Transport Area Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 6, 2021

G. White
CableLabs
T. Fossati
ARM

November 2, 2020

A Non-Queue-Building Per-Hop Behavior (NQB PHB) for Differentiated Services
draft-ietf-tsvwg-nqb-03

Abstract

This document specifies properties and characteristics of a Non-Queue-Building Per-Hop Behavior (NQB PHB). The purpose of this NQB PHB is to provide a separate queue that enables low latency and, when possible, low loss for application-limited traffic flows that would ordinarily share a queue with capacity-seeking traffic. This PHB is implemented without prioritization and without rate policing, making it suitable for environments where the use of either these features may be restricted. The NQB PHB has been developed primarily for use by access network segments, where queuing delays and queuing loss caused by Queue-Building protocols are manifested, but its use is not limited to such segments. In particular, applications to cable broadband links and mobile network radio and core segments are discussed. This document defines a standard Differentiated Services Code Point (DSCP) to identify Non-Queue-Building flows.

Table of Contents

1. Introduction

This document defines a Differentiated Services (DS) per-hop behavior (PHB) called "Non-Queue-Building Per-Hop Behavior" (NQB PHB), which is intended to enable networks to provide low latency and low loss for traffic flows that are relatively low data rate and that do not themselves materially contribute to queueing delay and loss. Such Non-Queue-Building flows (for example: interactive voice and video, gaming, machine to machine applications) are application limited

flows that are distinguished from traffic flows managed by an end-to-end congestion control algorithm.

The vast majority of packets that are carried by broadband access networks are, in fact, managed by an end-to-end congestion control algorithm, such as Reno, Cubic or BBR. These congestion control algorithms attempt to seek the available capacity of the end-to-end path (which can frequently be the access network link capacity), and in doing so generally overshoot the available capacity, causing a queue to build-up at the bottleneck link. This queue build up results in queuing delay (variable latency) and possibly packet loss that affects all of the applications that are sharing the bottleneck link.

In contrast to traditional congestion-controlled applications, there are a variety of relatively low data rate applications that do not materially contribute to queueing delay and loss, but are nonetheless subjected to it by sharing the same bottleneck link in the access network. Many of these applications may be sensitive to latency or latency variation, as well as packet loss, and thus produce a poor quality of experience in such conditions.

Active Queue Management (AQM) mechanisms (such as PIE [RFC8033], DOCSIS-PIE [RFC8034], or CoDel [RFC8289]) can improve the quality of experience for latency sensitive applications, but there are practical limits to the amount of improvement that can be achieved without impacting the throughput of capacity-seeking applications, particularly when only a few of such flows are present.

The NQB PHB supports differentiating between these two classes of traffic in bottleneck links and queuing them separately in order that both classes can deliver satisfactory quality of experience for their applications.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Overview: Non-Queue-Building Flows

There are many applications that send traffic at relatively low data rates and/or in a fairly smooth and consistent manner such that they are highly unlikely to exceed the available capacity of the network path between source and sink. These applications do not cause queues

to form in network buffers, but nonetheless can be subjected to packet delay and delay variation as a result of sharing a network buffer with applications that do cause queues. Many of these applications are negatively affected by excessive packet delay and delay variation. Such applications are ideal candidates to be queued separately from the capacity-seeking applications that are the cause of queue buildup, latency and loss.

These Non-queue-building (NQB) flows are typically UDP flows that don't seek the capacity of the link (examples: online games, voice chat, DNS lookups, real-time IoT analytics data). Here the data rate is limited by the Application itself rather than by network capacity - in many cases these applications only send a few packets per RTT. In contrast, Queue-building (QB) flows include traffic which uses the Traditional TCP or QUIC, with BBR or other TCP congestion controllers.

