

Transport Area Working Group
Internet-Draft
Intended status: Informational
Expires: May 5, 2019

B. Briscoe
CableLabs
November 1, 2018

**Interactions between Low Latency, Low Loss, Scalable Throughput (L4S)
and Differentiated Services
draft-briscoe-tsvwg-l4s-diffserv-02**

Abstract

L4S and Diffserv offer somewhat overlapping services (low latency and low loss), but bandwidth allocation is out of scope for L4S. Therefore there is scope for the two approaches to complement each other, but also to conflict. This informational document explains how the two approaches interact, how they can be arranged to complement each other and in which cases one can stand alone without needing the other.

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 <https://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 May 5, 2019.

Copyright Notice

Copyright (c) 2018 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 (<https://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](#) [2](#)
- [1.1. Terminology](#) [3](#)
- [1.2. Document Roadmap](#) [4](#)
- [2. Architectural Comparison of L4S and Diffserv](#) [4](#)
- [2.1. Overlaps between L4S and Diffserv](#) [4](#)
- [2.2. Differences between L4S and Diffserv](#) [4](#)
- [3. Low Latency Diffserv Classes within a DualQ Bandwidth Pool](#) [5](#)
- [4. DualQ Bandwidth Pool within a Hierarchy of \(Diffserv\) Bandwidth Queues](#) [9](#)
- [4.1. DualQ Complemented by an Assured Bandwidth Service](#) [10](#)
- [4.2. DualQ Complemented by a Guaranteed Low Latency Service](#) [12](#)
- [4.3. DualQ Complemented by a Scavenger Service](#) [13](#)
- [5. Coupling More than Two AQMs within a Bandwidth Pool](#) [14](#)
- [6. Applicability of Coupled AQM to Global DiffServ PHBs](#) [14](#)
- [7. Best Practice for Classification and Marking](#) [16](#)
- [7.1. Never Re-Mark a DSCP](#) [16](#)
- [7.2. Classification Order](#) [16](#)
- [7.2.1. Classification Order: Problem](#) [17](#)
- [7.2.2. Classification Order: Solutions](#) [17](#)
- [8. Policing and Traffic Conditioning](#) [17](#)
- [9. IANA Considerations](#) [18](#)
- [10. Security Considerations](#) [18](#)
- [11. Comments Solicited](#) [18](#)
- [12. Acknowledgements](#) [18](#)
- [13. References](#) [18](#)
- [13.1. Normative References](#) [18](#)
- [13.2. Informative References](#) [18](#)
- [Appendix A. Open Issues](#) [20](#)
- [Author's Address](#) [20](#)

1. Introduction

The Low Latency Low Loss Scalable throughput (L4S) Internet service [[I-D.ietf-tsvwg-l4s-arch](#)] provides a new Internet service that could eventually replace best efforts, but with ultra-low queuing delay and loss. A structure called the Dual-Queue Coupled AQM manages to provide the L4S service alongside a second queue for Classic Internet traffic, but without prejudging the bandwidth allocations between them. L4S is orthogonal to allocation of bandwidth, so it can be complemented by various bandwidth allocation approaches without prejudging which one.

Briscoe

Expires May 5, 2019

[Page 2]

The Differentiated Services (Diffserv) architecture [[RFC2475](#)] provides for various service classes, some defined globally, others defined locally per network domain. Certain of these service classes offer low latency and low loss, as well as differentiated allocation of bandwidth.

Thus, L4S and Diffserv offer somewhat overlapping services (low latency and low loss), but bandwidth allocation is out of scope for L4S. Therefore there is scope for the two approaches to complement each other, but also to conflict. This informational document explains how the two approaches interact, how they can be arranged to complement each other and in which cases one can stand alone without needing the other.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)]. In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC-2119](#) significance.

Classic service: The 'Classic' service is intended for all the congestion control behaviours that currently co-exist with TCP Reno [[RFC5681](#)] (e.g. TCP Cubic, Compound, SCTP, etc).

Low-Latency, Low-Loss and Scalable (L4S) service: The 'L4S' service is intended for traffic from scalable congestion control algorithms such as Data Centre TCP [[RFC8257](#)]. But it is also more general--it will allow a set of congestion controls with similar scaling properties to DCTCP to evolve.

Both Classic and L4S services can cope with a proportion of unresponsive or less-responsive traffic as well (e.g. DNS, VoIP, etc).

Pure L4S: L4S without unresponsive traffic.

Scalable Congestion Control: See [[I-D.ietf-tsvwg-l4s-arch](#)] for definition.

Classic Congestion Control: See [[I-D.ietf-tsvwg-l4s-arch](#)] for definition.

DualQ: Abbreviation for Dual-Queue Coupled AQM [[I-D.ietf-tsvwg-aqm-dualq-coupled](#)], which is not a specific AQM, but a framework for coupling two AQMs in order to provide L4S

service while doing no harm to 'Classic' traffic from traditional sources.

