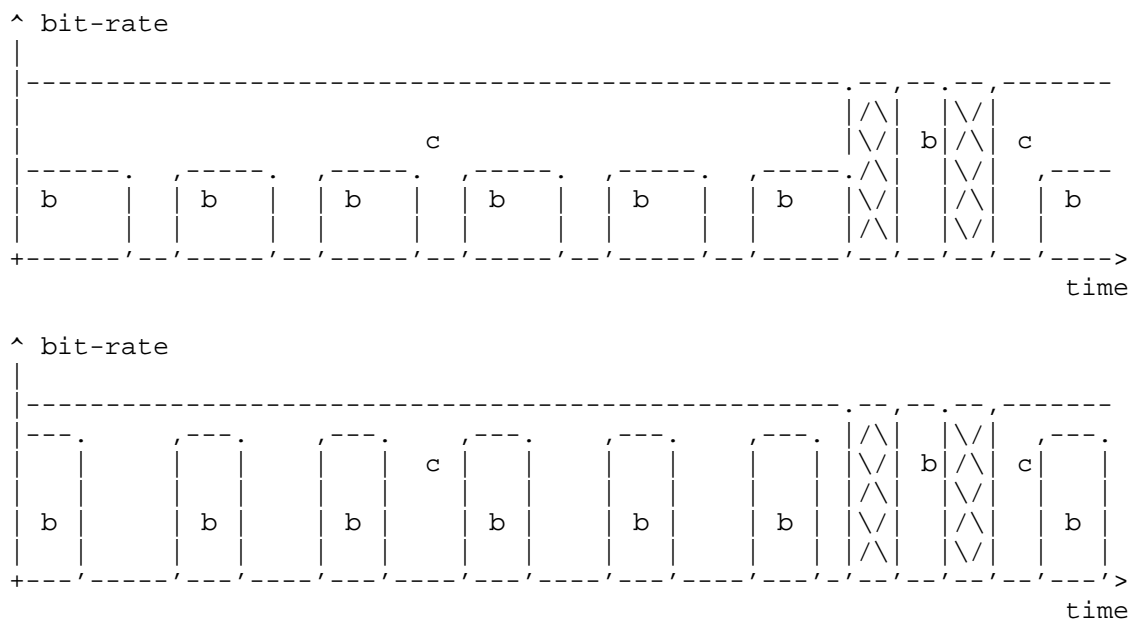


Figure 2: Weighted congestion controls with equal weights (upper) and unequal (lower)

It seems too good to be true that both tenants gain so much and lose so little by 'c' reducing its aggressiveness. The gains are unlikely to be as perfect as this simple model predicts, but we believe they will be nearly as substantial.

It might seem that everyone can keep gaining by everyone agreeing to reduce their weights, ad infinitum. However, the lower the weight,

the less signals the congestion control gets, so it starts to lose its control during dynamics. Nonetheless, congestion policing should encourage congestion control designs to keep reducing their weights, but they will have to stop when they reach the minimum necessary congestion in order to maintain sufficient control signals.

4.5. A Network of Links

So far we have only considered a single link. Congestion policing at the network edge is designed to work across a network of links, treating them all as a pool of resources, as we shall now explain. We will use the dual-homed topology shown in Figure 3 (stretching the bounds of ASCII art) as a very simple example of a pool of resources.

In this case where there are 48 servers (H_1, H_2, \dots, H_n where $n=48$) on the left, with on average 8 virtual machines (VMs) running on each (e.g. server n is running V_{n1}, V_{n2}, \dots to V_{nm} where $m = 8$). Each server is connected by two 1Gb/s links, one to each top-of-rack switch S_1 & S_2 . To the right of the switches, there are 6 links of 10Gb/s each, connecting onwards to customer networks or to the rest of the data centre. There is a total of $48 * 2 * 1\text{Gb/s} = 96\text{Gb/s}$ capacity between the 48 servers and the 2 switches, but there is only $6 * 10\text{Gb/s} = 60\text{Gb/s}$ to the right of the switches. Nonetheless, data centres are often designed with some level of contention like this, because at the ToR switches a proportion of the traffic from certain hosts turns round locally towards other hosts in the same rack.

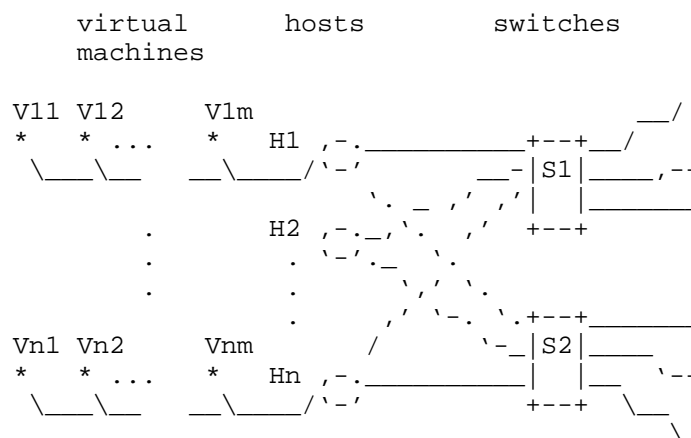


Figure 3: Dual-Homed Topology -- a Simple Resource Pool

The congestion policer proposed in this document is based on the 'hose' model, where a tenant's congestion allowance can be used for sending data over any path, including many paths at once. Therefore,

any one of the virtual machines on the left can use its allowance to contribute to congestion on any or all of the 6 links on the right (or any other link in the diagram actually, including those from the server to the switches and those turning back to other hosts).

Nonetheless, if congestion policers are to enforce performance isolation, they should stop one tenant squeezing the capacity available to another tenant who needs to use a particular bottleneck link or links. They should work whether the offending tenant is acting deliberately or merely carelessly.

The only way a tenant can become squeezed is if another tenant uses more of the bottleneck capacity, which can only happen if the other tenant sends more flows (or more aggressive flows) over that link. In the following we will call the tenant that is shifting flows 'active', and the ones already on a link 'passive'. These terms have been chosen so as not to imply one is bad and the other good -- just different.

The active tenant will increase flow completion times for all tenants (passive and active) using that bottleneck. Such an active tenant might shift flows from other paths to focus them onto one, which would not of itself use up any more congestion allowance (recall that a scalable congestion control uses up its congestion allowance at the same rate per flow whatever bit-rate it is going at Section 4.3 and therefore whatever path it is using). However, although the instantaneous rate at which the active tenant uses up its allowance won't alter, the increased completion times due to increased congestion will use up more of the active tenant's allowance over time (same rate but for more of the time). If the passive tenants are using up part of their allowances on other links, the increase in congestion will use up a relatively smaller proportion of their allowances. Once such an increase exceeds the active tenant's congestion allowance, the congestion policer will protect the passive tenants from further performance degradation.

A policer may not even have to directly intervene for tenants to be protected; load balancing may remove the problem first. Load balancing might either be provided by the network (usually just random), or some of the 'passive' tenants might themselves actively shift traffic off the increasingly congested bottleneck and onto other paths. Some of them might be using the multipath TCP protocol (MPTCP -- see experimental [RFC6356]) that would achieve this automatically, or ultimately they might shift their virtual machine to a different endpoint to circumvent the congestion hot-spot completely. Even if one passive tenant were not using MPTCP or could not shift easily, others shifting away would achieve the same outcome. Essentially, the deterrent effect of congestion policers

encourages everyone to even out congestion, shifting load away from hot spots. Then performance isolation becomes an emergent property of everyone's behaviour, due to the deterrent effect of policers, rather than always through explicit policer intervention.

{ToDo: Add numerical example}

In contrast, enforcement mechanisms based on scheduling algorithms like WRR or WFQ have to be deployed at each link, and each one works in isolation from the others. Therefore, each one doesn't know how much of other links the tenant is using. This is fine for networks with a single known bottleneck per customer (e.g. many access networks). However, in data centres there are many potential bottlenecks and each tenant generally only uses a share of a small number of them. A mechanism like WRR would not isolate anyone's performance if it gave every tenant the right to use the same share of all the links in the network, without regard to how many they were using.

The correct approach, as proposed here, is to give a tenant a share of the whole pool, not the same share of each link.

4.6. Links of Different Sizes

Congestion policing treats a Mb/s of capacity in one link as identical to a Mb/s of capacity in another link, even if the size of each link is different. For instance, consider the case where one of the three links to the right of each switch in Figure 3 were upgraded to 40Gb/s while the other two remained at 10Gb/s (perhaps to accommodate the extra traffic from a couple of the dual homed 1Gb/s servers being upgraded to dual-homed 10Gb/s).

Two congestion control algorithms running at the same rate will cause the same level of congestion probability, whatever size link they are sharing.

- o If 50 equal flows share a 10Gb/s link ($10\text{Gb/s} / 50 = 200\text{Mb/s}$ each) they will cause 0.01% congestion probability;
- o If 200 equal flows share a 40Gb/s link ($40\text{Gb/s} / 200 = 200\text{Mb/s}$ each) they will still cause 0.01% congestion probability;

This is because the congestion probability is determined by the congestion control algorithms, not by the link.

Therefore, if an average of 300 flows were spread across the above links (1x 40Gb/s and 2 x 10Gb/s), the numbers on each link would tend towards respectively 200:50:50, so that each flow would get 200Mb/s

and each link would have 0.01% congestion on it. Sometimes, there might be more flows on the bigger link, resulting in less than 200Mb/s per flow and congestion higher than 0.01%. However, whenever the congestion level was less on one link than another, congestion policing would encourage flows to balance out the congestion level across the links (as long as some flows could use congestion balancing mechanisms like MPTCP).

In summary, all the outcomes of congestion policing described so far (emulating WRR etc) apply across a pool of diverse link sizes just as much as they apply to single links.

4.7. Diverse Congestion Control Algorithms

Throughout this explanation we have assumed a scalable congestion control algorithm, which we justified Section 4.1 as the 'boundary' case if congestion policing had to intervene, which is all that is relevant when considering whether the policer can enforce performance isolation.

This performance isolation approach still works, whether or not the congestion controls in daily use by tenants fit this scalable model. A bulk congestion policer constrains the sum of all the congestion controls being used by a tenant so that they collectively remain below a large-scale envelope that is itself shaped like the sum of many scalable algorithms. Bulk congestion policers will constrain the overall congestion effect (the sum) of any mix of algorithms within it, including flows that are completely unresponsive to congestion. This is explained around Fig 3 of [CongPol].

{ToDo, summarise the relevant part of that paper here and perhaps even add ASCII art for the plot...}

{ToDo, bring in discussion of slow-start as effectively another variant of congestion control, with considerable overshoots, etc.}

The defining difference between the scalable congestion we have assumed and the congestion controls in widespread production operating systems (New Reno, Compound, Cubic, Data Centre TCP etc) is the way congestion probability decreases as flow-rate increases (for a long-running flow). With a scalable congestion control, if flow-rate doubles, congestion probability halves. Whereas, with most production congestion controls, if flow-rate doubles, congestion probability reduces to less than half. For instance, New Reno TCP reduces congestion to a quarter. The responses of Cubic and Compound are closer to the ideal scalable control than to New Reno, but they do not depart too far from TCP to ensure they can co-exist happily with New Reno.