4. The NQB PHB and its Relationship to the DiffServ Architecture

The IETF has defined the Differentiated Services architecture [RFC2475] with the intention that it allows traffic to be marked in manner that conveys the performance requirements of that traffic either quantitatively or in a relative sense (i.e. priority). The architecture defines the use of the DiffServ field [RFC2474] for this purpose, and numerous RFCs have been written that describe both standardized and recommended interpretations of the values (DiffServ Code Points) of the field, and of the treatments (traffic conditioning and per-hop-behaviors) that can be implemented to satisfy the performance requirements of traffic so marked.

While this architecture is powerful, and can be configured to meet the performance requirements of a variety of applications and traffic categories, or to achieve differentiated service offerings, it has proven problematic to enable its use for these purposes end-to-end across the Internet.

This difficulty is in part due to the fact that meeting (in an end-to-end context) the performance requirements of an application involves all of the networks in the path agreeing on what those requirements are, and sharing an interest in meeting them. In many cases this is made more difficult due to the fact that the performance "requirements" are not hard ones (e.g. applications will degrade in some manner as loss/latency/jitter increase), so the importance of meeting them for any particular application involves a judgment as to the value of avoiding some amount of degradation in quality for that application in exchange for an increase in the degradation of another application.

Further, in many cases the implementation of DiffServ PHBs involves prioritization of service classes with respect to one another, which results in the need to limit access to higher priority classes via mechanisms such as access control, admission control, traffic conditioning and rate policing, and/or to meter and bill for carriage of such traffic. These mechanisms can be difficult or impossible to implement in an end-to-end context.

Finally, some jurisdictions impose regulations that limit the ability of networks to provide differentiation of services, in large part based on the understanding that doing so ordinarily involves prioritization or privileged access to bandwidth, and thus a benefit to one class of traffic always comes at the expense of another.

In contrast, the NQB PHB has been designed with the goal that it avoids many of these issues, and thus could conceivably be deployed end-to-end across the Internet. The intent of the NQB DSCP is that it signals verifiable behavior as opposed to wants and needs. Also, the NQB traffic is to be given a separate queue with equal priority as default traffic, and given no reserved bandwidth other than the bandwidth that it shares with default traffic. As a result, the NQB PHB does not aim to meet specific application performance requirements, nor does it aim to provide a differentiated service class as defined in [RFC4594]. Instead the goal of the NQB PHB is to provide statistically better loss, latency, and jitter performance for traffic that is not itself the cause of those degradations. These attributes eliminate the inherent value judgments that underlie the handling of differentiated service classes in the DiffServ architecture as it has traditionally been defined, they also significantly simplify access control and admission control functions, reducing them to simple verification of behavior.

5. DSCP Marking of NQB Traffic

Applications that align with the description of NQB behavior in the preceding section SHOULD identify themselves to the network using a DiffServ Code Point (DSCP) so that their packets can be queued separately from QB flows.

There are many application flows that fall very neatly into one or the other of these categories, but there are also application flows that may be in a gray area in between (e.g. they are NQB on higher-speed links, but QB on lower-speed links).

If there is uncertainty as to whether an application's traffic aligns with the description of NQB behavior in the preceding section, the application SHOULD NOT mark its traffic with the NQB DSCP. In such a case, the application SHOULD instead implement a congestion control

mechanism, for example as described in [RFC8085] or [I-D.ietf-tsvwg-ecn-l4s-id].

This document recommends a DSCP of 42 (0x2A) to identify packets of NQB flows.

It is worthwhile to note again that the NQB designation and marking is intended to convey verifiable traffic behavior, not needs or wants. Also, it is important that incentives are aligned correctly, i.e. that there is a benefit to the application in marking its packets correctly, and no benefit to an application in intentionally mismarking its traffic. Thus, a useful property of nodes that support separate queues for NQB and QB flows would be that for NQB flows, the NQB queue provides better performance than the QB queue; and for QB flows, the QB queue provides better performance than the NQB queue. By adhering to these principles, there is no incentive for senders to mismark their traffic as NQB, and further, any mismarking can be identified by the network.