ECN field: The Explicit Congestion Notification field [[RFC3168](#)] in the IP header (v4 or v6). [[RFC8311](#)] has relaxed some of the restrictions that [RFC 3168](#) placed on the use of ECN, in order to enable experiments like L4S, among others.

Site: A home, mobile device, small enterprise or campus, where the network bottleneck is typically the access link to the site. Not all network arrangements fit this model but it is a useful, widely applicable generalisation.

1.2. Document Roadmap

{ToDo}

2. Architectural Comparison of L4S and Diffserv

This section compares the L4S architecture [[I-D.ietf-tsvwg-l4s-arch](#)] with the Diffserv architecture [[RFC2475](#)].

L4S uses an identifier [[I-D.ietf-tsvwg-ecn-l4s-id](#)] in the ECN field in IP packet headers that is orthogonal to the Diffserv field [[RFC2474](#)]. This is because the two approaches can either overlap or complement each other, as outlined in the following two subsections.

2.1. Overlaps between L4S and Diffserv

L4S provides a low queuing latency, low loss Internet Service. Specific Diffserv service classes also provide low latency and low loss.

This means that it is possible to mix traffic from certain Diffserv classes in the same queue as L4S traffic (see [Section 3](#)).

2.2. Differences between L4S and Diffserv

Bandwidth allocation: L4S is orthogonal to allocation of bandwidth, so it can be complemented by various bandwidth allocation approaches without prejudging which one. In contrast, with Diffserv it was never possible to completely separate control of latency and loss from allocation of bandwidth. The only bandwidth-related aspect of L4S is that it ensures that the capacity seeking behaviour of end-systems can scale with increasing flow rate.

Differentiation vs. General improvement: Diffserv concerns give and take of bandwidth, latency and loss between traffic classes. In contrast, the separation of L4S from Classic traffic in separate queues concerns incremental deployment of a general improvement in latency and loss, without taking from the other queue.

Open vs. closed loop control: The Diffserv architecture requires the source to keep traffic within a contract and, failing that, it has mechanisms to enforce the contract. In this respect, Diffserv is an open-loop control system that is primarily concerned with keeping traffic within capacity limits. Nonetheless, there is an element of closed-loop control in Diffserv. The weighted AQM (e.g. WRED) used for Assured Forwarding [[RFC2597](#)] expects traffic to seek to fill capacity and exploits the response to feedback of congestion controllers at traffic sources (closed-loop). Nonetheless, the Diffserv architecture still provides for traffic conditioners that tag traffic that is outside the bandwidth contract for each AF class (open-loop). Then out-of-contract traffic can be discarded if it would otherwise lead to congestion.

L4S uses a similar closed-loop mechanism to the weighted AQM used in Diffserv AF in order to ensure roughly equal per-flow throughput between the L4S and Classic queues. That is, L4S relies on the source's closed-loop response to feedback, not any open-loop obligation of each source to keep within a traffic contract. With L4S, any enforcement of per-flow throughput (whether open-loop or closed) is set aside as a separate issue that may or may not be addressed by separate mechanisms, dependent on policy.

Per bottleneck vs. per domain: L4S can be independently and incrementally deployed at certain bottlenecks. In contrast a Diffserv system is domain-based consisting of the per-hop behaviour of interior nodes and the traffic conditioning behaviour of boundary nodes, which have to be deployed as a coordinated whole.

Degree of multiplexing: Diffserv components such as traffic conditioning are less applicable in access networks where statistical multiplexing is low, whereas L4S was initially designed for access networks, but is also applicable at larger pinch-points (e.g. public peerings).

3. Low Latency Diffserv Classes within a DualQ Bandwidth Pool

The experimental Dual-Queue Coupled AQM [[I-D.ietf-tsvwg-aqm-dualq-coupled](#)] consists of a pair of queues. One provides a low latency low loss service but both have full access to

the same pool of bandwidth. When Diffserv was defined no mechanism like this was available that could provide low latency without also requiring bandwidth controls. All Diffserv's mechanisms for low latency and low loss use some form of priority over bandwidth, then apply a bandwidth constraint to prevent the lower priority traffic from being starved.

This Diffserv bandwidth constraint has a flip side - it can also provide a bandwidth assurance. However, in turn, bandwidth assurance has both positive and negative aspects. It certainly prevents other traffic encroaching on the bandwidth of the low latency class, but it also carves off a partition within which low latency sessions are more prone to encroach on each other.

The DualQ offers an alternative where low latency traffic can access the whole pool of bandwidth (in effect, the largest possible bandwidth constraint). This is expected to be preferred by many network operators and users who would rather not set a bandwidth limit for their low latency traffic - particularly at links in access networks where the very low level of flow multiplexing makes the bandwidth shares of different traffic classes nearly impossible to predict. Nonetheless, if a bandwidth partition is required for bandwidth assurance purposes, it can still be provided separately (see [Section 4](#)).