5. Design

The design involves the following elements, all involving changes solely in the hypervisor or operating systems, not network switches:

Congestion Information at Ingress: This information needs to be trusted by the operator of the data centre infrastructure, therefore it cannot just use the feedback in the end-to-end transport (e.g. TCP SACK or ECN echo congestion experienced flags) that might anyway be encrypted. Trusted congestion feedback may be implemented in either of the following two ways:

- A. either as a shim in both sending and receiving hypervisors using an edge-to-edge (host-host) tunnel, with feedback messages reporting congestion back to the sending host's hypervisor (in addition to the e2e feedback at the transport layer).
- B. or in the sending operating system using the congestion exposure protocol (ConEx [ConEx-Abstract-Mech]);

Approach a) could be applied solely to traffic from operating systems that do not yet support the simpler approach b)

The host-host feedback tunnel (approach a) is easier to implement if a tunnelling overlay is already in use in the data centre. For instance, we believe it would be possible to build the necessary feedback facilities using the proposed network virtualisation approach based on generic routing encapsulation (GRE) [nvGRE]. The tunnel egress would also need to be able to detect congestion. This would be simple for e2e flows with ECN enabled, because this will lead to ECN also being enabled in the outer IP header [RFC6040]. However, for non-ECN enabled flows, it is more problematic. It might be possible to add sequence numbers to the outer headers, as is done in many pseudowire technologies. However, a simpler alternative is possible in a data centre where the switches can be ECN-enabled. It would then be possible to enable ECN in the outer headers, even if the e2e transport is not ECN-capable (Not-ECT in the inner header). At the egress, if the outer is marked as 'congestion experienced', but the inner is not-ECT, the packet would have to be dropped, being the only congestion signal the e2e transport would understand. But before dropping it, the ECN marking in the outer would have served the purpose of a congestion signal to the tunnel egress. Beyond this, implementation details of approach a) are still work in progress.

If the ConEx option is used (approach b), a congestion audit function will also be required as a shim in the hypervisor (or

container) layer where data leaves the network and enters the receiving host. The ConEx option is only applicable if the guest OS at the sender has been modified to send ConEx markings. For IPv6 this protocol is defined in [conex-destopt]. The ConEx markings could be encoded in the IPv4 header by hiding them within the packet ID field as proposed in [intarea-ipv4-id-reuse].

Congestion Policing: A bulk congestion policing function would be associated with each tenant's virtual machine to police all the traffic it sends into the network. It would most likely be implemented as a shim in the hypervisor. It would be expected that various policer designs might be developed, but here we propose a simple but effective one in order to be concrete. A token bucket is filled with tokens at a constant rate that represents the tenant's congestion allowance. The bucket is drained by the size of every packet with a congestion marking, as described in [CongPol]. If approach a) were used to get "Congestion Information at the Ingress", the bucket would be drained by congestion feedback from the tunnel egress. If approach b) were used, the bucket would be drained by ConEx markings on the actual data packets being forwarded (ConEx re-inserts the e2e feedback from the transport receiver back onto packets on the forward data path).

{ToDo: Add details of congestion burst limiting}

While the data centre network operator only needs to police congestion in bulk, tenants may wish to enforce their own limits on individual users or applications, as sub-limits of their overall allowance. Given all the information used for policing is readily available to tenants in the transport layer below their sender, any such per-flow, per-user or per-application limitations can be readily applied. The tenant may operate their own fine-grained policing software, or such detailed control capabilities may be offered as part of the platform (platform as a service or PaaS) above the more general infrastructure as a service (IaaS).

Distributed Token Buckets: A customer may run virtual machines on multiple physical nodes, in which case the data centre operator would ensure that it deployed a policer in the hypervisor on each node where the customer was running a VM, at the time each VM was instantiated. The DC operator would arrange for them to collectively enforce the per-customer congestion allowance, as a distributed policer.

A function to distribute a customer's tokens to the policer associated with each of the customer's VMs would be needed. This could be similar to the distributed rate limiting of [DRL]. Alternatively, a logically centralised bucket of congestion tokens

could be used with simple 1-1 communication between it and each local token bucket in the hypervisor under each VM.

Importantly, traditional bit-rate tokens cannot simply be reassigned from one VM to another without implications on the balance of network loading (requiring operator intervention each time), whereas congestion tokens can be freely reassigned between different VMs, because a congestion token is equivalent at any place or time in a network;

As well as distribution of tokens between the VMs of a tenant, it would similarly be feasible to allow transfer of tokens between tenants, also without breaking the performance isolation properties of the system. Secure token transfer mechanisms could be built above the underlying policing design described here. Therefore the details of token transfer need not concern us here, and can be deferred to future work.

Switch/Router Support: Network switches/routers would not need any modification. However, both congestion detection by the tunnel (approach a) and ConEx audit (approach b) would be easier if switches supported ECN.

Data centre TCP might be used as well, although not essential. DCTCP requires ECN and is designed for data centres. DCTCP requires modified sender and receiver TCP algorithms as well as a more aggressive active queue management algorithm in the L3 switches. The AQM involves a step threshold at a very shallow queue length for ECN marking.

6. Parameter Setting

{ToDo: }

7. Incremental Deployment

7.1. Migration

A pre-requisite for ingress congestion policing is the function entitled "Congestion Information at Ingress " in Section 5. Tunnel feedback (approach a) is a more processing intensive change to the hypervisors, but it can be deployed unilaterally by the data centre operator in all hypervisors (or containers), without requiring support in guest operating systems.

Using ConEx markings (approach b) is only applicable if a particular guest OS supports the marking of outgoing packets with ConEx markings. But if available this is simpler and more efficient.

Both functions could be implemented in each hypervisor, and a simple filter could be installed to allow ConEx packets through into the data centre network (approach a) without going through the feedback tunnel shim, while non-ConEx packets would need to be tunnelled and to elicit tunnel feedback (approach b). This would provide an incremental deployment scenario with the best of both worlds: it would work for unmodified guest OSs, but for guest OSs with ConEx support, it would require less processing (therefore being faster) and not require the considerable overhead of a duplicate feedback channel between hypervisors (sending and forwarding a large proportion of tiny packets).

{ToDo: Note that the main reason for preferring ConEx information will be because it is designed to represent a conservative expectation of congestion, whereas tunnel feedback represents congestion only after it has happened.}

7.2. Evolution

Initially, the approach would be confined to intra-data centre traffic. With the addition of ECN support on network equipment in the WAN between data centres, it could straightforwardly be extended to inter-data centre scenarios, including across interconnected backbone networks.

Having proved the approach within and between data centres and across interconnect, more mass-market devices might be expected to be turned on support for ECN feedback, and ECN might be turned on in equipment in wider networks most likely to be bottlenecks (access and backhaul).

8. Related Approaches

The Related Work section of [CongPol] provides a useful comparison of the approach proposed here against other attempts to solve similar problems.

When the hose model is used with Diffserv, capacity has to be considerably over-provisioned for all the unfortunate cases when multiple sources of traffic happen to coincide even though they are all in-contract at their respective ingress policers. Even so, every node within a Diffserv network also has to be configured to limit higher traffic classes to a maximum rate in case of really unusual traffic distributions that would starve lower priority classes. Therefore, for really important performance assurances, Diffserv is used in the 'pipe' model where the policer constrains traffic separately for each destination, and sufficient capacity is provided at each network node for the sum of all the peak contracted rates for paths crossing that node.

In contrast, the congestion policing approach is designed to give full performance assurances across a meshed network (the hose model), without having to divide a network up into pipes. If an unexpected distribution of traffic from all sources focuses on a congestion hotspot, it will increase the congestion-bit-rate seen by the policers of all sources contributing to the hot-spot. The congestion policers then focus on these sources, which in turn limits the severity of the hot-spot.

The critical improvement over Diffserv is that the ingress edges receive information about any congestion occurring in the middle, so they can limit how much congestion occurs, wherever it happens to occur. Previously Diffserv edge policers had to limit traffic generally in case it caused congestion, because they never knew whether it would (open loop control).

Congestion policing mechanisms could be used to assure the performance of one data flow (the 'pipe' model), but this would involve unnecessary complexity, given the approach works well for the 'hose' model.

Therefore, congestion policing allows capacity to be provisioned for the average case, not for the near-worst case when many unlikely cases coincide. It assures performance for all traffic using just one traffic class, whereas Diffserv only assures performance for a small proportion of traffic by partitioning it off into higher priority classes and over-provisioning relative to the traffic contracts sold for for this class.

{ToDo: Refer to Section 4 for comparison with WRR & WFQ}

Seawall {ToDo} [Seawall]

9. Security Considerations

10. IANA Considerations

This document does not require actions by IANA.

11. Conclusions

{ToDo}

12. Acknowledgments

13. Informative References

- [ConEx-Abstract-Mech] Mathis, M. and B. Briscoe, "Congestion Exposure (ConEx) Concepts and Abstract Mechanism", draft-ietf-conex-abstract-mech-03 (work in progress), October 2011.
- [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>>.
- [DRL] Raghavan, B., Vishwanath, K., Ramabhadran, S., Yocum, K., and A. Snoeren, "Cloud control with distributed rate limiting", ACM SIGCOMM CCR 37(4)337--348, 2007, <<http://doi.acm.org/10.1145/1282427.1282419>>.
- [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.
- [RFC3649] Floyd, S., "HighSpeed TCP for Large Congestion Windows", RFC 3649, December 2003.
- [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.
- [RFC6356] Raiciu, C., Handley, M., and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols", RFC 6356, October 2011.
- [Seawall] Shieh, A., Kandula, S., Greenberg, A., and C. Kim, "Seawall: Performance Isolation in Cloud Datacenter Networks", Proc 2nd USENIX Workshop on Hot Topics in Cloud Computing ,

June 2010, <<http://research.microsoft.com/en-us/projects/seawall/>>.

[conex-dc_tr] Briscoe, "Network Performance Isolation in Data Centres by Congestion Exposure to Edge Policers", BT Technical Report TR-DES8-2011-004, November 2011.

Work in progress

[conex-destopt] Krishnan, S., Kuehlewind, M., and C. Ucendo, "IPv6 Destination Option for Conex", draft-ietf-conex-destopt-01 (work in progress), October 2011.

[intarea-ipv4-id-reuse] Briscoe, B., "Reusing the IPv4 Identification Field in Atomic Packets", draft-briscoe-intarea-ipv4-id-reuse-01 (work in progress), March 2012.

[nvgre] Sridhavan, M., Greenberg, A., Venkataramaiah, N., Wang, Y., Duda, K., Ganga, I., Lin, G., Pearson, M., Thaler, P., and C. Tumuluri, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-01 (work in progress), July 2012.

Appendix A. Summary of Changes between Drafts

Detailed changes are available from
<http://tools.ietf.org/html/draft-briscoe-conex-data-centre>

From draft-briscoe-conex-initial-deploy-02 to
draft-briscoe-conex-data-centre-00:

- * Split off data-centre scenario as a separate document, by popular request.

Authors' Addresses

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/>

Murari Sridharan
Microsoft
1 Microsoft Way
Redmond, WA 98052

Phone:
Fax:
EMail: muraris@microsoft.com
URI:

Congestion Exposure (ConEx) Working
Group
Internet-Draft
Intended status: Informational
Expires: April 27, 2015

M. Mathis
Google, Inc
B. Briscoe
BT
October 24, 2014

Congestion Exposure (ConEx) Concepts, Abstract Mechanism and
Requirements
draft-ietf-conex-abstract-mech-13

Abstract

This document describes an abstract mechanism by which senders inform the network about the congestion recently encountered by packets 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 April 27, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

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. Constraints on the Audit Function	8
3.3. Requirements for non-abstract ConEx specifications	9
4. Encoding Congestion Exposure	11
4.1. Naive Encoding	11
4.2. Null Encoding	12
4.3. ECN Based Encoding	12
4.4. Independent Bits	13
4.5. Codepoint Encoding	13
4.6. Units Implied by an Encoding	14
5. Congestion Exposure Components	15
5.1. Network Devices (Not modified)	15
5.2. Modified Senders	15
5.3. Receivers (Optionally Modified)	16
5.4. Policy Devices	16
5.4.1. Congestion Monitoring Devices	16
5.4.2. Rest-of-Path Congestion Monitoring	17
5.4.3. Congestion Policers	17
5.5. Audit	18
6. Support for Incremental Deployment	21
7. IANA Considerations	24
8. Security Considerations	24
9. Acknowledgements	25
10. Comments Solicited	26
11. References	26
11.1. Normative References	26
11.2. Informative References	26

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. This document lays out requirements for concrete protocol specifications.

