Transport Area Working Group
Internet-Draft
Intended status: Informational
Expires: September 12, 2019

G. White
K. Sundaresan
B. Briscoe
CableLabs

March 11, 2019

Low Latency DOCSIS - Technology Overview
draft-white-tsvwg-lld-00

Abstract

NOTE: This document is a reformatted version of [LLD-white-paper].

The evolution of the bandwidth capabilities - from kilobits per second to gigabits - across generations of DOCSIS cable broadband technology has paved the way for the applications that today form our digital lives. Along with increased bandwidth, or "speed", the latency performance of DOCSIS technology has also improved in recent years. Although it often gets less attention, latency performance contributes as much or more to the broadband experience and the feasibility of future applications as does speed.

Low Latency DOCSIS technology (LLD) is a specification developed by CableLabs in collaboration with DOCSIS vendors and cable operators that tackles the two main causes of latency in the network: queuing delay and media acquisition delay. LLD introduces an approach wherein data traffic from applications that aren't causing latency can take a different logical path through the DOCSIS network without getting hung up behind data from applications that are causing latency, as is the case in today's Internet architectures. This mechanism doesn't interfere with the way applications share the total bandwidth of the connection, and it doesn't reduce one application's latency at the expense of others. In addition, LLD improves the DOCSIS upstream media acquisition delay with a faster request-grant loop and a new proactive scheduling mechanism. LLD makes the internet experience better for latency sensitive applications without any negative impact on other applications.

The latest generation of DOCSIS equipment that has been deployed in the field - DOCSIS 3.1 - experiences typical latency performance of around 10 milliseconds (ms) on the Access Network link. However, under heavy load, the link can experience delay spikes of 100 ms or more. LLD systems can deliver a consistent 1 ms delay on the DOCSIS network for traffic that isn't causing latency, imperceptible for nearly all applications. The experience will be more consistent with much smaller delay variation.

LLD can be deployed by field-upgrading DOCSIS 3.1 cable modem and cable modem termination system devices with new software. The technology includes tools that enable automatic provisioning of these new services, and it also introduces new tools to report statistics of latency performance to the operator.

Cable operators, DOCSIS equipment manufacturers, and application providers will all have to act in order to take advantage of LLD. This white paper explains the technology and describes the role that each of these parties plays in making LLD a reality.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Let's begin with bandwidth (or "speed"): the amount of data that can be delivered across a network connection over a period of time. Sometimes bandwidth is very important to the broadband experience, particularly when an application is trying to send or receive large amounts of data, such as watching videos on Netflix, downloading videos/music, syncing file-shares or email clients, uploading a video to YouTube or Instagram, or downloading a new application or system update. Other times, bandwidth (or bandwidth alone) isn't enough, and latency has a big effect on the user experience.

Latency is the time that it takes for a short message (a packet, in networking terminology) to make it across the network from the sender to the receiver and for a response to come back. Network latency is commonly measured as round-trip-time and is sometimes referred to as "ping time." Applications that are more interactive or real-time,

like web browsing, online gaming, and video conferencing/chatting, perform the best when latency is kept low, and adding more bandwidth without addressing latency doesn't make things better.

When multiple applications share the broadband connection of one household (e.g., several users doing different activities at the same time), each of those applications can have an impact on the performance of the others. They all share the total bandwidth of the connection (so more active applications mean less bandwidth for each one), and they can all cause the latency of the connection to increase.

It turns out that applications today that want to send a lot of data all at once do a reasonably good job of sharing the bandwidth in a fair manner, but they actually cause a pretty big latency problem when they do it because they send data too quickly and expect the network to queue it up. We call these applications "queue-building" applications, e.g., video streaming (Netflix). There are also plenty of other applications that don't send data too quickly, so they don't cause latency. We call these "non-queue-building" applications, e.g., video chatting (FaceTime).

LLD separates these two types of traffic into two logical queues, which greatly improves the latency experienced by the non-queue-building applications (many of which may be latency-sensitive) without having any downside for the queue-building applications. In addition, two queues allow LLD to support a next-generation application protocol that can scale up to sending data at 10 Gbps and beyond while maintaining ultra-low queuing delay, which means that in the future, there may not be queue-building applications at all.