The DualQ classifies packets with the ECN field set to ECT(1) or CE into the low latency low loss (L) queue. The L queue maintains a low latency low loss service primarily because an L4S source paces its packets and is linearly responsive to ECN markings, which earns it the right to set the ECT(1) codepoint [[I-D.ietf-tsvwg-ecn-l4s-id](#)] [[RFC8311](#)].

Nonetheless, a low level of non-L4S traffic can share the L queue without compromising the low latency and low loss of the service. Certain existing Diffserv classes are already intended as low latency and low loss services. An operator could use the DualQ instead of traditional Diffserv queues to give a few of these classes the benefit of low latency and access to the whole pool of bandwidth.

However, that would only be safe for those Diffserv service classes that would not risk ruining the low latency of the service. Therefore, an operator must take care to only classify a Diffserv traffic class into the L queue if it is expected to send smoothly without multi-packet bursts. Below we give examples of classes that should (and should not) be safe to mix into the L queue.

Table 1 lists the Diffserv service classes that have been allocated global use Diffserv codepoints (DSCPs) from Pool 1. They are

described in [RFC 4594](#) [[RFC4594](#)] and its updates ([\[RFC5865\]](#) and [\[I-D.ietf-tsvwg-le-phb\]](#) so far). An operator that only deploys a DualQ [\[I-D.ietf-tsvwg-aqm-dualq-coupled\]](#) but not the relevant Diffserv PHBs could classify those with an 'L' in the 'Coupled Queue' column (or local use DSCPs with similar characteristics) into its L queue, irrespective of the setting of the ECN field.

Service Class Name	DSCP Name	DSCP	AQM?	Coupled Queue
Network Control{1}	CS7	111000	Y & N	L if L4S
Network Control	CS6	110000	Y & N	L if L4S
OAM	CS2	010000	Y & N	L if L4S
Signalling	CS5	101000	N	L if L4S{2}
Telephony	EF	101110	N	L
RFC 5865	Voice-Admit	101100	N	L{3}
R-T Interactive	CS4	100000	N	L if L4S{4}
MM Conferencing	AF4n	100nn0	Y	L if L4S
Broadcast Video	CS3	011000	N	L if L4S{4}
MM Streaming	AF3n	011nn0	Y	L if L4S
Low Latency Data	AF2n	010nn0	Y	L if L4S
High Thru'put Data	AF1n	001nn0	Y	L if L4S{5}
Standard	BE/DF/CS0	000000	Y	L if L4S
Low Priority Data	LE{6}	000001	Y	L if L4S{7}

Some service class names have been abbreviated to fit. Abbreviations are expanded in [RFC 4594](#) or its updates. For the assured forwarding (AF) DSCP names, the digit 'n' represents 1, 2 or 3 and the corresponding binary digits 'nn' in the DSCP value represent 01,10 or 11. The 'Coupled Queue' column is explained in the text.

Table 1: Mapping of [RFC4594](#) Diffserv Service Classes in a Coupled AQM

Notes for Table 1:

- {1}: Reserved by [RFC 2474](#) [[RFC2474](#)].
- {2}: Superficially, CS5 is a candidate for classification into the L queue irrespective of its ECN field, given application signalling is bursty but usually lightweight. However, at least one major equipment vendor uses CS5 by default to indicate unresponsive broadcast video traffic (to which [RFC 4594](#) allocates CS3).
- {3}: Voice-Admit [[RFC5865](#)] could be given priority over Expedited Forwarding (EF) [[RFC3246](#)].

- {4}: The Real-Time Interactive and Broadcast Video service classes (or any equivalent local-use classes) are intended for inelastic traffic. Therefore they would not be expected to mark themselves as ECN-capable. If they did they would be claiming to be elastic and therefore eligible for classification into the L queue (subject to any policing). These classes should not be classified into the L queue on the basis of DSCP alone, because high bandwidth unresponsive traffic with potentially variable rate is not compatible with the L4S service.
- {5}: High Throughput Data (or any equivalent local-use class) might use the L4S service because of its support for scalable congestion control.
- {6}: [[I-D.ietf-tsvwg-le-phb](#)] updates [RFC 4594](#) to deprecate using CS1 for Lower Effort (LE).
- {7}: If a packet is marked LE and ECT(1) and the operator has solely provided a DualQ, this recommends that the packet is classified into the L queue. This could result in LE traffic competing for bandwidth with other classes of traffic in the L queue, but at least it should not harm the latency of other traffic. This is because the ECT(1) marking means the source "MUST" use a scalable congestion control [[I-D.ietf-tsvwg-ecn-l4s-id](#)], but the LE marking only means it "SHOULD" use an LBE congestion control [[I-D.ietf-tsvwg-le-phb](#)].