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 queueing 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).

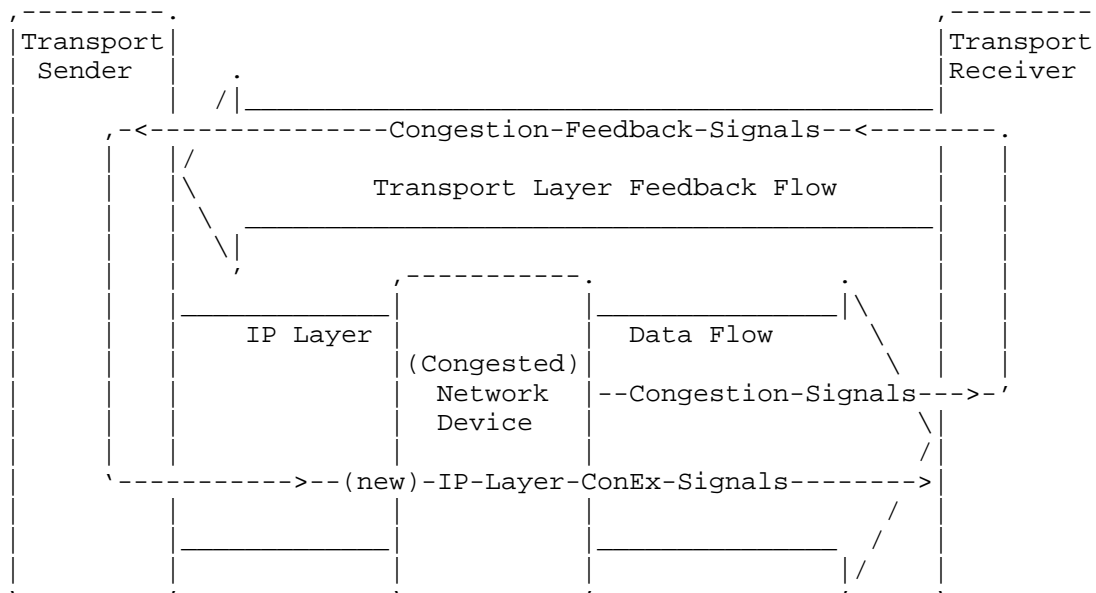


Figure 1: The Flow of Congestion and ConEx Signals

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 to 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 [RFC7141] 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 needs to 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 and/or at coarser granularity than single microflows. 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 [RFC6817] (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 signal in 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 detected an ECN congestion experienced (CE) mark.

Credit: The transport is building up credit to signal advance notice of the risk of packets contributing to congestion, in contrast to signalling only after inherently delayed feedback of actual congestion.

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.

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

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. It MAY need to be visible if other encapsulating headers are used to interconnect networks. 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], [RFC6679] for example).
- d. The ConEx signal SHOULD be accurate and auditable. The general approach for auditing 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).

- e. The ConEx signals for packet loss and ECN marking SHOULD have distinct encodings because they are likely to require different auditing techniques.
- f. Additionally there SHOULD be an auditable ConEx Credit signal. A sender can use Credit to indicate potential future congestion, for example as often seen during startup. ConEx Credit is intended to overestimate congestion actually experienced across the network.

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. Constraints on the Audit Function

The role of the audit function and constraints on it 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 connection and while it is in progress.
- 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 sidesteps 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 (2014) the 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. Any individual packet may need to represent any ECN codepoint and any ConEx signal value independently. So, 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 (network layer encoding requirement L in Section 3.3) says 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 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. So, ideally, a ConEx sender would be able to reflect these in the next packet. However, there will be many invalid combinations of flags (e.g. Not-ConEx combined with any of the ConEx-marked flags), which a malicious sender could use 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 an audit function measuring in bytes (see Section 4.6).

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-carrying 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.

[RFC7141] advises that congestion indications should be interpreted in units of bytes when responding to congestion, at least on today's Internet. [RFC6789] takes the same view in its definition of congestion-volume, again for 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).

RFC 7141 constructs an argument for why equipment tends to be built so that the bottleneck will be the bit-carrying capacity of its interfaces not its packet processing capacity. However, RFC 7141 acknowledges that the position may change in future, and notes that new techniques will need to be developed to distinguish packet- and bit-congestion.

Given this document describes an abstract ConEx mechanism, it is intended to be timeless. Therefore it 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 [RFC7141].

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-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 it 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.

The native loss or ECN signaling mechanism required for compliance with existing congestion control standards (e.g. RTCP, SCTP) will typically be sufficient for the Sender to generate ConEx signals.

TCP's loss feedback is sufficient for ConEx if SACK is used [RFC2018]. However, the original specification for ECN in TCP [RFC3168] 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 in the TCP-ECN case an optional modification to the receiver could be recommended for precision (see [I-D.ietf-tcpm-accecn-reqs], which is based on the approach originally 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/or Credit 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 and/or Credit-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 could be to introduce a throttle (discard some traffic) immediately upstream of the congestion monitor. Alternatively, this throttle could introduce delay using 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 sanctions based on ConEx signals will create an intrinsic motivation for users to understate the congestion that they are causing. So, without strong audit functions, the ConEx signal would become understated to the point of being useless. Therefore the most important feature of an encoding design is likely to be the robustness of the auditing it supports.

The general goal of an auditor is to make sure that any ConEx-enabled traffic is sent with sufficient ConEx-Re-Echo and ConEx-Credit signals. A concrete definition of the ConEx protocol MUST define what sufficient means.

If a ConEx-enabled transport does not carry sufficient ConEx signals, then an auditor is likely to apply some sanction to that 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, a tactical audit approach could be to 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 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.

Similarly, the job of incentivising the sending of ConEx-enabled packets is proper solely to policy devices, independent of the audit function. The audit function's job is policy-neutral, so it should be solely confined to checking for correctness within those packets that have been marked as ConEx-capable. Even if there are Not-ConEx packets mixed with ConEx packets within a flow, audit will not need to monitor any Not-ConEx packets.

Note that in the future it might prove to be desirable to provide advice on uniformly implementing sanctions, because otherwise insufficient sanctions could impair 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. The auditor would treat all the remaining flows without any ConEx-marked packets as a single misbehaving aggregate.

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 needs to counter the tactic where a source cheats until caught by sampling, then simply discards that flow ID and starts cheating with a new one (termed 'identifier white-washing when caught').

For the the concrete ConEx protocol encoding defined in [I-D.ietf-conex-destopt], ConEx Credit and ConEx-Re-Echo signals are intended to be audited separately. The Credit signal can be audited directly against actual congestion (loss and ECN). However, there will be an inherent delay of at least one round trip between a congestion signal and the subsequent ConEx-Re-Echo signal it triggers, as shown in Figure 1. Therefore ConEx-Re-Echo signals will need to be audited with some allowance for this delay. Further discussion of design and implementation choices for functions intended to audit this concrete ConEx encoding can be found in [I-D.wagner-conex-audit].

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 that support incremental deployment in the concrete ConEx specifications, 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 with the regular receiver for the transport in question, without requiring any ConEx-specific modifications. This is true for modern transport protocols (RTCP, SCTP etc) and it is even true for TCP, as long as the receiver supports SACK, which is widely deployed anyway. However, it is not true for ECN feedback in TCP. The need for more precise ECN feedback in TCP is not exclusive to ConEx, for instance Data Centre TCP (DCTCP [DCTCP]) uses precise feedback to good effect. Therefore, if a receiver offers precise feedback, [I-D.ietf-tcpm-accecn-reqs] it will be best if ConEx uses it (see Section 5.3). Alternatively, without sufficiently precise congestion feedback from the receiver, the source may have to conservatively send extra ConEx markings in order to avoid understating congestion.

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 theoretically 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 some 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.ietf-conex-destopt] 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, an operator relying on ConEx signals is recommended to enable ECN in queues wherever possible. This is because 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 audit devices itself.

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 the concrete ConEx protocol defined in [I-D.ietf-conex-destopt] 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:

- o 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.
- o 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 Dukkkipati, Mirja Kuehlewind, Caitlin Bestler, Marcelo Bagnulo Braun, John Leslie, Ingemar Johansson and David Wagner.

Bob Briscoe's work on this specification received part-funding from the European Union's Seventh Framework Programme FP7/2007-2013 under Trilogy 2 project, grant agreement no. 317756. The views expressed here are solely those of the author.

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.sciencedirect.com/science/article/pii/S0005109899001351>>.
- [FairerFaster] Briscoe, B., "A Fairer, Faster Internet Protocol", IEEE Spectrum Dec 2008:38--43, December 2008, <<http://bobbbriscoe.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-02
(work in progress), July 2013.
- [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-02
(work in progress), July 2013.
- [I-D.ietf-conex-destopt] Krishnan, S., Kuehlewind, M., and
C. Ucendo, "IPv6 Destination Option
for ConEx",
draft-ietf-conex-destopt-05 (work
in progress), October 2013.
- [I-D.ietf-tcp-modifications] Kuehlewind, M. and R.
Scheffenegger, "TCP modifications
for Congestion Exposure", draft-
ietf-conex-tcp-modifications-04
(work in progress), July 2013.
- [I-D.ietf-tcpm-accecn-reqs] Kuehlewind, M. and R.
Scheffenegger, "Problem Statement
and Requirements for a More
Accurate ECN Feedback",
draft-ietf-tcpm-accecn-reqs-04
(work in progress), October 2013.
- [I-D.wagner-conex-audit] Wagner, D. and M. Kuehlewind,
"Auditing of Congestion Exposure
(ConEx) signals",
draft-wagner-conex-audit-01 (work
in progress), February 2014.
- [RFC2018] Mathis, M., Mahdavi, J., Floyd, S.,
and A. Romanow, "TCP Selective
Acknowledgment Options", RFC 2018,
October 1996.

- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.
- [RFC3514] Bellovin, S., "The Security Flag in the IPv4 Header", RFC 3514, April 1 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.
- [RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, August 2012.
- [RFC6789] Briscoe, B., Woundy, R., and A. Cooper, "Congestion Exposure (ConEx) Concepts and Use Cases", RFC 6789, December 2012.
- [RFC6817] Shalunov, S., Hazel, G., Iyengar, J., and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)", RFC 6817, December 2012.
- [RFC7141] Briscoe, B. and J. Manner, "Byte and Packet Congestion Notification", BCP 41, RFC 7141,

February 2014.

[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://portal.acm.org/citation.cfm?id=1080091.1080124>>.

[Refb-dis]

Briscoe, B., "Re-feedback: Freedom with Accountability for Causing Congestion in a Connectionless Internetwork", UCL PhD Dissertation, 2009, <<http://discovery.ucl.ac.uk/16274/>>.

[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

E-Mail: mattmathis@google.com

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

Phone: +44 1473 645196
E-Mail: bob.briscoe@bt.com
URI: <http://bobbbriscoe.net/>

CONEX WG
Internet-Draft
Intended status: Informational
Expires: January 10, 2013

D. Kutscher
F. Mir
R. Winter
NEC
S. Krishnan
Y. Zhang
Ericsson
CJ. Bernardos
UC3M
July 9, 2012

Mobile Communication Congestion Exposure Scenario
draft-ietf-conex-mobile-00

Abstract

This memo describes a mobile communications use case for congestion exposure (CONEX) with a particular focus on mobile communication networks such as 3GPP Evolved Packet System (EPS). The draft provides a brief overview of the architecture of these networks (both access and core networks), current QoS mechanisms and then discusses how congestion exposure concepts could be applied. Based on this, this memo suggests a set of requirements for CONEX mechanisms that particularly apply to mobile networks.