As of the writing of this document, the Low Latency DOCSIS specifications have just been published ([DOCSIS-MULPIv3.1], [DOCSIS-CCAP-OSSIv3.1], [DOCSIS-CM-OSSIv3.1]), and DOCSIS equipment manufacturers are working on building support for the functionality. In addition, work is underway in the Internet Engineering Task Force to standardize low-latency architectures across the broader Internet ecosystem.

2. Latency in DOCSIS Networks

Low Latency DOCSIS technology is the next step in a progression of latency improvements that have been made to the DOCSIS specifications by CableLabs in recent years. Table 1 provides a snapshot of the milestones in round-trip latency performance with DOCSIS technology from the first DOCSIS 3.0 equipment to DOCSIS 3.1 equipment that supports [RFC8034] Active Queue Management, and finally the new Low Latency DOCSIS, which achieves ~1 ms of round-trip latency. The

table references three metrics that describe the range of latencies added by the DOCSIS network link that would be experienced by a broadband user. The first, "When Idle," refers to a broadband connection that is not being actively used by the customer. The second, "Under Load," represents average latency while the user is actively using the service (e.g., streaming video). Finally, the third, "99th Percentile," gives an indication of the maximum latency that a customer would commonly experience in real usage scenarios. The table uses order-of-magnitude numbers because the actual performance will vary because of a number of factors including DOCSIS channel configuration and actual application usage pattern.

For latency-sensitive applications, the 99th percentile value has the most impact on user experience.

TABLE 1. EVOLUTION OF LATENCY PERFORMANCE IN DOCSIS NETWORKS (ROUND-TRIP TIME IN MILLISECONDS BETWEEN THE CM AND CMTS)

	When Idle	Under Load	99th Percentile
DOCSIS 3.0 Early Equipment	~10 ms	~1000 ms	~1000 ms
DOCSIS 3.0 w/ Buffer Control	~10 ms	~100 ms	~100 ms
DOCSIS 3.1 Active Queue Management	~10 ms	~10 ms	~100 ms
Low Latency DOCSIS 3.1	~1 ms	~1 ms	~1 ms

Table 1

The latency described in Table 1 is caused by a series of factors in the DOCSIS cable modem (CM) and cable modem termination system (CMTS). Figure 1 in [LLD-white-paper] illustrates the range of latencies caused by those factors in DOCSIS 3.1 networks.

The lowest two latency sources in Figure 1 in [LLD-white-paper] have minor impacts on overall latency.

The "Switching/Forwarding" delay represents the amount of time it takes for the CM and CMTS to make the decision to forward a packet. This has a very minor impact on overall latency.

The "Propagation" delay (the amount of time it takes for a signal to travel on the HFC plant) is set by the speed of light and the distance from CM to CMTS. Not much can be done to affect latency from this source.

Of the sources in Figure 1 in [LLD-white-paper], the top three significantly drive latency performance.

The range of the "Serialization/Encoding" delay comes from the upstream and downstream channel configuration options available to the operator. Some of these configurations provide significant robustness benefits at the expense of latency, whereas others may be less robust to noise but provide very low latency. The LLD specification does not modify the set of options available to the operator. Rather, operators should be encouraged to use the lowest latency channel configurations that they can, given the plant conditions.

The "Media Acquisition" delay is a result of the shared-medium scheduling currently provided by DOCSIS technology, in which the CMTS arbitrates access to the upstream channel via a request-grant mechanism.

The "Queuing" delay is mainly caused by the current TCP protocol and its variants. Applications today that need to seek out as much bandwidth as possible use a transport protocol like TCP (or the TCP-replacement known as QUIC), which uses a "congestion control" algorithm (such as Reno, Cubic, or BBR) to adjust to the available bandwidth at the bottleneck link through the network. Typically, this will be the last mile link - the DOCSIS link for cable customers - where the bandwidth available for each application often varies rapidly as the activity of all the devices in the household varies.

With today's congestion control algorithms, the sender ramps up the sending rate until it's sending data faster than the bottleneck link can support. Packets then start queuing in a buffer at the entrance to the link, i.e. the CM or CMTS. This queue of packets grows quickly until the device decides to discard some newly arriving packets, which triggers the sender to pause for a bit in order to allow the buffer to drain somewhat before resuming sending. This process is an inherent feature of the TCP family of Internet transport protocols, and it repeats over and over again until the file transfer completes. In doing so, it causes latency and packet loss for all of the traffic that shares the broadband link.