5.1. End-to-end usage and DSCP Re-marking

In contrast to the existing standard DSCPs, many of which are typically only meaningful within a DiffServ Domain (e.g. an AS or an enterprise network), this DSCP is expected to be used end-to-end across the Internet. Some network operators typically bleach (zero out) the DiffServ field on ingress into their network [Custura][Barik], and in some cases apply their own DSCP for internal usage. Bleaching the NQB DSCP is not expected to cause harm to default traffic, but it will severely limit the ability to provide NQB treatment end-to-end. Absent an explicit agreement to the contrary, networks that support the NQB PHB SHOULD preserve the NQB DSCP when forwarding via an interconnect from or to another network.

The fact that this DSCP is intended for end-to-end usage does not preclude networks from mapping the NQB DSCP to some other value for internal usage, as long as the NQB DSCP is restored when forwarding to another network. Additionally, it is not precluded for interconnecting networks to negotiate (via an SLA or some other agreement) a different DSCP to use to signal NQB across the interconnect.

Reports on existing deployments of DSCP manipulation [Custura][Barik] categorize the remarking behaviors into the following six policies: bleach all traffic (set DSCP to zero), set the top three bits (the former Precedence bits) on all traffic to 0b000, 0b001, or 0b010, set the low three bits on all traffic to 0b000, or remark all traffic to a particular (non-zero) DSCP value. There were no observations reported in which traffic was marked 42 by any of these policies.

Thus it appears that these remarking policies would be unlikely to result in QB traffic being marked as NQB. In terms of the fate of NQB-marked traffic that is subjected to one of these policies, the result would be that NQB marked traffic would be indistinguishable from some subset (possibly all) of other traffic. In the policies where all traffic is remarked using the same (zero or non-zero) DSCP, the ability for a subsequent network hop to differentiate NQB traffic via DSCP would clearly be lost entirely. In the policies where the top three bits are overwritten, NQB would receive the same marking as AF41, AF31, AF21, AF11 (as well as the currently unassigned DSCPs 2, 50, 58), with all of these code points getting mapped to DSCP=2, AF11 or AF21 (depending on the overwrite value used). Since the recommended usage of the standardized code points in that list include high throughput data for store and forward applications (and it is impossible to predict what future use would be assigned to the currently unassigned values) it would seem inadvisable for a node to attempt to treat all such traffic as if it were NQB marked. For the policy in which the low three bits are set to 0b000, the NQB value would be mapped to CS5 and would be indistinguishable from CS5, VA, EF (and the unassigned DSCPs 41, 43, 45). Traffic marked using the existing standardized DSCPs in this list are likely to share the same general properties as NQB traffic (non capacity-seeking, very low data rate or relatively low and consistent data rate). Furthermore, as this remarking policy results in an overt enforcement of the IP Precedence compatibility configuration discussed in [RFC4594] Section 1.5.4, and to the extent that this compatibility is maintained in the future, any future recommended usages of the currently unassigned DSCPs in that list would be likely to similarly be somewhat compatible with NQB treatment. Here there may be an opportunity for a node to provide the NQB PHB or the CS5 PHB and retain some of the benefits of NQB marking. As a result, nodes supporting the NQB PHB MAY additionally classify CS5 marked traffic into the NQB queue.

Note: Unless agreed otherwise between the interconnecting partners, interconnects that implement [RFC8100] for DiffServ interconnection would consider the NQB DSCP as an unrecognized or unsupported DSCP, and would thus re-mark it to CS0.