Those classes with an 'L' in the 'DualQ-Coupled' column would not be expected to have the ECT(1) codepoint set because they are generally unresponsive to congestion. Nonetheless, they could coexist in the same queue as L4S traffic because traffic in all of these classes is expected to arrive smoothly, not in bursts of more than a few packets. Therefore an operator could configure a DualQ Coupled AQM to classify such packets into the L queue solely based on their DSCP, irrespective of their ECN codepoint [[I-D.ietf-tsvwg-ecn-l4s-id](#)].

Otherwise, [[I-D.ietf-tsvwg-ecn-l4s-id](#)] requires that any other DSCP has no effect on classification into the L queue. Thus a packet of any other DSCP will not be classified into the L queue unless it carries an ECT(1) or CE codepoint in the ECN field. This is shown as 'L if L4S' in in the 'DualQ-Coupled' column of Table 1.

4. DualQ Bandwidth Pool within a Hierarchy of (Diffserv) Bandwidth Queues

The DualQ Coupled AQM offers an L queue that provides low latency low loss service but it pools bandwidth with the Classic (C) service as if they shared a single FIFO. As explained earlier, unlike previous Diffserv low latency mechanisms, the L queue can offer low latency without needing to limit its bandwidth.

Typically the DualQ will be able to use all the bandwidth available to a customer site, e.g. a household, a campus or a mobile node, as a single pool. However, this section considers scenarios where the network operator might want to carve off a fraction of a site's bandwidth for other purposes, for instance:

1. to ensure that a particularly demanding application (e.g. a virtual reality session) survives even if excess traffic overloads the remainder of the site's bandwidth;
2. to give guaranteed low latency to a particular application (e.g. industrial process control), if the statistically assured low latency of the L queue is insufficiently stable;
3. to provide a bandwidth scavenger service that will have no effect on any other applications at the site, but will scavenge any unused bandwidth, for instance to transfer backups or large data sets.

In all cases, it is assumed that the DualQ has to be able to borrow back any of the carved off bandwidth that is unused by the other service.

The following three subsections present solutions for each of the above scenarios. Depending on the reader's viewpoint, each scenario can be seen as:

- o either taking a queue within an existing Diffserv hierarchy and splitting it into L4S and Classic queues;
- o or building a queuing hierarchy around a pre-existing dual L4S/Classic queue.

In each case, the DualQ remains as an indivisible 'atomic' component as if it were a single queue with a single pool of bandwidth (but that can either be used for low latency or classic service).

The three examples represent the three main ways that this queue-like 'atom' can be included in a hierarchy of other queues. Without loss

of generality only one other queue complements the DualQ in each case, but it would be straightforward to extend the examples with more queues.

Although these examples are framed in the context of IP and Diffserv, similar queuing hierarchies could be constructed at a lower layer, as long as it supported a similar capability to ECN and a similar Traffic Class identifier to Diffserv.

4.1. DualQ Complemented by an Assured Bandwidth Service

Figure 1 shows a DualQ complemented by an additional queue to add a bandwidth assured service. It is assumed that the operator classifies certain packets into the assured bandwidth queue, perhaps by class of service, source address or 5-tuple flow ID.

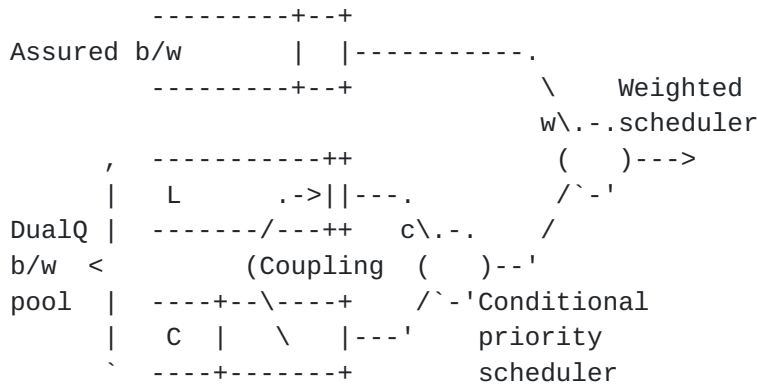


Figure 1: How to Complement a DualQ with an Assured Bandwidth Service

The DualQ is used as if it were an indivisible 'atomic' component, unchanged from its original description in [\[I-D.ietf-tsvwg-aqm-dualq-coupled\]](#):

- o The outputs of the AQMs in the two queues (L and C) are coupled together so that L4S sources leave enough space for C packets so that all 'standard' flows get roughly equal throughput;
- o A scheduler recombines the outputs of the two queues, giving conditional priority to L packets (the condition prevents starvation of the C queue if any L traffic misbehaves).