LLD tackles the two main causes of latency in the network: queuing delay and media acquisition delay.

- LLD addresses Queueing Delay by allowing non-queue-building applications to avoid waiting behind the delays caused by the current TCP or its variants. At a high level, the low-latency architecture consists of a dual-queue approach that treats both queues as a single pool of bandwidth.

- o LLD cuts Media Acquisition Delay by using a faster request-grant loop and by adding support for a new proactive scheduler that can provide extremely low latency service.

In addition, LLD introduces detailed statistics on queueing delay via histogram calculations performed by the CM (for upstream) and CMTS (for downstream). Furthermore, CableLabs is working with a broad cross-section of stakeholders in the IETF to standardize an end-to-end service architecture that can leverage LLD to enable even high bandwidth TCP flows to achieve ultra-low queuing delay. This technology will be important for future, interactive high-data-rate applications like holographic light field experiences, as well as for enabling higher performance versions of today's applications like web and video conferencing.

The sections below describe these features in more detail.

3. New Dual-Queue Approach

Of all the features of LLD, the dual-queue mechanism has by far the greatest impact on round-trip latency and latency variation. The concept of the dual-queue approach is that the majority of the applications that use the internet can be divided into two categories:

- o Queue-Building Applications: These application traffic flows frequently send data faster than the path between sender and receiver can support. The most common instance of queue-building flows are flows that use the current TCP or QUIC protocols. As discussed above, these capacity-seeking protocols use a legacy congestion control algorithm that probes for available capacity on the path by sending data faster than the path can support and expecting the network to queue the excess data in internal buffers. The majority of traffic (by volume) today is queue-building. Some examples of queue-building applications are video streaming (e.g., Netflix, YouTube) and application downloads.

- o Non-Queue-Building Applications: These application traffic flows very rarely send data faster than the path can support. They come in two subcategories:

  - * Today's self-limited, non-capacity-seeking apps, such as multiplayer online games and IP communication apps (such as Skype or FaceTime). These applications send data at a relatively low data rate and generally space their packets out in a manner that does not cause a queue to form in the network.

* Future capacity-seeking TCP/QUIC applications that adopt the new L4S congestion control algorithm (see Appendix A) and so can immediately respond to fast congestion signals sent by the network. These applications are still in development, as networks must first support L4S before applications are able to take advantage, but some prime candidates are web browsing, cloud VR, and interactive light field experiences.

Queue-building (QB) application flows are the source of queuing delay, and today's non-queue-building (NQB) apps typically suffer from the latency caused by the QB flows.

The purpose of the dual-queue mechanism is to segment queue-building traffic from non-queue-building traffic in a manner that can be readily implemented in DOCSIS 3.1 equipment and that doesn't alter the overall bandwidth of the broadband service.

By segmenting these two types of applications into separate queues, each can get optimal performance. The QB traffic can build a queue and achieve the necessary and expected throughput performance, and the NQB traffic can take advantage of the available lower latencies by avoiding the delay caused by the QB flows. It is important to note that this segmentation of traffic isn't for purposes of giving one class of traffic benefits at the expense of the other - it isn't a high-priority queue and a low-priority queue. Instead, each queue is optimized for the distinct features and requirements of the two classes of traffic, enabling increased functionality and adding value for the broadband user. This is smart network management at work.

3.1. Low-Latency Aggregate Service Flows

DOCSIS 3.1 equipment, like equipment built against earlier versions of the specification, supports a number of upstream and downstream Service Flows (SFs). These Service Flows are logical pipes that are defined by their configured Quality of Service (QoS) parameters (most commonly, the rate shaping parameters [MULPIv3.1] that specify the speed of user connections) and that carry a subset of the traffic to/from a particular CM, as specified by a set of packet classifiers configured by the operator. Traditionally, each Service Flow provides near-complete isolation of its traffic from the traffic transiting other Service Flows (those on the same CM as well as those on other CMs) - each Service Flow has its own buffer and queue and is scheduled independently by the CMTS.

Typically, the operator defines a service offering via the configuration of a single upstream Service Flow and a single downstream Service Flow with rate shaping enabled, and all of the user's traffic transits these two Service Flows.