5.2. Aggregation of the NQB PHB with other DiffServ PHBs

Networks and nodes that aggregate service classes as discussed in [RFC5127] may not be able to provide a PDB/PHB that meets the requirements of this document. In these cases it is recommended that NQB-marked traffic be aggregated with standard, elastic, best-effort traffic, although in some cases a network operator may instead choose to aggregate NQB traffic with Real-Time traffic. Either approach comes with trade-offs: aggregating with best-effort could result in a

degradation of loss/latency/jitter performance, while aggregating with Real-Time may create an incentive for mismarking of non-compliant traffic. In either case, [RFC5127] requires that such aggregations preserve the notion of each end-to-end service class that is aggregated, and recommends preservation of the DSCP as a way of accomplishing this. Compliance with this recommendation would serve to limit the negative impact that such networks would have on end-to-end performance for NQB traffic.

Nodes that support the NQB PHB may choose to aggregate other service classes into the NQB queue. Candidate service classes for this aggregation would include those that carry inelastic traffic that has low to very-low tolerance for loss, latency and/or jitter as discussed in [RFC4594]. These could include Network Control, Telephony, Signaling, Real-Time Interactive and Broadcast Video.

6. Non-Queue-Building PHB Requirements

A node supporting the NQB PHB makes no guarantees on latency or data rate for NQB marked flows, but instead aims to provide a bound on queuing delay for as many such marked flows as it can, and shed load when needed.

A node supporting the NQB PHB MUST provide a queue for non-queue-building traffic separate from the queue used for queue-building traffic.

NQB traffic, in aggregate, SHOULD NOT be rate limited or rate policed separately from queue-building traffic of equivalent importance.

The NQB queue SHOULD be given equal priority compared to queue-building traffic of equivalent importance. The node SHOULD provide a scheduler that allows QB and NQB traffic of equivalent importance to share the link in a fair manner, e.g. a deficit round-robin scheduler with equal weights.

A node supporting the NQB PHB SHOULD treat traffic marked as Default (DSCP=0) as QB traffic having equivalent importance to the NQB marked traffic. A node supporting the NQB DSCP MUST support the ability to configure the classification criteria that are used to identify QB and NQB traffic having equivalent importance.

The NQB queue SHOULD have a buffer size that is significantly smaller than the buffer provided for QB traffic. It is expected that most QB traffic is optimized to make use of a relatively deep buffer (e.g. on the order of tens or hundreds of ms) in nodes where support for the NQB PHB is advantageous (i.e. bottleneck nodes). Providing a similarly deep buffer for the NQB queue would be at cross purposes to

providing very low queueing delay, and would erode the incentives for QB traffic to be marked correctly.

It is possible that due to an implementation error or misconfiguration, a QB flow would end up getting mismarked as NQB, or vice versa. In the case of an NQB flow that isn't marked as NQB and ends up in the QB queue, it would only impact its own quality of service, and so it seems to be of lesser concern. However, a QB flow that is mismarked as NQB would cause queuing delays and/or loss for all of the other flows that are sharing the NQB queue.

To prevent this situation from harming the performance of the real NQB flows, network elements that support differentiating NQB traffic SHOULD support a "traffic protection" function that can identify QB flows that are mismarked as NQB, and reclassify those flows/packets to the QB queue. Such a function SHOULD be implemented in an objective and verifiable manner, basing its decisions upon the behavior of the flow rather than on application-layer constructs. One example algorithm can be found in [I-D.briscoe-docsis-q-protection]. There are some situations where such function may not be necessary. For example, a network element designed for use in controlled environments, e.g. enterprise LAN may not require a traffic protection function. Similarly, flow queueing systems obviate the need for an explicit traffic protection function. Additionally, some networks may prefer to police the application of the NQB DSCP at the ingress edge, so that in-network traffic protection is not needed.

7. Impact on Higher Layer Protocols

Network elements that support the NQB PHB and that support traffic protection as discussed in the previous section introduce the possibility that flows classified into the NQB queue could experience out of order delivery. This is particularly true if the traffic protection algorithm makes decisions on a packet-by-packet basis. In this scenario, a flow that is (mis)marked as NQB and that causes a queue to form in this bottleneck link could see some of its packets forwarded by the NQB queue, and some of them redirected to the QB queue. Depending on the queueing latency and scheduling within the network element, this could result in packets being delivered out of order. As a result, the use of the NQB DSCP by a higher layer protocol carries some risk that out of order delivery will be experienced.