A weighted scheduler, e.g. weighted round robin (WRR), is used to combine the outputs of the assured bandwidth queue and the DualQ. It is configured with weight *w* for the assured bandwidth queue. Then, packets requesting assured bandwidth will have priority access to fraction *w* of the link capacity. However, whenever the assured

bandwidth queue is idle or under-utilized, the DualQ can borrow the balance of the bandwidth. Likewise the assured bandwidth queue can borrow more than fraction w if the DualQ under-utilizes its remaining share.

Note that a weighted scheduler such as WRR can be used to implement the conditional priority scheduler between the L and C queues. However, the system will not work as intended if the two weighted schedulers in series are replaced by a single three-input weighted scheduler. This is because, whenever one queue under-uses its weighted share, a weighted scheduler allows the other queue to borrow unused capacity. Whenever traffic is present in the C queue, the coupling ensures that L traffic makes space for it by underutilizing its share of the first scheduler. If the assured bandwidth queue was also served by the same scheduler, the assured bandwidth service would continually borrow the spare capacity left by the L queue that was intended for the C queue.

The assured bandwidth service could itself also support applications using low latency low loss and scalable throughput (L4S). This would be done by serving assured bandwidth traffic with a DualQ (Figure 2) and, as usual, confining legacy queue-building traffic to the C queue.

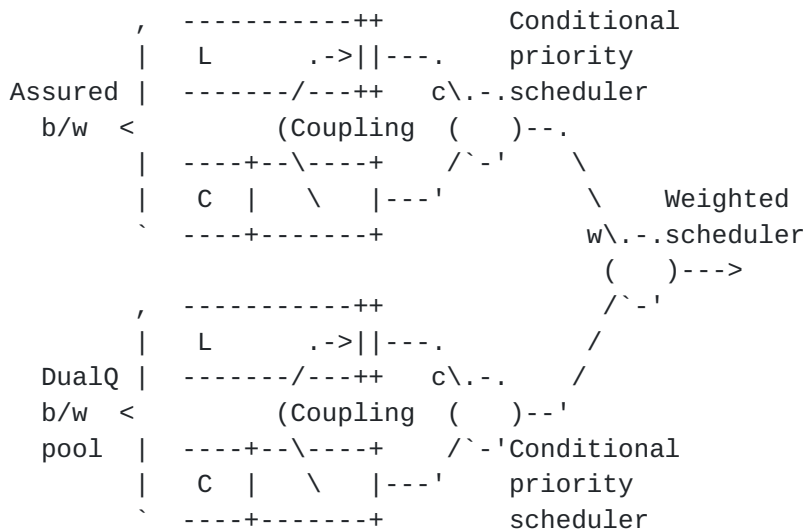


Figure 2: How to Complement a DualQ with an Assured Bandwidth Service that also Supports L4S

The symmetry of Figure 2 reveals that both DualQs actually have assured bandwidth. Nonetheless, the label 'Assured bandwidth' is only really meaningful from a per-application perspective if the

traffic classified into that DualQ is limited to a small number of application sessions at any one time.

4.2. DualQ Complemented by a Guaranteed Low Latency Service

Figure 3 shows a DualQ complemented by an additional queue to add a guaranteed latency service. It is assumed that the operator classifies certain packets into the guaranteed latency queue, perhaps by class of service, source address or 5-tuple flow ID.

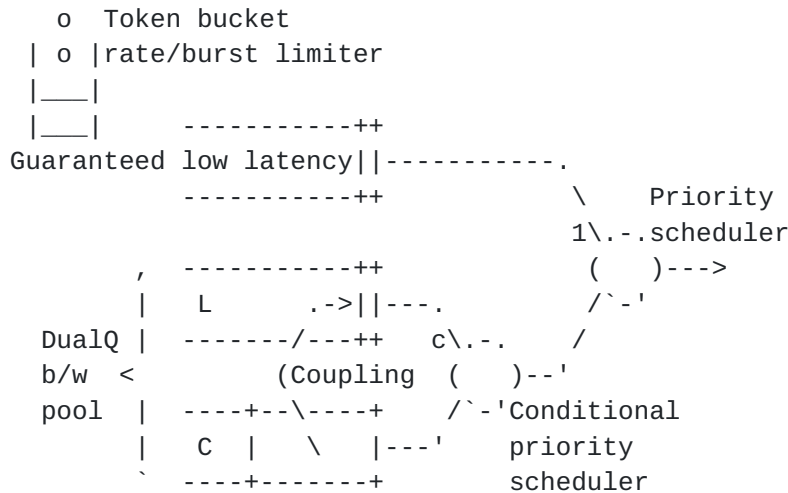


Figure 3: How to Complement a DualQ with a Guaranteed Low Latency Service

As in all the previous example, the DualQ is used as if it were an indivisible 'atomic' component.