The DOCSIS 3.1 specification already includes optional support in the CMTS for a mechanism to group any number of the Service Flows serving a particular CM. LLD leverages and extends this "Aggregate Service Flow" (ASF) feature to establish (and group) a pair of Service Flows in each direction specifically to enable low-latency services. One of the Service Flows in the pair (the "Low Latency Service Flow") will carry NQB traffic, and the other Service Flow (the "Classic Service Flow") will carry QB traffic. The Aggregate Service Flow is configured for the service's rate shaping setting, and the two constituent Service Flows inside the Aggregate have rate shaping disabled. The result is that the operator can configure the total aggregate rate of the service offering in each direction and does not have to configure (or even consider) how much of the user's traffic is likely to be NQB vs QB.

Figure 2 in [LLD-white-paper] illustrates an example configuration of broadband service as it might look in a current DOCSIS deployment, as well as how it would look with Low Latency DOCSIS. In the traditional configuration, there is a single downstream Service Flow with a rate of 100 Mbps and a single upstream Service Flow with a rate of 20 Mbps. In the LLD configuration, there is a single downstream Aggregate Service Flow with a rate of 100 Mbps, containing two individual Service Flows, one for Low Latency traffic and one for Classic traffic. Similarly, there is single upstream Aggregate Service Flow with a rate of 20 Mbps, containing two individual Service Flows for Low Latency and Classic traffic.

The CMTS will enforce the Aggregate "Max Sustained Traffic Rate" (AMSR), and the end-user's applications determine how much of the aggregate bandwidth they consume irrespective of which SF they use - just as they do today with a single DOCSIS SF.

As described later, Inter-Service-Flow scheduling is arranged to make the ASF function as a single pool of bandwidth.

3.2. Identifying NQB Packets - Default Classifiers

By default, the traffic within an Aggregate Service Flow is segmented into the two constituent Service Flows by a set of packet classifiers (see Figure 3 in [LLD-white-paper]) that examine the Differentiated Services (DiffServ) Field and the Explicit Congestion Notification (ECN) Field, which are standard elements of the IPv4/IPv6 header [RFC3168]. Specifically, packets with an NQB DiffServ value or an ECN field indicating either ECN Capable Transport 1 (ECT(1)) or Congestion Experienced (CE) will get mapped to the Low Latency Service Flow, and the rest of the traffic will get mapped to the Classic Service Flow.

As of the writing of this draft, it is proposed that the DiffServ value 0x2A be standardized in IETF/IANA to indicate NQB [I-D.white-tsvwg-nqb]. Certain existing DiffServ values may also be classified as NQB by default, such as Expedited Forwarding (EF).

The expectation is that non-queue-building traffic sources (applications) will either mark their packets with an NQB DiffServ value or support ECN.

Although the DiffServ Field is being used to indicate NQB behavior, that does not imply adoption of the Differentiated Services architecture as it is typically understood. In the traditional DiffServ architecture, applications indicate a desire for a particular treatment of their packets - often implemented as a priority level - which in essence conveys a value judgement as to the importance of that traffic relative to the traffic of other applications. Such an architecture can work just fine in a managed environment where all applications conform to a common view of their relative priority levels and so can be trusted to mark their packets appropriately. It fails, however, when applications need to send packets across trust boundaries between networks, where there would be no common view on their relative importance. As a result, the DiffServ architecture is often used within managed networks (corporate networks, campus networks, etc.) but is not used on the Internet.

LLD's usage of the DiffServ Field to indicate NQB sidesteps this fundamental problem by eliminating the subjective value judgement on the relative importance of applications. Instead, this usage of the DiffServ Field describes objectively verifiable behavior on the part of the application - that it will not build a queue. Therefore, networks can verify that the marking has been applied properly before a packet is allowed into the Low Latency Service Flow queue (see Section 3.4).

The ECN classifiers enable LLD's support of the IETF's Low Latency Low Loss Scalable throughput (L4S) service [I-D.ietf-tsvwg-ecn-l4s-id], which is an evolution of the original ECN facility to support applications needing both high bandwidth and low latency (see Appendix A).

3.3. Coupled AQM

To manage queuing delay, both the Low Latency Service Flow queue and the Classic Service Flow queue support Active Queue Management (AQM) (see Figure 4 in [LLD-white-paper]).