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 10, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Overview of 3GPP's Evolved Packet System (EPS)	3
3. CONEX Use Cases in the Mobile Communication Scenario	5
3.1. CONEX as a Basis for Traffic Management	6
3.2. CONEX to Incentivize Scavenger Transports	7
3.3. Accounting for Congestion Volume	8
3.4. CONEX as a Form of Differential QoS	9
3.5. Partial vs. Full Deployment	9
3.6. Summary	10
4. CONEX in the EPS	11
4.1. Possible Deployment Scenarios	11
4.2. Implementing CONEX Functions in the EPS	14
4.2.1. CONEX Protocol Mechanisms	15
4.2.2. CONEX Functions in the Mobile Network	15
5. Summary	17
6. IANA Considerations	19
7. Security Considerations	19
8. Informative References	19
Appendix A. Acknowledgments	21
Authors' Addresses	21

1. Introduction

Mobile data traffic continues to grow rapidly. The challenge wireless operators face is to support more subscribers with higher bandwidth requirements. To meet the bandwidth demand, there is a need for new technologies that assist the operators in efficiently utilizing the available network resources. Two specific areas where such new technologies could be deemed useful are resource allocation and flow management. Analysis of widely available statistics for network traffic from cellular networks are available, reveals that most flows are short-lived and low-volume, but there a few large flows that constitute a large part of the overall traffic volume. Measurements have also shown that a small number of users is responsible for the majority of traffic in cellular networks. In view of such highly skewed user behavior and limited and expensive resources (Wireless Spectrum), resource allocation and usage accountability are two important issues for operators to solve in order to achieve a better and fair network resource utilization. CONEX, as described in [I-D.ietf-conex-concepts-uses], is a technology that can be used to do so.

The CONEX congestion exposure mechanism is intended as a general technology that could be applied as a key element of congestion management solutions in a variety of use cases. The IETF CONEX WG will however work on a specific use case, where the end hosts and the network that contains the destination end host are CONEX-enabled but other networks need not be.

A specific example of such a use case can be a mobile communication network such as a 3GPP Evolved Packet System (EPS) network, where UEs (User Equipment, i.e. mobile end hosts), servers and caches, the access network and possibly an operator's core network can be CONEX-enabled. I.e., hosts support the CONEX mechanisms, and the network provides policing/auditing functions at its edges.

This document provides a brief overview of the architecture of such networks (access and core networks), current QoS mechanisms and then discusses how congestion exposure concepts can benefit such networks and how they should be applied. Using this use case as a basis, a set of requirements for CONEX mechanisms are described.

2. Overview of 3GPP's Evolved Packet System (EPS)

This section provides an overview of 3GPP's "Evolved Packet System" (EPS [3GPP.36.300]) as a specific example of a mobile communication architecture in order to illustrate congestion exposure applicability in this memo. There are other mobile communication architectures.

The EPS architecture and its standardized interfaces are depicted in Figure 1. The EPS provides IP connectivity to user equipment (UE) (i.e., mobile nodes) and access to operator services, such as global Internet access and voice communications. The EPS comprises the radio access network called evolved UMTS Terrestrial Radio Access Network (E-UTRAN) and the core network called Evolved Packet Core (EPC). QoS is supported through an EPS bearer concept, providing hierarchical bindings within the network.

The evolved NodeB (eNB), the Long Term Evolution (LTE) base station, is part of the access network that provides radio resource management, header compression, security and connectivity to the core network through the S1 interface. In an LTE network, the control plane signaling traffic and the data traffic are handled separately. The eNBs transmit the control traffic and data traffic separately via two logically separate interfaces.

The Home Subscriber Server, HSS, is a database that contains user subscriptions and QoS profiles. The Mobility Management Entity, MME, is responsible for user authentication, bearer establishment and modification and maintenance of the UE context.

The Serving gateway, S-GW, is the mobility anchor and manages the user plane data tunnels during the inter-eNB handovers. It tunnels all user data packets and buffers downlink IP packets destined for UEs that happen to be in idle mode.

The Packet Data Network (PDN) Gateway, P-GW, is responsible for IP address allocation to the UE and is a tunnel endpoint for mobility protocols. It is also responsible for charging, packet filtering, and policy-based control of flows. It interconnects the mobile network to external IP networks, e.g. the Internet.

In this architecture, data packets are not sent directly on an IP network between the eNB and the gateways. Instead, every packet is tunneled over a tunneling protocol - the GPRS Tunneling Protocol (GTP [3GPP.29.060]) over UDP/IP. A GTP path is identified in each node with the IP address and a UDP port number on the eNB/gateways. The GTP protocol carries both the data traffic (GTP-U tunnels) and the control traffic (GTP-C tunnels [3GPP.29.274]). Alternatively Proxy Mobile IP (PMIPv6) is used on the S5 interface.

The above is very different from an end-to-end path on the Internet where the packet forwarding is performed at the IP level. Importantly, we observe that these tunneling protocols give the operator a large degree of flexibility to control the congestion mechanism incorporated with the GTP/PMIPv6 protocols.

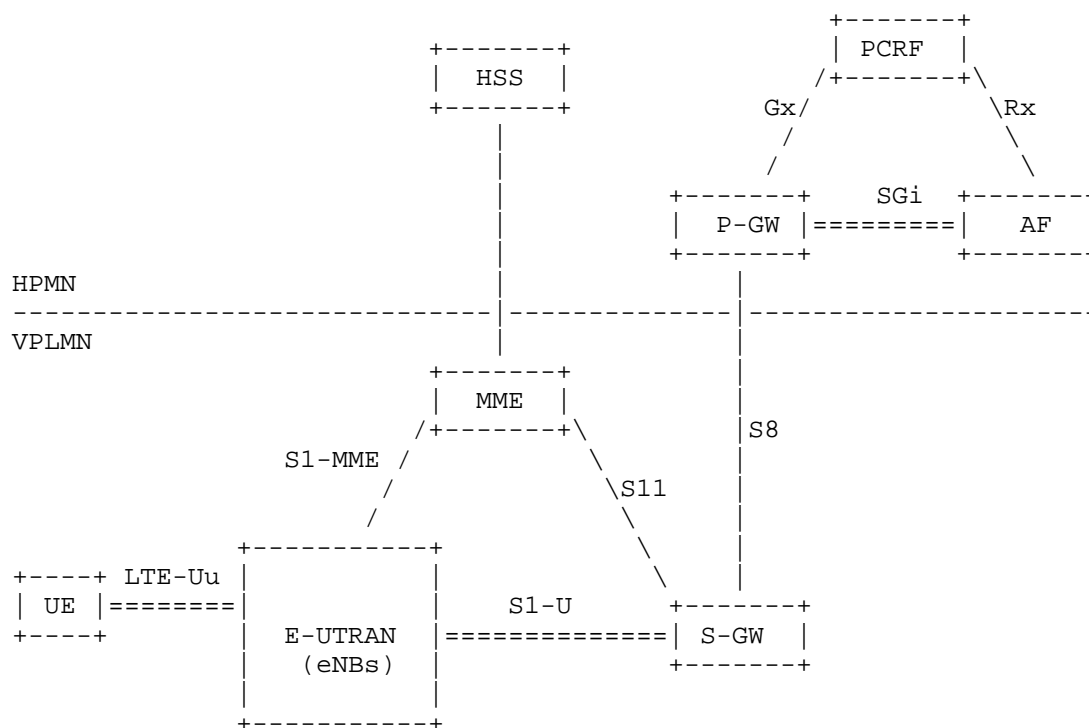


Figure 1: EPS (non-roaming) architecture overview

3. CONEX Use Cases in the Mobile Communication Scenario

In general, quality of service and good network resource utilization are important requirements for mobile communication network operators. Radio access and backhaul networks are considered scarce resources, and bandwidth (and radio resource) demand is difficult to predict precisely due to user mobility, radio propagation effects etc. Hence today's architectures and protocols go to significant extent in order to provide network-controlled quality of service -- for instance by 3GPP's EPS bearer model that enables the network to allocate service data flows (SDFs) to certain EPS bearers with specific quality of service classes (which can be used for fine-granular per-application service differentiation).

In the following, we discuss ways how congestion exposure could be beneficial for supporting resource management in such mobile communication networks. [I-D.ietf-conex-concepts-uses] describes fundamental congestion exposure concepts and a set of use cases for applying congestion exposure mechanisms to realize different traffic

management functions, accounting etc. Here, we relate these CONEX use cases to the general mobile communication scenario in order to validate the use cases for this scenario.

3.1. CONEX as a Basis for Traffic Management

Traffic management is a very important function in mobile communication networks. Since wireless resources are considered scarce and since user mobility and shared bandwidth in the wireless access create certain dynamics with respect to available bandwidth, these resources are traditionally managed very tightly (admission control for bearer establishment etc.).

In EPS, the QoS requirements for different applications running on a UE are supported by a bearer concept which is managed by the network. Each bearer has an associated QoS Class identifier (QCI) and an Allocation and Retention Policy (ARP) that has been standardized for uniform traffic handling (across implementations). For the necessary QoS across the mobile network, an EPS bearer is maintained that crosses different interfaces in the network and maps to lower layer bearers for packet forwarding. A radio bearer transports traffic between a UE and eNB whereas S1 bearer transports traffic between the eNB and S-GW. Primarily LTE offers two types of bearer: Guaranteed Bit rate bearer for real time communication, e.g., Voice calls etc. and Non-Guaranteed bit rate, e.g., best effort traffic for web access etc. Packets mapped to the same EPS bearer receive the same bearer level packet forwarding treatment.

In the light of the significant increase of overall data volume in 3G networks, Deep-Packet-Inspection (DPI) is often considered a desirable function to have in the EPC -- on, for example, a PDN (Packet Data Network) gateway, and some operators do in fact deploy DPI today. 3GPP has a current work item on "Service Awareness and Privacy Policies" that is chartered to add DPI-related extensions to the PCC architecture [3GPP.23.203]. The (optional) DPI entity in the EPC is called "Traffic Detection Function" (TDF), and it performs application detection and reporting of detected application and its service data flow description to the Policy Control and Charging Rules Function (PCRF) for performing functions such as traffic blocking, redirection, policing for selected flows.

Congestion exposure can be employed to address these requirements for tight resource management in different ways:

1. It can enhance DPI by providing flow policy-based traffic management. At present, DPI-based resource management is often used to prioritize certain application classes with respect to others in overload situations, so that effectively more users can

be served on the network. In overload situations, operators use DPI to identify dispensable flows and make them yield to other flows (of different application classes) through policing. Such traffic management is thus based on static configuration and some estimation about the future per-flow bandwidth demand. With congestion exposure it would be possible to assess, in a more accurate and dynamic fashion, the congestion that certain flows are causing. This information can then be input to a policer that can optimize network utilization (better than a pure DPI-based approach can do).

2. It can reduce the need for DPI by allowing for a bulk packet traffic management system that does not have to consider flows' application classes and individual sessions. Instead traffic management would be based on the current cost (contribution to congestion) incurred by different flows and enable operators to apply policing/accounting depending on their preference. Such traffic management would be simpler and more robust (no real-time flow application type identification required, no static configuration of application classes) and perform better as decisions can be taken based on real-time actual cost contribution.
3. It can be used to more effectively trigger the offload of selected traffic to a non-3GPP network. Nowadays, it is common that users are equipped with dual mode mobile phones (e.g., integrating third/fourth generation cellular and WiFi radio devices) capable of attaching to available networks either sequentially or simultaneously. With this scenario in mind, 3GPP is currently looking at mechanisms to seamlessly and selectively switch over a single IP flow (e.g., user application) to a different radio access, while keeping all other ongoing connections untouched. The decision on when and which IP flows move is typically based on static configured rules, whereas the use of CONEX mechanisms could also factor in real-time congestion events in the decision.