8. The NQB PHB and Tunnels

[RFC2983] discusses tunnel models that support DiffServ. It describes a "uniform model" in which the inner DSCP is copied to the outer header at encapsulation, and the outer DSCP is copied to the inner header at decapsulation. It also describes a "pipe model" in which the outer DSCP is not copied to the inner header at decapsulation. Both models can be used in conjunction with the NQB PHB. In the case of the pipe model, any DSCP manipulation (re-marking) of the outer header by intermediate nodes would be discarded at tunnel egress, potentially improving the possibility of achieving NQB treatment in subsequent nodes.

As is discussed in [RFC2983] tunnel protocols that are sensitive to reordering can result in undesirable interactions if multiple DSCP PHBs are signaled for traffic within a tunnel instance. This is true for NQB marked traffic as well. If a tunnel contains a mix of QB and NQB traffic, and this is reflected in the outer DSCP in a network that supports the NQB PHB, it would be necessary to avoid a reordering-sensitive tunnel protocol in order to avoid these undesirable interactions.

9. Relationship to L4S

Traffic flows marked with the NQB DSCP as described in this draft are intended to be compatible with [I-D.ietf-tsvwg-l4s-arch], with the result being that NQB traffic and L4S traffic can share the low-latency queue in an L4S dual-queue node [I-D.ietf-tsvwg-aqm-dualq-coupled]. Compliance with the DualQ coupled AQM requirements is considered sufficient to enable fair allocation of bandwidth between the QB and NQB queues.

10. Configuration and Management

As required above, nodes supporting the NQB PHB provide for the configuration of classifiers that can be used to differentiate between QB and NQB traffic of equivalent importance. The default for such classifiers is recommended to be the assigned NQB DSCP (to identify NQB traffic) and the Default (0) DSCP (to identify QB traffic).

11. Example Use Cases

11.1. DOCSIS Access Networks

Residential cable broadband Internet services are commonly configured with a single bottleneck link (the access network link) upon which the service definition is applied. The service definition, typically

an upstream/downstream data rate tuple, is implemented as a configured pair of rate shapers that are applied to the user's traffic. In such networks, the quality of service that each application receives, and as a result, the quality of experience that it generates for the user is influenced by the characteristics of the access network link.

To support the NQB PHB, cable broadband services MUST be configured to provide a separate queue for NQB marked traffic. The NQB queue MUST be configured to share the service's rate shaping bandwidth with the queue for QB traffic.

11.2. Mobile Networks

Historically, mobile networks have been configured to bundle all flows to and from the Internet into a single "default" EPS bearer whose buffering characteristics are not compatible with low-latency traffic. The established behaviour is rooted partly in the desire to prioritise operators' voice services over competing over-the-top services and partly in the fact that the addition of bearers was prohibitive due to expense. Of late, said consideration seems to have lost momentum (e.g., with the rise in Multi-RAB (Radio Access Bearer) devices) and the incentives might now be aligned towards allowing a more suitable treatment of Internet real-time flows.

To support the NQB PHB, the mobile network SHOULD be configured to give UEs a dedicated, low-latency, non-GBR, EPS bearer, e.g. one with QCI 7, in addition to the default EPS bearer; or a Data Radio Bearer with 5QI 7 in a 5G system (see Table 5.7.4-1: Standardized 5QI to QoS characteristics mapping in [SA-5G]).

A packet carrying the NQB DSCP SHOULD be routed through the dedicated low-latency EPS bearer. A packet that has no associated NQB marking SHOULD be routed through the default EPS bearer.