A strict priority scheduler is used to combine the outputs of the guaranteed latency queue and the DualQ. Guaranteed low latency traffic is shown as subject to a token bucket that limits rate and tightly limits burst size, which ensures that:

- o Excessive guaranteed latency traffic cannot abuse its priority and cause the DualQ to starve;
- o Guaranteed latency traffic cannot ruin its own latency guarantees - it has to keep to a the traffic contract enforced by the token bucket.

In a traditional Diffserv architecture, the token bucket would be deployed at the ingress network edge, to limit traffic at each entry point. Alternatively, the token bucket could be deployed directly in front of the queue, where it would only limit the total traffic from

all entry points to the network. For an access link into a network, these two alternative would amount to the same thing.

Whenever the guaranteed latency queue is idle or under-utilized, the DualQ can borrow the balance of the bandwidth. However, the guaranteed latency queue cannot borrow more than the token bucket allows, even if the DualQ under-utilizes its remaining share.

4.3. DualQ Complemented by a Scavenger Service

Figure 3 shows a DualQ complemented by an additional queue to add a bandwidth scavenger service. It is assumed that the operator classifies certain packets into the scavenger queue, probably by class of service, e.g. the global-use Lower Effort (LE) Diffserv codepoint [[I-D.ietf-tsvwg-le-phb](#)].

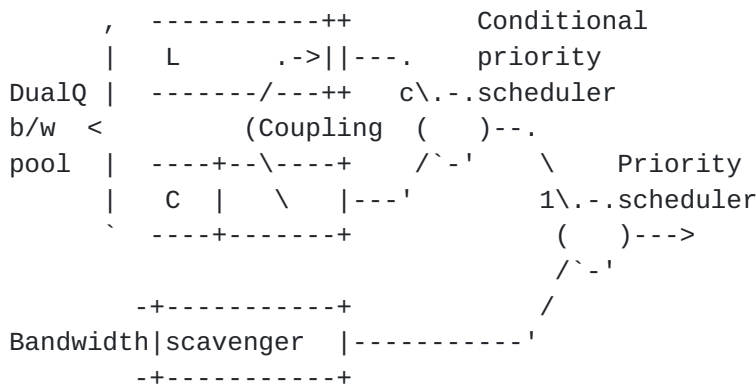


Figure 4: How to Complement a DualQ with a Bandwidth Scavenger Service

As in all the previous example, the DualQ is used as if it were an indivisible 'atomic' component.

A strict priority scheduler is used to combine the outputs of the DualQ and the scavenger service. Section 2 of [[I-D.ietf-tsvwg-le-phb](#)] suggests alternative mechanisms.

Whenever the DualQ is idle or under-utilized, the scavenger service can borrow the balance of the bandwidth. In contrast to the previous guaranteed latency example, no rate limiter is needed on the DualQ because, by definition, the scavenger service is expected to starve if the higher priority service is using all the capacity.

5. Coupling More than Two AQMs within a Bandwidth Pool

The Diffserv Assured Forwarding (AF) classes of service [RFC2597] use an AQM with differently weighted outputs, e.g. WRED, to provide weighted congestion feedback to the transport layer. Flows classified to use a higher weight AQM each take more of the available capacity, because the weighted AQM has fooled their congestion controller into detecting that the bottleneck is more lightly loaded.

A similar mechanism can be used to add throughput differentiation to either or both of the queues within a DualQ. Figure 5 illustrates an example with an AQM offering three weights within the L queue, where L1 gets the highest throughput per flow. It would be a matter of operator policy to choose which of the three L4S AQMs the Classic AQM would couple to. If it were coupled to L3, then C and L3 flows would get roughly equal throughput, while L2 and L1 flows would get more.

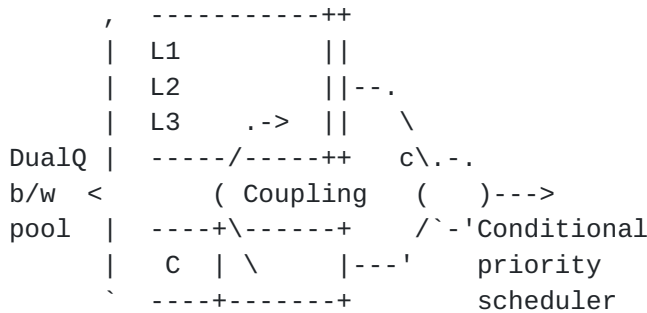


Figure 5: Coupling the Classic AQM to Multiple L4S AQMs

Note: this structure seems straightforward to implement, but the authors are not aware of any implementation or evaluation of AQMs that are both weighted and coupled to other AQMs.

6. Applicability of Coupled AQM to Global Diffserv PHBs

As has been explained, Diffserv always divides up bandwidth and divides up latency along the same lines as a consequence, whereas the DualQ Coupled AQM solely provides latency separation without bandwidth separation (the idea being that bandwidth separation can be added if needed, using Diffserv mechanisms).