In summary, it can be said that traffic management in 3GPP EPS and other mobile communication architectures is very important. Currently, more static approaches based on admission control and static QoS are in use, but recently, there has been a perceived need for more dynamic mechanisms such as DPI. Adding CONEX support might thus require slight changes the PCC architecture, depending on the scope and impact of a CONEX-based traffic management approach.

3.2. CONEX to Incentivize Scavenger Transports

As 3G and LTE networks are turning into universal access networks that are shared between mobile (smart) phone users, mobile users with

laptop PCs, home users with LTE access etc., it is likely that capacity-sharing among different users and application flows becomes more important in the mobile communication network as a fine-granular differentiation would be too costly.

Most of this traffic is likely to be classified as best-effort traffic, without differentiating (for example) periodic OS updates, application store downloads from web (browser)-based or other more real-time communication. Having said that, the general argument for scavenger transports apply. Especially when wireless and backhaul resources are scarce, incentivizing users to use less-than best effort transport for non-interactive background communication would improve the overall utility of the network. It can be argued that, if this would be done with a CONEX approach, it could be done in a more effective and cost-efficient way compared to the aforementioned DPI mechanisms.

This would work best if the network did not do any traffic class segregation below the IP layer, i.e., if all traffic would be in the same traffic class. In principle, this would be possible to implement with current specifications.

3.3. Accounting for Congestion Volume

3G and LTE networks provide extensive support for accounting and charging already, for example cf. the Policy Charging Control (PCC) architecture. In fact, most operators today account transmitted data volume on a very fine granular basis and either correlate monthly charging to the exact number of packets/bytes transmitted, or employ some form of flat rate (or flexible flat rate), often with a so-called fair-use policy. With such policies, users are typically limited to an administratively configured maximum bandwidth limit, after they have used their data contractual volume budget for the charging period.

Changing this data volume-based accounting to a congestion-based accounting would be possible in principle, especially since there already is an elaborate per-user accounting system available. Also, an operator-provided mobile communication network can be seen as a network domain within such congestion volume accounting would be possible, without requiring any support from the global Internet. Traffic normally leaves/enters the operator's network via well-defined egress/ingress points that would be ideal candidates for policing functions. Moreover, in most commercially operated networks, accounting is performed for both received and sent data, which would facilitate congestion volume accounting as well.

With respect to the current PCC framework, accounting for congestion

volume could be added as another feature to the "Usage Monitoring Control" capability that is currently based on data volume. This would not require any new interface (reference points) at all.

3.4. CONEX as a Form of Differential QoS

As mentioned above, 3GPP mobile communication networks provide an elaborate QoS architecture. In LTE, the idea is to map different traffic classes onto different logical channels (bearers) with individual QoS configuration.

It can be argued whether this approach is sufficient in a world where most traffic is on TCP port 80 and whether some more application control would be useful.

With CONEX, accurate downstream path information would be visible to ingress network operators, which can respond to incipient congestion in time. This can be equivalent to offering different levels of QoS, e.g. premium service with zero congestion response.

Again, CONEX could be used in two different ways:

1. as additional information to assist network functions to impose different QoS for different application sessions; and
2. as a tool to let applications decide on their response to congestion notification, while incentivizing them to react (in general) appropriately, e.g., by enforcing overall limits for congestion contribution or by accounting and charging for such congestion contribution.

3.5. Partial vs. Full Deployment

In general CONEX lends itself to partial deployment as the mechanism does not require all routers and hosts to support congestion exposure. Moreover, assuming a policing infrastructure has been put in place, it is not required to modify all hosts. Since CONEX is about senders exposing congestion contribution to the network, senders need to be made CONEX-aware (assuming a congestion notification mechanisms such as ECN is in place).

[I-D.briscoe-conex-initial-deploy] provides specific examples of how CONEX deployment can be initiated, focusing unilateral deployment by single networks, i.e., by partial deployment.

In mobile communication networks that would for example allow early partial CONEX deployment in the downlink direction only, i.e., servers, gateways and caches would support CONEX but UEs (mobile

hosts) would not.

When moving towards full deployment in a specific operator's network, different ways for introducing CONEX support on UEs are feasible. Since mobile communication networks are multi-vendor networks, standardizing CONEX support on UEs (e.g., in 3GPP specifications) appears useful. Still, not all UEs would have to support CONEX, and operators would be free to choose their policing approach in such deployment scenarios. Leveraging existing PCC architectures, 3GPP network operators could for example decide policing/accounting approaches per UE -- i.e., apply fixed volume caps for non-CONEX UEs and more flexible schemes for CONEX-enabled UEs.

Moreover, it should be noted that network support for CONEX is a feature that some operators may implement to deploy if they wish, but it is not required that all operators (or all other networks) do so.

Depending on the extent of CONEX support, specific aspects such as roaming have to be taken into account. I.e., what happens when a user is roaming in a CONEX-enabled network, but their UE is not CONEX-enabled and vice versa. Although these may not be fundamental problems, they need to be considered. For supporting mobility in general, it can be required to shift users' policing state during hand-over. There is existing work in [raghavan2007] on distributed rate limiting and in [nec.euronf-2011] on specific optimizations for congestion exposure and policing in mobility scenarios.

Another aspect to consider is the addition of Selected IP Traffic Offload (SIPTO) and Local Breakout (LIPA), also see [3GPP.23.829], i.e., the idea that some traffic (e.g., high-volume Internet traffic) is actually not passed through the EPC but is offloaded at a "break-out point" closer to (or in) the access network. On the other hand, CONEX can also enable more dynamic decisions on what traffic to actually offload by considering congestion exposure in bulk traffic aggregates -- thus making traffic offload more effective.

3.6. Summary

In summary, the 3GPP EPS is a system architecture that can benefit from congestion exposure in multiple ways, as we have shown by this brief description of CONEX use cases in this environment. Dynamic traffic and congestion management is an acknowledged important requirement for the EPS, also illustrated by the current DPI-related work for EPS.

Moreover, we believe that networks such as an EPS mobile communication network would be quite amenable for deploying CONEX as a mechanism, since they represent clearly defined and well separated

operational domains, in which local CONEX deployment would be possible. Aside from roaming (which needs to be considered for a specific solution), such a deployment is fully under the control of a single operator, which can enable operator-local enhancement without the need for major changes to the architecture.

In 3GPP EPS, interfaces between all elements of the architecture are subject to standardization, including UE interfaces and eNodeB interfaces, so that a more general approach, involving more than one single operator's network, can be feasible as well.

4. CONEX in the EPS

The CONEX mechanism is still work in progress in the IETF working group. Still, we would like to discuss a few options for how such a mechanism (and possibly additional policing functions) could eventually be deployed in 3GPP's EPS. Note that this description of options is not intended as a complete set of possible approaches -- it is merely intended for discussing a few options. More details will be provided in a future revision of this document.

4.1. Possible Deployment Scenarios

There are different possible ways how CONEX functions on hosts and network elements can be used. For example, CONEX could be used for a limited part of the network only -- e.g., for the access network -- congestion exposure and sender adaptation could involve the mobile nodes or not, or, finally, the CONEX feedback loop could extend beyond a single operator's domain or not.

We present three different deployment scenarios for congestion exposure in the figures below:

1. In Figure 2 CONEX is supported by servers for sending data (here: web servers in the Internet and caches in an operator's network) but not by UEs (neither for receiving nor sending). An operator who chooses to run a policing function on the network ingress (e.g., on the P-GW) can still benefit from congestion exposure without requiring any change on UEs.
2. CONEX is universally employed between operators (as depicted in Figure 3), with an end-to-end CONEX feedback loop. Here, operators could still employ local policies, congestion accounting schemes etc., and they could use information about congestion contribution for determining interconnection agreements.

3. Isolated CONEX domains as depicted in Figure 4, CONEX is solely applied locally, in the operator network, and there is no end-to-end congestion exposure. This could be the case when CONEX is only implemented in a few networks, or when operators decide to not expose ECN and account for congestion for inter-domain traffic. Independent of the actual scenario, it is likely that there will be border gateways (as in today's deployments) that are associated with policing and accounting functions.