11.3. WiFi Networks

WiFi networking equipment compliant with 802.11e generally supports either four or eight transmit queues and four sets of associated Enhanced Multimedia Distributed Control Access (EDCA) parameters (corresponding to the four WiFi Multimedia (WMM) Access Categories) that are used to enable differentiated media access characteristics.

While some WiFi equipment may be capable (in some cases via firmware update) of supporting the NQB PHB requirements by providing a separate queue for NQB marked traffic that shares an Access Category with default traffic, many currently deployed devices cannot be configured in this way.

Implementations typically utilize the IP DSCP field to select a transmit queue, but should be considered as Non-Differentiated Services-Compliant Nodes as described in Section 4 of [RFC2475] because in widely deployed WiFi networks, this transmit queue selection is a local implementation characteristic that is not part of a consistently operated DiffServ domain or region. As a result this document discusses interoperability with these existing WiFi networks, in addition to PHB compliance.

As discussed in [RFC8325], most existing WiFi implementations use a default DSCP to User Priority mapping that utilizes the most significant three bits of the DiffServ Field to select "User Priority" which is then mapped to the four WMM Access Categories. In order to increase the likelihood that NQB traffic is provided a separate queue from QB traffic in existing WiFi equipment, the 42 code point is preferred for NQB. This would map NQB to UP_5 which is in the "Video" Access Category. Similarly, systems that utilize [RFC8325], SHOULD map the NQB code point to UP_5 in the "Video" Access Category.

While the DSCP to User Priority mapping can enable WiFi systems to support the NQB PHB requirement for segregated queuing, many currently deployed WiFi systems may not be capable of supporting the remaining NQB PHB requirements in Section 6. This is discussed further below.

Existing WiFi devices are unlikely to support a traffic protection algorithm, so traffic mismarked as NQB is not likely to be detected and remedied by such devices.

Furthermore, in their default configuration, existing WiFi devices utilize EDCA parameters that result in statistical prioritization of the "Video" Access Category above the "Best Effort" Access Category. If left unchanged, this would violate the NQB PHB requirement for equal prioritization, and could erode the principle of alignment of incentives. In order to preserve the incentives principle, WiFi systems SHOULD configure the EDCA parameters for the Video Access Category to match those of the Best Effort Access Category.

In cases where a network operator is delivering traffic into an unmanaged WiFi network outside of their control (e.g. a residential ISP delivering traffic to a customer's home network), the network operator should presume that the existing WiFi equipment does not support the safeguards that are provided by the NQB PHB requirements, and thus should take precautions to prevent issues. In these situations, the operator SHOULD deploy a policing function on NQB marked traffic that minimizes the potential for starvation of traffic

marked Default, for example by limiting the rate of such traffic to a set fraction of the customer's service rate.

As an additional safeguard, and to prevent the inadvertent introduction of problematic traffic into unmanaged WiFi networks, network equipment that is intended to deliver traffic into unmanaged WiFi networks (e.g. an access network gateway for a residential ISP) MUST by default remap the NQB DSCP to Default. Such equipment MUST support the ability to configure the remapping, so that (when appropriate safeguards are in place) traffic can be delivered as NQB-marked.

12. Acknowledgements

Thanks to Bob Briscoe, Greg Skinner, Toke Hoeiland-Joergensen, Luca Muscariello, David Black, Sebastian Moeller, Ruediger Geib, Jerome Henry, Steven Blake, Jonathan Morton, Roland Bless, Kevin Smith, Martin Dolly, and Kyle Rose for their review comments.

13. IANA Considerations

This document assigns the Differentiated Services Field Codepoint (DSCP) 42 ('0b101010', 0x2A) from the "Differentiated Services Field Codepoints (DSCP)" registry (https://www.iana.org/assignments/dscp-registry/) ("DSCP Pool 1 Codepoints", Codepoint Space xxxxx0, Standards Action) to denote Non-Queue-Building behavior.