In this draft so far, various queuing structures have been described in terms of the way they separate bandwidth and latency. Operators with existing Diffserv deployments may put the question the other way round and ask whether the DualQ Coupled AQM can be used to isolate low latency traffic within the bandwidth allocated to one of the standardized Diffserv PHBs. For instance:

- o Bandwidth has been allocated to Network Control traffic, but some BGP speakers have been upgraded to a low latency Scalable TCP while others still use Classic TCP. However it's not possible to predict how much bandwidth one or the other needs at any one time. So it would be useful to isolate the low latency BGP and all the control signalling from the delay caused by the legacy BGP speaker, without having to decide how to carve up the Network Control bandwidth.

- o Bandwidth has been allocated to Assured Forwarding (AF) traffic but it all shares the same WRED queue and therefore all suffers the same delay. So it would be useful to isolate the AF traffic that supports low latency congestion control from the rest. However, again, it is not possible to predict how many flows of each type there will be at any one time.

Table 2 lists all the PHBs with standardized global-use DSCPs from [\[RFC4594\]](#) and the right-hand 'Latency Separation?' column identifies all those that could benefit from an unknowable and variable fraction of their traffic being separated between ultra-low and regular delay using a DualQ Coupled AQM. There is no implication that it is sensible to do this in any of the cases; just that it is possible.

For convenience, the 'Mechanism' column also answers the question "How do PHBs for the global-use DSCPs map to the scenarios in this draft?"

Service Class Name	DSCP Name	AQM?	Mechanism	Latency Separation?
Network Control	CS7	Y&N	Figure 1 or Figure 2	Y
Network Control OAM	CS6	Y&N	Figure 1 or Figure 2	Y
Signalling	CS2	Y&N	Figure 1 or Figure 2	Y
Telephony	CS5	N	Figure 1	N
RFC 5865	EF	N	Section 4.2	N
R-T	VA	N	Section 4.2	N
Interactive	CS4	N	Figure 2	Y
MM Conferencing	AF4n	Y	Section 5	Y
Broadcast	CS3	N	Figure 2	N
Video				
MM Streaming	AF3n	Y	Section 5	Y
Low Latency Data	AF2n	Y	Section 5	Y
High Thru'put Data	AF1n	Y	Section 5	Y
Standard	BE/DF/CS0	Y	Section 3	Y
Low Priority Data	LE	Y	Section 4.3	n/a

Table 2: Applicability of a Coupled AQM to [RFC4594](#) Diffserv PHBs

7. Best Practice for Classification and Marking

7.1. Never Re-Mark a DSCP

It is not a DualQ's job to alter Diffserv codepoints to attempt to make other downstream AQMs classify selected packets in certain ways. Each DualQ Coupled AQM is independently (but hopefully consistently) configured to select certain DSCPs for classification into the L queue. It never alters the DSCP nor the ECN codepoint (except setting CE to indicate that congestion was experienced) [[I-D.ietf-tsvwg-aqm-dualq-coupled](#)].

7.2. Classification Order

7.2.1. Classification Order: Problem

The above wide range of possible structures raises the question of which order it would be more efficient for classifier rules to take: DSCP before ECN, ECN before DSCP or some hybrid.

On the one hand, for a structure like that in Figure 1 it would make sense to classify on DSCP first, then ECN. Otherwise, if packets were classified on ECN first, an extra merge stage would be required because the assured bandwidth queue handles all ECN codepoints for a particular DSCP.

On the other hand, for a structure like that in Figure 5 it would make sense to classify on ECN first, then DSCP. Otherwise, again an extra merge stage would be needed, because the C queue handles all DSCPs but only some ECN codepoints.

A hybrid of these two scenarios would be possible, for instance where the L queue in Figure 1 was further broken down into three weighted AQMs, as in Figure 5. In this case, the ideal matching order would be DSCP, ECN, DSCP.

7.2.2. Classification Order: Solutions

Probably the most straightforward solution would be to classify in a single stage over all 8 octets of the IPv6 Traffic Class field or the former IPv4 TOS octet, irrespective of the boundary between the 6-bit DS field and the 2-bit ECN field [[RFC3260](#)]. As long as hardware supports this, it will be possible because all the inputs to the queues are at the same level of hierarchy, even though the outputs form a multi-level hierarchy of schedulers in some cases.

Pre-existing classifier hardware might consider the 6-bit and 2-bit fields as separate. Then it would seem most efficient for the order of the classifiers to depend on the structure of the queues being classified (given the structure has to have been designed before the classifiers are designed).

8. Policing and Traffic Conditioning

{ToDo: L4S latency policing is discussed in the Security Considerations section of [[I-D.ietf-tsvwg-l4s-arch](#)]. This section will compare Diffserv traffic conditioning with L4S latency policing.}

9. IANA Considerations

This specification contains no IANA considerations.