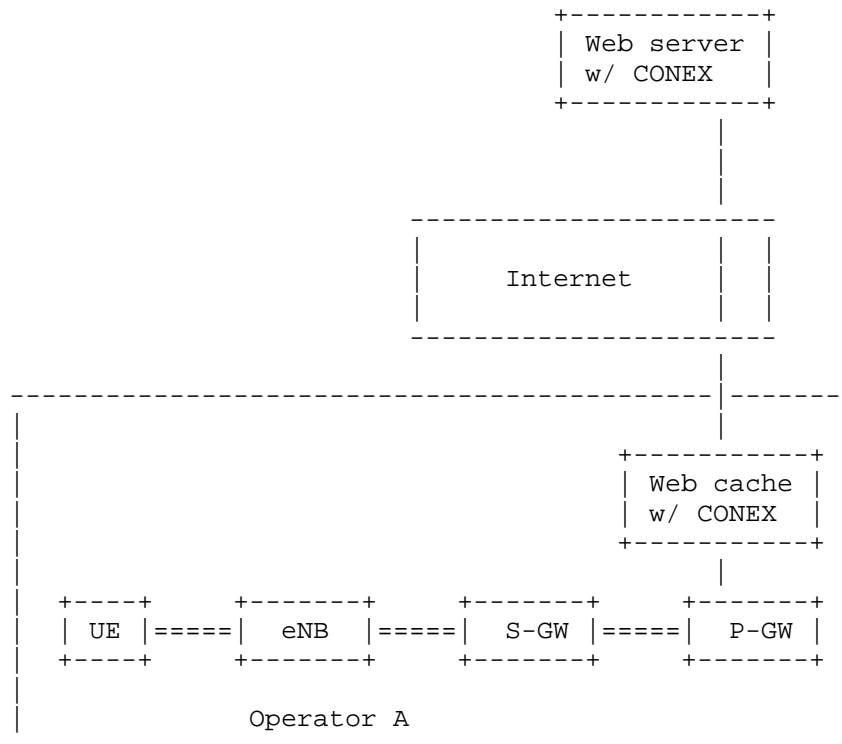


Figure 2: CONEX support on servers and caches

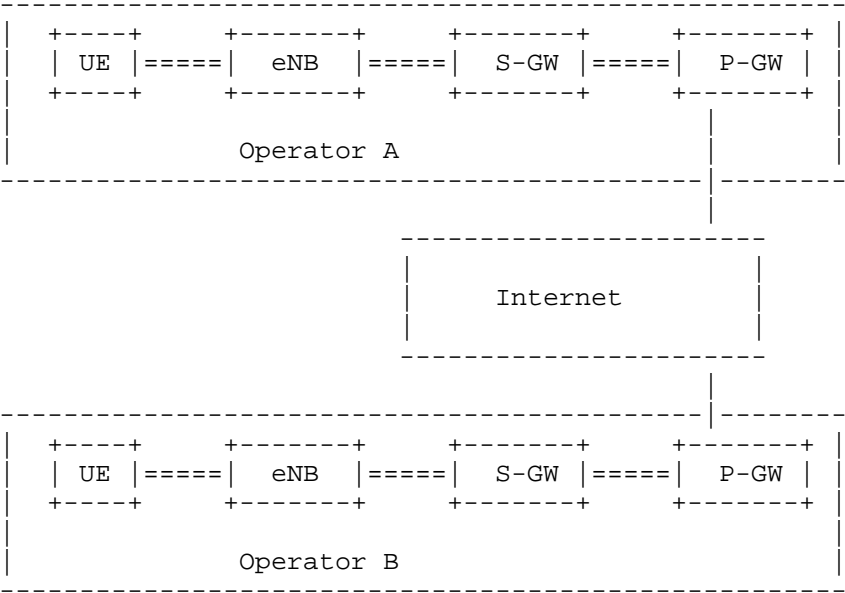


Figure 3: CONEX deployment across operator domains

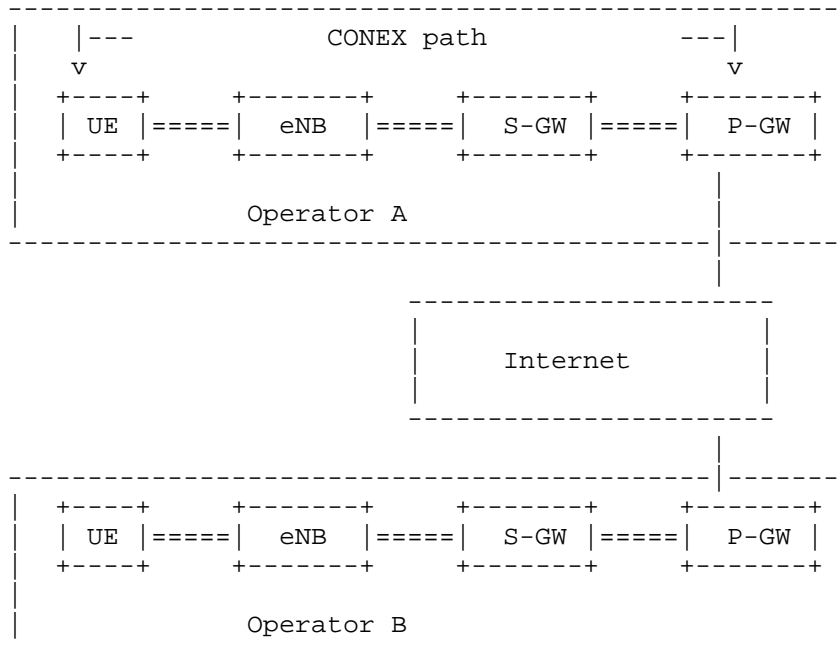


Figure 4: CONEX deployment in a single operator domain

We consider all three scenarios to be relevant and believe that all of them are within the scope of the CONEX WG charter. A more detailed description will be provided in a future version of this document.

4.2. Implementing CONEX Functions in the EPS

We expect a CONEX solution to consist of different functions that should be considered when implementing congestion exposure in 3GPP's EPS. [I-D.ietf-conex-abstract-mech] is describing the following congestion exposure components:

- o Modified senders that send congestion exposure information in response to congestion feedback).
- o Receivers that generate congestion feedback (leveraging existing behavior or requiring new functions).
- o Audit functions that audit CONEX signals against actual congestion, e.g., by monitoring flows or aggregate of flows.

- o Policy devices that monitor congestion exposure information and act on the flows according to the operator's policy.

Two aspects are important to consider: 1) how the CONEX protocol mechanisms would be implemented and what modifications to existing networks would be required and 2) where CONEX functional entities would be placed best (to allow for a non-invasive addition). We discuss these two aspects in the following sections.

4.2.1. CONEX Protocol Mechanisms

As described in [I-D.briscoe-conex-initial-deploy], the most important step in introducing CONEX (initially) is adding the congestion exposure functionality to senders. For an initial deployment, no further modification to senders and receivers would be required. Specifically, there is no fundamental dependency on ECN, i.e., CONEX can be introduced without requiring ECN to be implemented.

Congestion exposure information for IPv6 [I-D.ietf-conex-destopt] is contained in a destination option header field, which requires minimal changes at senders and nodes that want to assess path congestion -- and that does not affect non-CONEX nodes in a network.

In 3GPP networks, IP tunneling is used intensively, i.e., using either IP-in-GTP-U or PMIPv6 (i.e., IP-in-IP) tunnels. In general, the CONEX destination option of encapsulated packets should be made available for network nodes on the tunnel path, i.e., a tunnel ingress should copy the CONEX destination option field to the outer header.

For an effective and efficient capacity sharing, we envisage the deployment of ECN in conjunction with CONEX so that ECN-enabled receivers and senders get more accurate and more timely information about their flows congestion contribution. ECN is already partially introduced into 3GPP networks: Section 11.6 in [3GPP.36.300] specifies the usage of ECN for congestion notification on the radio link (between eNB and UE), and [3GPP.26.114] specifies how this can be leveraged for voice codec adaptation. A complete, end-to-end support of ECN would require specification of tunneling behaviour, which should be based on [RFC6040] (for IP-in-IP tunnels) and on [I-D.briscoe-tsvwg-ecn-encap-guidelines]. Specifically, a specification for tunneling ECN in GTP-U will be needed.

4.2.2. CONEX Functions in the Mobile Network

In the following, we discuss some possible placement strategies for CONEX functional entities (addressing both policing and auditing

functions) in the EPS and for possible optimizations for both the uplink and the downlink.

In general, CONEX information (exposed congestion) is declared by a sender and remains unchanged on the path, hence reading CONEX information (e.g., by policing functions) is placement-agnostic. Auditing CONEX normally requires assessing declared congestion contribution and current actual congestion. If the latter is, for example, done using ECN, such a function would best be placed at the end of the path.

In order to provide a comprehensive CONEX-based capacity management framework for EPS, it would be advantageous to consider user contribution to congestion for both the radio access and the core network. For a non-invasive introduction of CONEX, it can be beneficial to combine CONEX functions with existing logical EPS entities. For example, potential places for CONEX policing and auditing functions would then be eNBs, S-GWs or the P-GWs. Operator deployments may of course still provide additional intermediary CONEX-enabled IP network elements.

For a more specific discussion it will be beneficial to distinguish downlink and uplink traffic directions (also see [nec.globecom2010] for a more detailed discussion). In today's networks and usage models, downlink traffic is dominating (also reflected by the asymmetric capacity provided by the LTE radio interface). That does however not imply that uplink congestion is not an issue, since the asymmetric maximum bandwidth configuration can create a smaller bottleneck for uplink traffic -- and there are of course backhaul links, gateways etc. that could be overloaded as well.

For managing downlink traffic -- e.g., in scenarios such as the one depicted in Figure 2, operators can have different requirements for policing traffic. Although policing is in principle location-agnostic, it is important to consider requirements related to the EPS architecture (Figure 1) such as tunneling between P-GWs and eNBs. Policing can require access to subscriber information (e.g., congestion contribution quota) or user-specific accounting, which suggests that the CONEX function could be co-located with the P-GW that already has an interface towards the PCRF.

Still, policing can serve different purposes. For example, if the objective is to police bulk traffic induced by peer networks, additional monitoring functions can be placed directly at corresponding ingress points to monitor traffic and possible drive out-of-band functions such as triggering border contract penalties.

The auditing function which should be placed at the end of the path

(at least after/at the last bottleneck) would likely be placed best on the eNB (wireless base station).

For the uplink direction, there are naturally different options for designing monitoring and policy enforcement functions. A likely approach can be to monitor congestion exposure on central gateway nodes (such as P-GWs) that provide the required interfaces to the PCRF, but to perform policing actions in the access network, i.e., in eNBs, e.g., to police traffic at the ingress, before it reaches concentration points in the core network.

Such a setup would enable all the CONEX use cases described in Section 3, without requiring significant changes to the EPS architecture, while enabling operators to re-use existing infrastructure, specifically wireless base stations, PCRF and HSS systems.

For CONEX functions on elements such as the S-GWs and P-GWs, it is important to consider mobility and tunneling protocol requirements. LTE provides two alternative approaches: Proxy-Mobile-IPv6 (PMIPv6, [3GPP.23.402]) and GPRS Tunneling Protocol (GTP). For the propagation of congestion information (responses) tunneling considerations are therefore very important.

In general, policing will be done based on per-user (per subscriber) information such as congestion quota, current quota usage etc. and network operator policies, e.g., specifying how to react to persistent congestion contribution etc. In the EPS, per-user information is normally part of the user profile (stored in the HSS) that would be accessed by PCC entities such as the PCRF for dynamic updates, enforcement etc.

A more detailed description of the different approaches and their respective advantages will be provided in a future revision of this document.

5. Summary

We have shown how congestion exposure can be useful for efficient resource management in mobile communication networks. The premise for this discussion was the observation that data communication, specifically best-effort bulk data transmission, is becoming a commodity service whereas resources are obviously still limited -- which calls for efficient, scalable, yet effective capacity sharing in such networks.

CONEX can be a mechanism that enables such capacity sharing, while

allowing operators to apply these mechanisms in different ways, e.g., for implementing different use cases as described in Section 3. It is important to note that CONEX is fundamentally a mechanism that can be applied in different ways -- to realize different operators policies.

We have described a few possibilities for adding CONEX as a mechanism to 3GPP LTE-based networks and have shown how this could be done incrementally (starting with partial deployment). It is quite feasible that such partial deployments be done on a per-operator-domain basis, without requiring changes to standard 3GPP interfaces. For a network-wide deployment, e.g., with congestion exposure between operators, more considerations might be needed.

We have also identified a few implications/requirements that should be taken into consideration when enabling congestion exposure in such networks:

Performance: In mobile communication networks -- with more expensive resources and more stringent QoS requirements -- the feasibility of applying CONEX as well as its performance and deployment scenarios need to be examined closer. For instance, a mobile communication network may encounter longer delay and higher loss rates, which can impose specific requirements on the timeliness and accuracy of congestion exposure information.

Mobility: One of the unique characteristics in cellular network is the presence of user mobility compared to wired networks. As the user location changes, the same device can be connected to the network via different base stations (eNodeBs) or even go through switching gateways. Thus, the CONEX scheme must to be able to carry latest congestion information per user/flow across multiple network nodes in real time.

Multi-access: In cellular network, multiple access technologies can co-exist. In such cases, a user can use multiple access technologies for multiple applications or even a single application simultaneously. If the congestion policies are set based on each user, then CONEX should have the capability to enable information exchange across multiple access domains.

Tunneling: Both 3G and LTE networks make extensive usage of tunneling. The CONEX mechanism should be designed in a way to support usage with different tunneling protocols such as PMIPv6 and GTP. For ECN-based congestion notification, [RFC6040] specifies how the ECN field of the IP header should be constructed on entry and exit from IP-in-IP tunnels, and [I-D.briscoe-tsvwg-ecn-encap-guidelines] provides guidelines for

adding congestion notification to protocols that encapsulate IP.

Roaming: Independent of the specific architecture, mobile communication networks typically differentiate between non-roaming and roaming scenarios. Roaming scenarios are typically more demanding regarding implementing operator policies, charging etc. It can be expected that this would also hold for deploying CONEX. A more detailed analysis of this problem will be provided in a future revision of this document.

It is important to note that CONEX is intended to be used as a supplement and not a replacement to the existing QoS mechanisms in mobile networks. For example, CONEX deployed in 3GPP mobile networks can provide useful input to the existing 3GPP PCC mechanisms by supplying more dynamic network information to supplement the fairly static information used by the PCC. This would enable the mobile network to make better policy control decisions than is possible with only static information.

6. IANA Considerations

No IANA considerations.

7. Security Considerations

Security considerations for applying CONEX to EPS include, but are not limited to, the security considerations that apply to the CONEX protocols.

8. Informative References

[3GPP.23.203]

3GPP, "Policy and charging control architecture", 3GPP TS 23.203 10.7.0, June 2012.

[3GPP.23.402]

3GPP, "Architecture enhancements for non-3GPP accesses", 3GPP TS 23.402 10.7.0, March 2012.

[3GPP.23.829]

3GPP, "Local IP Access and Selected IP Traffic Offload (LIPA-SIPTO)", 3GPP TR 23.829 10.0.1, October 2011.

[3GPP.26.114]

3GPP, "IP Multimedia Subsystem (IMS); Multimedia

telephony; Media handling and interaction", 3GPP TS 26.114 10.4.0, June 2012.