14. Security Considerations

There is no incentive for an application to mismark its packets as NQB (or vice versa). If a queue-building flow were to mark its packets as NQB, it could experience excessive packet loss (in the case that traffic protection is not supported by a node) or it could receive no benefit (in the case that traffic protection is supported). If a non-queue-building flow were to fail to mark its packets as NQB, it could suffer the latency and loss typical of sharing a queue with capacity seeking traffic.

In order to preserve low latency performance for NQB traffic, networks that support the NQB PHB will need to ensure that mechanisms are in place to prevent malicious NQB-marked traffic from causing excessive queue delays. This document recommends the implementation of a traffic protection mechanism to achieve this goal, but recognizes that other options may be more desirable in certain situations.

The NQB signal is not integrity protected and could be flipped by an on-path attacker. This might negatively affect the QoS of the tampered flow.

15. Informative References

[Barik] Barik, R., Welzl, M., Elmokashfi, A., Dreibholz, T., and S. Gjessing, "Can WebRTC QoS Work? A DSCP Measurement Study", ITC 30, September 2018.

[Custura] Custura, A., Venne, A., and G. Fairhurst, "Exploring DSCP modification pathologies in mobile edge networks", TMA , 2017.

[I-D.briscoe-docsis-q-protection] Briscoe, B. and G. White, "Queue Protection to Preserve Low Latency", draft-briscoe-docsis-q-protection-00 (work in progress), July 2019.

[I-D.ietf-tsvwg-aqm-dualq-coupled] Schepper, K., Briscoe, B., and G. White, "DualQ Coupled AQMs for Low Latency, Low Loss and Scalable Throughput (L4S)", draft-ietf-tsvwg-aqm-dualq-coupled-12 (work in progress), July 2020.

[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-10 (work in progress), March 2020.

[I-D.ietf-tsvwg-l4s-arch] Briscoe, B., Schepper, K., Bagnulo, M., and G. White, "Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture", draft-ietf-tsvwg-l4s-arch-07 (work in progress), October 2020.

[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>.

[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>.

[RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, DOI 10.17487/RFC2983, October 2000, <https://www.rfc-editor.org/info/rfc2983>.

[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>.

[RFC5127] Chan, K., Babiarz, J., and F. Baker, "Aggregation of Diffserv Service Classes", RFC 5127, DOI 10.17487/RFC5127, February 2008, <https://www.rfc-editor.org/info/rfc5127>.

[RFC8033] Pan, R., Natarajan, P., Baker, F., and G. White, "Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem", RFC 8033, DOI 10.17487/RFC8033, February 2017, <https://www.rfc-editor.org/info/rfc8033>.

[RFC8034] White, G. and R. Pan, "Active Queue Management (AQM) Based on Proportional Integral Controller Enhanced PIE) for Data-Over-Cable Service Interface Specifications (DOCSIS) Cable Modems", RFC 8034, DOI 10.17487/RFC8034, February 2017, <https://www.rfc-editor.org/info/rfc8034>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8100] Geib, R., Ed. and D. Black, "Diffserv-Interconnection Classes and Practice", RFC 8100, DOI 10.17487/RFC8100, March 2017, <https://www.rfc-editor.org/info/rfc8100>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8289] Nichols, K., Jacobson, V., McGregor, A., Ed., and J. Iyengar, Ed., "Controlled Delay Active Queue Management", RFC 8289, DOI 10.17487/RFC8289, January 2018, <https://www.rfc-editor.org/info/rfc8289>.

[RFC8325] Szigeti, T., Henry, J., and F. Baker, "Mapping Diffserv to IEEE 802.11", RFC 8325, DOI 10.17487/RFC8325, February 2018, <https://www.rfc-editor.org/info/rfc8325>.

[SA-5G] 3GPP, "System Architecture for 5G", TS 23.501, 2019.

Authors' Addresses

Greg White
CableLabs

Email: g.white@cablelabs.com

Thomas Fossati
ARM

Email: Thomas.Fossati@arm.com