10. Security Considerations

{ToDo}

11. Comments Solicited

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

12. Acknowledgements

Thanks to Greg White, David Black, Wes Eddy and Gorry Fairhurst for their useful discussions prior to this -00 draft.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

13.2. Informative References

[I-D.ietf-tsvwg-aqm-dualq-coupled]
Schepper, K., Briscoe, B., Bondarenko, O., and I. Tsang, "DualQ Coupled AQMs for Low Latency, Low Loss and Scalable Throughput (L4S)", [draft-ietf-tsvwg-aqm-dualq-coupled-07](#) (work in progress), October 2018.

[I-D.ietf-tsvwg-ecn-l4s-id]
Schepper, K. and B. Briscoe, "Identifying Modified Explicit Congestion Notification (ECN) Semantics for Ultra-Low Queuing Delay (L4S)", [draft-ietf-tsvwg-ecn-l4s-id-03](#) (work in progress), July 2018.

[I-D.ietf-tsvwg-l4s-arch]
Briscoe, B., Schepper, K., and M. Bagnulo, "Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture", [draft-ietf-tsvwg-l4s-arch-03](#) (work in progress), October 2018.

[I-D.ietf-tsvwg-le-phb]

Bless, R., "A Lower Effort Per-Hop Behavior (LE PHB)",
[draft-ietf-tsvwg-le-phb-06](#) (work in progress), October
2018.

- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black,
"Definition of the Differentiated Services Field (DS
Field) in the IPv4 and IPv6 Headers", [RFC 2474](#),
DOI 10.17487/RFC2474, December 1998,
<<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z.,
and W. Weiss, "An Architecture for Differentiated
Services", [RFC 2475](#), DOI 10.17487/RFC2475, December 1998,
<<https://www.rfc-editor.org/info/rfc2475>>.
- [RFC2597] Heinanen, J., Baker, F., Weiss, W., and J. Wroclawski,
"Assured Forwarding PHB Group", [RFC 2597](#),
DOI 10.17487/RFC2597, June 1999,
<<https://www.rfc-editor.org/info/rfc2597>>.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition
of Explicit Congestion Notification (ECN) to IP",
[RFC 3168](#), DOI 10.17487/RFC3168, September 2001,
<<https://www.rfc-editor.org/info/rfc3168>>.
- [RFC3246] Davie, B., Charny, A., Bennet, J., Benson, K., Le Boudec,
J., Courtney, W., Davari, S., Firoiu, V., and D.
Stiliadis, "An Expedited Forwarding PHB (Per-Hop
Behavior)", [RFC 3246](#), DOI 10.17487/RFC3246, March 2002,
<<https://www.rfc-editor.org/info/rfc3246>>.
- [RFC3260] Grossman, D., "New Terminology and Clarifications for
Diffserv", [RFC 3260](#), DOI 10.17487/RFC3260, April 2002,
<<https://www.rfc-editor.org/info/rfc3260>>.
- [RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration
Guidelines for DiffServ Service Classes", [RFC 4594](#),
DOI 10.17487/RFC4594, August 2006,
<<https://www.rfc-editor.org/info/rfc4594>>.
- [RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion
Control", [RFC 5681](#), DOI 10.17487/RFC5681, September 2009,
<<https://www.rfc-editor.org/info/rfc5681>>.

- [RFC5865] Baker, F., Polk, J., and M. Dolly, "A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic", [RFC 5865](#), DOI 10.17487/RFC5865, May 2010, <<https://www.rfc-editor.org/info/rfc5865>>.
- [RFC8257] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., and G. Judd, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", [RFC 8257](#), DOI 10.17487/RFC8257, October 2017, <<https://www.rfc-editor.org/info/rfc8257>>.
- [RFC8311] Black, D., "Relaxing Restrictions on Explicit Congestion Notification (ECN) Experimentation", [RFC 8311](#), DOI 10.17487/RFC8311, January 2018, <<https://www.rfc-editor.org/info/rfc8311>>.

Appendix A. Open Issues

- o The Abstract promises "in which cases one can stand alone without needing the other". That's in the text, but not highlighted as such.
- o Document Roadmap TBA
- o Mapping to 802.11 user priorities (or LTE QCIs)? Not strictly within the scope, but perhaps desirable to add, or at least to mention how L4S (experimental) would affect [RFC8325](#) which gives (standards track) mappings between Diffserv and 802.11.
- o Identify L4S-friendly rate policers
- o Comparison between L4S policing and Diffserv traffic conditioning is TBA
- o Security Considerations are TBA (largely depends on the previous bullet)

Author's Address

Bob Briscoe
CableLabs
UK

Email: ietf@bobbriscoe.net
URI: <http://bobbriscoe.net/>