[3GPP.29.060]

3GPP, "General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface", 3GPP TS 29.060 3.19.0, March 2004.

[3GPP.29.274]

3GPP, "3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3", 3GPP TS 29.274 10.7.0, June 2012.

[3GPP.36.300]

3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2", 3GPP TS 36.300 10.8.0, July 2012.

[I-D.briscoe-conex-initial-deploy]

Briscoe, B., "Initial Congestion Exposure (ConEx) Deployment Examples", draft-briscoe-conex-initial-deploy-02 (work in progress), March 2012.

[I-D.briscoe-tsvwg-ecn-encap-guidelines]

Briscoe, B., "Guidelines for Adding Congestion Notification to Protocols that Encapsulate IP", draft-briscoe-tsvwg-ecn-encap-guidelines-00 (work in progress), March 2011.

[I-D.ietf-conex-abstract-mech]

Mathis, M. and B. Briscoe, "Congestion Exposure (ConEx) Concepts and Abstract Mechanism", draft-ietf-conex-abstract-mech-04 (work in progress), March 2012.

[I-D.ietf-conex-concepts-uses]

Briscoe, B., Woundy, R., and A. Cooper, "ConEx Concepts and Use Cases", draft-ietf-conex-concepts-uses-04 (work in progress), March 2012.

[I-D.ietf-conex-destopt]

Krishnan, S., Kuehlewind, M., and C. Ucendo, "IPv6 Destination Option for Conex", draft-ietf-conex-destopt-02 (work in progress), March 2012.

- [RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, November 2010.
- [nec.euronf-2011]
Mir, Kutscher, and Brunner, "Congestion Exposure in Mobility Scenarios", in proceedings of 7th EURO-NF CONFERENCE ON NEXT GENERATION INTERNET, June 2011.
- [nec.globecom2010]
Kutscher, Lundqvist, and Mir, "Congestion Exposure in Mobile Wireless Communications", in proceedings of IEEE GLOBECOM 2010, December 2010.
- [raghavan2007]
Raghavan, Vishwanath, Ramabhadran, Yocum, and Snoeren, "Cloud Control with Distributed Rate Limiting", in proceedings of ACM SIGCOMM 2007, 2007.
- DOI: <http://doi.acm.org/10.1145/1282427.1282419>

Appendix A. Acknowledgments

We would like to thank Bob Briscoe and Ingemar Johansson for their support in shaping the overall idea and in improving the draft by providing constructive comments.

Authors' Addresses

Dirk Kutscher
NEC
Kurfuersten-Anlage 36
Heidelberg,
Germany

Phone:
Email: kutscher@neclab.eu

Faisal Ghias Mir
NEC
Kurfuersten-Anlage 36
Heidelberg,
Germany

Phone:
Email: faisal.mir@neclab.eu

Rolf Winter
NEC
Kurfuersten-Anlage 36
Heidelberg,
Germany

Phone:
Email: winter@neclab.eu

Suresh Krishnan
Ericsson
8400 Blvd Decarie
Town of Mount Royal, Quebec
Canada

Phone:
Email: suresh.krishnan@ericsson.com

Ying Zhang
Ericsson
200 Holger Way
San Jose, CA 95134
USA

Phone:
Email: ying.zhang@ericsson.com

Carlos J. Bernardos
Universidad Carlos III de Madrid
Av. Universidad, 30
Leganes, Madrid 28911
Spain

Phone: +34 91624 6236
Email: cjbc@it.uc3m.es
URI: <http://www.it.uc3m.es/cjbc/>

Congestion Exposure (ConEx)
Internet-Draft
Intended status: Experimental
Expires: November 11, 2012

M. Kuehlewind, Ed.
University of Stuttgart
R. Scheffenegger
NetApp, Inc.
May 10, 2012

TCP modifications for Congestion Exposure
draft-ietf-conex-tcp-modifications-02

Abstract

Congestion Exposure (ConEx) is a mechanism by which senders inform the network about the congestion encountered by previous packets on the same flow. This document describes the necessary modifications to use ConEx with the Transmission Control Protocol (TCP).

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 November 11, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Language	3
2. Sender-side Modifications	3
3. Accounting congestion	4
3.1. ECN	5
3.1.1. Accurate ECN feedback	6
3.1.2. Classic ECN support	6
3.2. Loss Detection with/without SACK	8
4. Setting the ConEx IPv6 Bits	8
4.1. Setting the E and the L Bit	9
4.2. Credit Bits	9
4.3. Loss of ConEx information	10
5. Timeliness of the ConEx Signals	10
6. Acknowledgements	11
7. IANA Considerations	11
8. Security Considerations	11
9. References	11
9.1. Normative References	11
9.2. Informative References	12
Appendix A. Revision history	12
Authors' Addresses	14

1. Introduction

Congestion Exposure (ConEx) is a mechanism by which senders inform the network about the congestion encountered by previous packets on the same flow. This document describes the necessary modifications to use ConEx with the Transmission Control Protocol (TCP). The ConEx signal is based on loss or ECN marks [RFC3168] as a congestion indication. This congestion information is retrieved by the sender based on existing feedback mechanisms from the receiver to the sender in TCP.

With standard TCP without Selective Acknowledgments (SACK) [RFC2018] the actual number of losses is hard to detect, thus we recommend to enable SACK when using ConEx. However, we discuss both cases, with and without SACK support, later on.

Explicit Congestion Notification (ECN) is defined in such a way that only a single congestion signal is guaranteed to be delivered per Round-trip Time (RTT) from the receiver to the sender. For ConEx a more accurate feedback signal would be beneficial. Such an extension to ECN is defined in a separate document [draft-kuehlewind-conex-accurate-ecn], as it can also be useful for other mechanisms, as e.g. [DCTCP] or whenever the congestion control reaction should be proportional to the experienced congestion. ConEx also works with classic ECN but it is less accurate when multiple congestion markings occur within on RTT.

ConEx is currently/will be defined as an destination option for IPv6. The use of four bits have been defined, namely the X (ConEx-capable), the L (loss experienced), the E (ECN experienced) and C (credit) bit.

1.1. Requirements Language

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

2. Sender-side Modifications

A ConEx sender MUST negotiate for both SACK and ECN or the more accurate ECN feedback in the TCP handshake if these TCP extension are available at the sender. Depending on the capability of the receiver, the following operation modes exist:

- o Full-ConEx (SACK and accurate ECN feedback)

- o accECN-ConEx (no SACK but accurate ECN feedback)
- o ECN-ConEx (no SACK and no accurate ECN feedback but 'classic' ECN)
- o SACK-ECN-ConEx (SACK and 'classic' instead of accurate ECN)
- o SACK-ConEx (SACK but no ECN at all)
- o Basic-ConEx (neither SACK nor ECN)

A ConEx sender MUST expose congestion to the network according to the congestion information received by ECN or based on loss information provided by the TCP feedback loop. A TCP sender SHOULD account congestion byte-wise (and not packet-wise). A sender MUST mark subsequent packets (after the congestion notification) with the respective ConEx bit in the IP header.

With SACK only the number of lost bytes is known, but not the number of packets carrying these bytes. With classic ECN only an indication is given that a marking occurred which is not giving an exact number of bytes nor packets. As network congestion is usually byte-congestion, the exact number of bytes should be taken into account if available to make the ConEx signal as exact as possible.

The congestion accounting based on different operation modes is described in the next section and the handling of the IPv6 bits itself in the subsequent section afterwards.

3. Accounting congestion

A TCP sender SHOULD account congestion byte-wise (and not packet-wise) based the congestion information received by ECN or loss detection provided by TCP. For this purpose a TCP sender will maintain two different counters for number outstanding bytes that need to be ConEx marked either with the E bit or the L Bit.

The outstanding bytes accounted based on ECN feedback information are maintained in the congestion exposure gauge (CEG). The accounting of these bytes from the ECN feedback is explained in more detail next.

The outstanding bytes for congestion indications based on loss are maintained in the loss exposure gauge (LEG) and the accounting is explained in subsequent to the CEG accounting.

The subtraction of bytes which have been ConEx marked from both counters is explained in the next section.

Usually all bytes of an IP packet must be accounted. Therefore the sender SHOULD take the headers into account. If equal sized packets or at least equally distributed packet sizes can be assumed the sender MAY only account the TCP payload bytes, as the ConEx marked packets as well as the original packets causing the congestion will both contain about the same number of headers.

If a sender sends different sized packets with unequally distributed packet sizes, the sender might be able to reconstruct the exact number of headers based on information which packet sizes has been sent in the last RTT. Otherwise if no additional information is available the worst case number of headers SHOULD be estimated in a conservative way based on a minimum packet size (of all packets sent in the last RTT).

3.1. ECN

ECN is an IP/TCP mechanism that allows network nodes to mark packets with the Congestion Experienced (CE) mark instead of (early) dropping them when congestion occurs. As soon as a CE mark is seen at the receiver, with classic ECN it will feed this information back to the sender by setting the Echo Congestion Experienced (ECE) bit in the TCP header until a packet with Congestion Window Reduced (CWR) bit in the TCP header is received to acknowledge the reception of the congestion notification. The sender sets the CWR bit in the TCP header once when the first ECE of a congestion notification is received.

A receiver can support the accurate ECN feedback scheme, the 'classic' ECN or neither. In the case ECN is not supported at all, the transport is not ECN-capable and no ECN marks will occur, thus the E bit will never be set. In the other cases a ConEx sender MUST maintain a gauge for the number of outstanding bytes that have to be ConEx marked with the E bit, the congestion exposure gauge (CEG).

The CEG is increased when ECN information is received from an ECN-capable receiver supporting the 'classic' ECN scheme or the accurate ECN feedback scheme. When the ConEx sender receives an ACK indicating one or more segments were received with a CE mark, CEG is increased by the appropriate number of bytes.

In case of duplicate acknowledgements the number of acknowledged bytes will be zero even though (CE marked) data has been received. Therefore, we calculated a variable DeliveredData. DeliveredData covers the total number of bytes that the current ACK indicates have been delivered to the receiver, relative to all past ACKs. With SACK, DeliveredData is increased by the number of bytes given by changes in the SACK information. Note the change in in the SACK

information can also be negative if the number of acknowledged bytes increases. Without SACK, DeliveredData is estimated to be 1 SMSS on duplicate acknowledgements, and on a subsequent partial or full ACK, DeliveredData is estimated to be the change in acknowledged bytes, minus one SMSS for each preceding duplicate ACK.

$$\text{DeliveredData} = \text{acked_bytes} + \text{SACK_diff} + (\text{is_dup}) * 1\text{SMSS} - (\text{is_after_dup}) * \text{num_dup} * 1\text{SMSS}$$

The two cases, with and without more accurate ECN depending on the receiver capability, are discussed in the following sections.

3.1.1. Accurate ECN feedback

With a more accurate ECN feedback scheme either the number of marked packets/received CE marks or directly the number of marked bytes is known. In the later case the CEG can directly be increased by the number of marked bytes. Otherwise if D is assumed to be the number of marks, the gauge CEG has to be increased by the amount of bytes sent which were marked:

$$\text{CEG} += \min(\text{SMSS} * D, \text{DeliveredData})$$

3.1.2. Classic ECN support

A ConEx sender that communicates with a classic ECN receiver (conforming to [RFC3168] or [RFC5562]) MAY run in one of these modes:

- o Full compliance mode:

The ConEx sender fully conforms to all the semantics of the ECN signaling as defined by [RFC5562]. In this mode, only a single congestion indication can be signaled by the receiver per RTT. Whenever the ECE flag toggles from "0" to "1", the gauge CEG is increased at maximum by the SMSS:

$$\text{CEG} += \min(\text{SMSS}, \text{DeliveredData})$$

Note that under severe congestion, a session adhering to these semantics may not provide enough ConEx marks. This may cause appropriate sanctions by an audit device in a ConEx enabled network.

- o Simple compatibility mode:

The sender will set the CWR permanently to force the receiver to signal only one ECE per CE mark. Unfortunately, the use of delayed ACKs [RFC5681], as it is usually done today, will prevent

a feedback of every CE mark. An CWR confirmation will be received before the ECE can be sent out with the next ACK. With an ACK rate of M, about $M-1/M$ CE indications will not be signaled back by the receiver (e.g. 50% with $M=2$ for delayed ACKs). Thus, in this mode the ConEx sender MUST increase CEG as if M congestion notification were received for each received ECE signal:

```
CEG += min(M*SMSS, DeliveredData + (M-1)*SMSS)
```

In case of a congestion event with low congestion (that means when only a very smaller number of packets get marked), the sender might miss the whole congestion event. On average the sender will send sufficient ConEx marks due to the scheme proposed above but these ConEx marks might be shifted in time. Regarding congestion control it is not a general problem to miss a congestion event as, by chance, a marking scheme in the network node might also miss a certain flow. In the case where no other flow is reacting, the congestion level will increase and it will get more likely that the congestion feedback is delivered. To provide a fair share over time, a TCP sender implementing this simple ECN compatibility mode could react more strongly when receiving an ECN feedback signal. This of course depends on the congestion control used.

o Advanced compatibility mode:

To avoid the loss of ECN feedback information in the proposed simple compatibility mode, a sender could set CWR only on those data segments, that will actually trigger a (delayed) ACK. The sender would need an additional control loop to estimate which data segment will trigger an ACK. Such a more sophisticated heuristic could extract congestion notifications more timely. In addition, if this advanced compatibility mode is used, further heuristics SHOULD be implemented, to determine the value of each ECE notification. E.g. for each consecutive ACK received with the ECE flag set, CEG should be increased by $\min(M*SMSS, DeliveredData)$. Else if the predecessor ACK was received with the ECE flag cleared, CEG need only be increased at maximum by one SMSS:

```
if previous_marked: CEG += min( M*SMSS, DeliveredData)
else: CEG += min(SMSS, DeliveredData)
```

This heuristic is conservative during more serious congestion, and more relaxed at low congestion levels.

3.2. Loss Detection with/without SACK

For all the data segments that are determined by a ConEx sender as lost, (at least) the same number of TCP payload bytes MUST be sent with the ConEx L bit set. Loss detection typically happens by use of duplicate ACKs, or the firing of the retransmission timer. A ConEx sender MUST maintain a loss exposure gauge (LEG), indicating the number of outstanding bytes that must be sent with the ConEx L bit. When a data segment is retransmitted, LEG will be increased by the size of the TCP payload packet containing the retransmission, assuming equal sized segments such that the retransmitted packet will have the same number of header as the original ones. When sending subsequent segments, the ConEx L bit is set as long as LEG is positive, and LEG is decreased by the size of the sent TCP payload with the ConEx L bit set.

Any retransmission may be spurious. To accommodate that, a ConEx sender SHOULD make use of heuristics to detect such spurious retransmissions (e.g. F-RTO [RFC5682], DSACK [RFC3708], and Eifel [RFC3522], [RFC4015]). When such a heuristic has determined, that a certain number of packets were retransmitted erroneously, the ConEx sender should subtract the payload size of these TCP packets from LEG.

Note that the above heuristics delays the ConEx signal by one segment, and also decouples them from the retransmissions themselves, as some control packets (e.g. pure ACKs, window probes, or window updates) may be sent in between data segment retransmissions. A simpler approach would be to set the ConEx signal for each retransmitted data segment. However, it is important to remember, that a ConEx signal and TCP segments do not natively belong together.

If SACK is not available or SACK information has been reset for any reason, spurious retransmission are more likely. In this case it might be valuable to slightly delay the ConEx loss feedback until a spurious retransmission might be detected. But the ConEx signal MUST NOT be delayed more than one RTT.

4. Setting the ConEx IPv6 Bits

ConEx is currently/will be defined as an destination option for IPv6. The use of four bits have been defined, namely the X (ConEx-capable), the L (loss experienced), the E (ECN experienced) and C (credit) bit.

By setting the X bit a packet is marked as ConEx-capable. All packets carrying payload MUST be marked with the X bit set including retransmissions. No congestion feedback information are available

about control packets as pure ACKs which are not carrying any payload. Thus these packet should not be taken into account when determining ConEx information. These packet MUST carry a ConEx Destination Option with the X bit unset.

4.1. Setting the E and the L Bit

As long as the CEG or LEG counter is positive, ConEx-capable packets SHOULD be marked with E or L respectively, and the CEG or LEG counter is decreased by the TCP payload bytes carried in this packet. If the CEG or LEG counter is negative, the respective counter SHOULD be reset to zero within one RTT after it was decreased the last time or one RTT after recovery if no further congestion occurred.

4.2. Credit Bits

The ConEx abstract mechanism requires that the transport SHOULD signal sufficient credit in advance to cover any reasonably expected congestion during its feedback delay. To be very conservative the number of credits would need to equal the number of packets in flight, as every packet could get lost or congestion marked. With a more moderate view, only an increase in the sending rate should cause congestion.

TODO: More general description to maintain always at least $\text{flight_size} - \text{flight_size_prev}$ credits. -> Can the number of credits in the audit decrease?

For TCP sender using the [RFC5681] congestion control algorithm, we recommend to only send credit in Slow Start, as in Congestion Avoidance an increase of one segment per RTT should only cause a minor amount of congestion marks (usually at max one). If a more aggressive congestion control is used, a sufficient amount of credits need to be set.

In TCP Slow Start the sending rate will increase exponentially and that means double every RTT. Thus the number of credits should equal half the number of packets in flight in every RTT. Under the assumption that all marks will not get invalid for the whole Slow Start phase, marks of a previous RTT have to be summed up. Thus the marking of every fourth packet will allow sufficient credits in Slow Start as it can be seen in Figure 1.

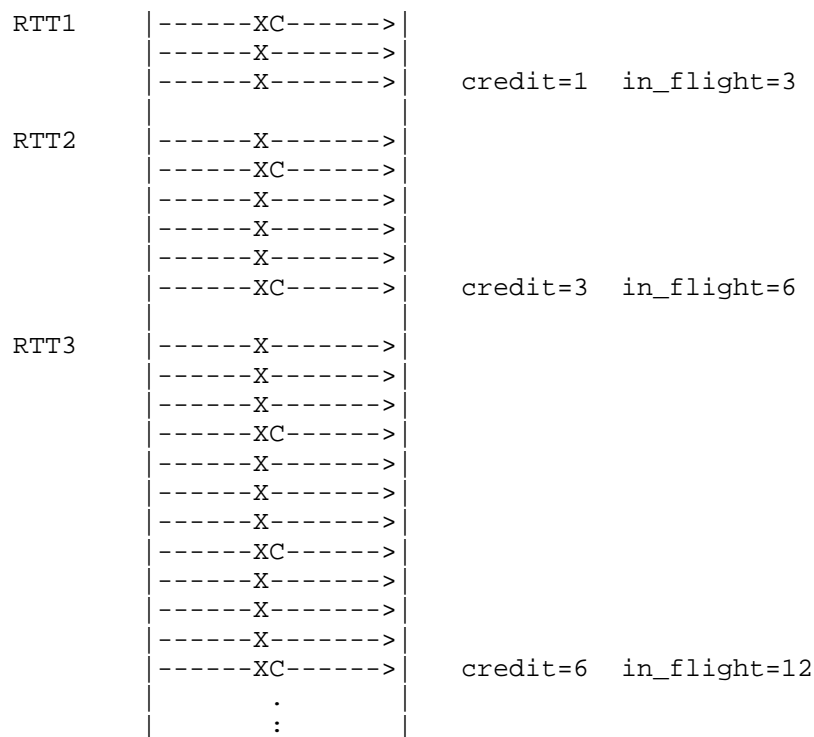


Figure 1: Credits in Slow Start (with an initial window of 3)

4.3. Loss of ConEx information

The audit can have wrong infos if e.g. ConEx got lost on the channel (or a wrong number of ConEx marking has been estimated by the sender due to a lack of feedback information). In this case audit might penalize a sender wrongly. TODO: Further action needed by the sender?

5. Timeliness of the ConEx Signals

ConEx signals will anyway be evaluated with a slight time delay of about one RTT by a network node. Therefore, it is not absolutely necessary to immediately signal ConEx bits when they become known (e.g. L and E bits), but a sender SHOULD send the ConEx signaling with the next available packet. In cases where it is preferable to slightly delay the ConEx signal, the sender MUST NOT delay the ConEx signal more than one RTT.

Multiple ConEx bits may become available for signaling at the same

time, for example when an ACK is received by the sender, that indicates that at least one segment has been lost, and that one or more ECN marks were received at the same time. This may happen during excessive congestion, where buffer queues overflow and some packets are marked, while others have to be dropped nevertheless. Another possibility when this may happen are lost ACKs, so that a subsequent ACK carries summary information not previously available to the sender.

6. Acknowledgements

The authors would like to thank Bob Briscoe who contributed with this initial ideas and valuable feedback.

7. IANA Considerations

This document does not have any requests to IANA.

8. Security Considerations

With some of the advanced ECN compatibility modes it is possible to miss congestion notifications. Thus a sender will not decrease its sending rate. If the congestion is persistent, the likelihood to receive a congestion notification increases. In the worst case the sender will still react correctly to loss. This will prevent a congestion collapse.

9. References

9.1. Normative References

- [RFC2018] Mathis, M., Mahdavi, J., Floyd, S., and A. Romanow, "TCP Selective Acknowledgment Options", RFC 2018, October 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.
- [RFC5562] Kuzmanovic, A., Mondal, A., Floyd, S., and K. Ramakrishnan, "Adding Explicit Congestion Notification (ECN) Capability to TCP's SYN/ACK Packets", RFC 5562,

June 2009.

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

9.2. Informative References

- [DCTCP] Alizadeh, M., Greenberg, A., Maltz, D., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., and M. Sridharan, "DCTCP: Efficient Packet Transport for the Commoditized Data Center", Jan 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.
- [RFC3522] Ludwig, R. and M. Meyer, "The Eifel Detection Algorithm for TCP", RFC 3522, April 2003.
- [RFC3708] Blanton, E. and M. Allman, "Using TCP Duplicate Selective Acknowledgement (DSACKs) and Stream Control Transmission Protocol (SCTP) Duplicate Transmission Sequence Numbers (TSNs) to Detect Spurious Retransmissions", RFC 3708, February 2004.
- [RFC4015] Ludwig, R. and A. Gurtov, "The Eifel Response Algorithm for TCP", RFC 4015, February 2005.
- [RFC5682] Sarolahti, P., Kojo, M., Yamamoto, K., and M. Hata, "Forward RTO-Recovery (F-RTO): An Algorithm for Detecting Spurious Retransmission Timeouts with TCP", RFC 5682, September 2009.
- [draft-kuehlewind-conex-accurate-ecn]
Kuehlewind, M. and R. Scheffenegger, "Accurate ECN Feedback in TCP", draft-kuehlewind-conex-accurate-ecn-00 (work in progress), Jun 2011.

Appendix A. Revision history

RFC Editor: This section is to be removed before RFC publication.

00 ... initial draft, early submission to meet deadline.

01 ... refined draft, updated LEG "drain" from per-packet to RTT-

based.

Authors' Addresses

Mirja Kuehlewind (editor)
University of Stuttgart
Pfaffenwaldring 47
Stuttgart 70569
Germany

Email: mirja.kuehlewind@ikr.uni-stuttgart.de

Richard Scheffenegger
NetApp, Inc.
Am Euro Platz 2
Vienna, 1120
Austria

Phone: +43 1 3676811 3146
Email: rs@netapp.com