In the case of the Classic Service Flow, the queue implements the same state-of-the-art Active Queue Management techniques used in today's DOCSIS 3.1 networks. For upstream Classic Service Flows, the DOCSIS 3.1 specification mandates that the CM implement the DOCSIS-PIE (Proportional-Integral-Enhanced AQM Algorithm), which introduces packet drops at an appropriate rate to drive the queue delay to the default target value of 10 ms. For downstream Classic Service Flows, the AQM in the CMTS is still vendor specific.

In the case of the Low Latency Service Flow, the queue supports L4S congestion controllers by implementing an Immediate Active Queue Management algorithm that utilizes ECN marking instead of packet drops. By default, the algorithm does not mark the packet if the queuing delay is less than 0.475 milliseconds and always marks the packet if the delay is greater than 1 ms. Between those configurable values, the algorithm marks at a rate that ramps up from 0% to 100% over the range. In addition, per [I-D.ietf-tsvwg-aqm-dualq-coupled], the Immediate AQM in the Low Latency Queue is coupled to the Classic Queue AQM so that congestion in the Classic Queue will induce ECN marking in the Low Latency Queue that will act to balance the per-flow throughput across all of the flows in both queues. L4S congestion control and the role of the dual-queue-coupled-aqm in providing flow balance is described further in Appendix A.

To enable the Low Latency Queue to rapidly dequeue an arrived burst of traffic, the Inter-Service-Flow scheduler gives a higher weight to the Low Latency Queue than it does to the Classic Queue. The coupling to the Low Latency AQM counterbalances the weighted scheduler by making low-latency applications leave space for Classic traffic. This ensures that the weighted scheduler does not give priority over bandwidth, as a traditional weighted scheduler would.

3.4. Queue Protection

Because of the small buffer size of the Low Latency Queue, classic TCP flows or other queue-building flows would see poor performance (due to high packet loss) if they were to end up in the Low Latency Queue. In addition, they would destroy the latency performance for the non-queue-building flows, negating the primary benefits of LLD.

To prevent this situation, the packets that are classified to the Low Latency queue pass through a "Queue Protection" function (see Figure 5 in [LLD-white-paper]), which scores each flow's contribution to the growth of the queue. If the queue delay exceeds a threshold, the Queue Protection function identifies the flow or flows that have contributed most to the growth of the queue delay, and it redirects future packets from those flows to the Classic Service Flow. This

mechanism is performed objectively and statistically, without examining the identifiers or contents of the data being transmitted.

4. Upstream Scheduling Improvements

The DOCSIS upstream Media Access Control (MAC) Layer uses a request-grant mechanism. When data to be transmitted arrive at the CM, a request message is sent from the CM to the CMTS. The CMTS schedules the individual transmission bursts for all the CMs and communicates this via a bandwidth allocation map (MAP) message. Each MAP message describes the upstream transmission opportunities (grants) for a time interval and is sent shortly before the interval to which it applies.

When a CM has data to send, it waits for a "contention request" transmission opportunity. During that opportunity, it sends a short request message indicating the amount of data it has to send. It then waits for a subsequent MAP message granting it a transmission opportunity in which to send its data. This time interval between the arrival of the packet at the CM and the time at which the data arrives at the CMTS on the upstream channel is known as the Request-Grant Delay (see Figure 6 in [LLD-white-paper]). In the absence of queuing delay, this delay is generally 2-8 ms.

4.1. Faster Request Grant Loop

LLD lowers the request-grant delay by requiring support for a shorter MAP Interval and a shorter MAP Processing Time (see Figure 7 in [LLD-white-paper]).

The MAP interval is the amount of time that each MAP message describes. The MAP interval is also the time interval between consecutive MAP messages. Reducing the MAP interval means that the CMTS processes incoming requests more frequently, thus shortening the amount of time that a request might wait at the CMTS before being processed. A shorter MAP interval also means that grants are not scheduled as far into the future within each MAP message.

The MAP Processing Time is the amount of time the CMTS uses to perform its scheduling calculations. With a shorter MAP Processing Time, there is less delay between a request being received at the CMTS and the resulting grant being scheduled.

The LLD specification requires support for a nominal MAP interval of 1 ms or less for OFDMA upstream channels, in place of the 2-4 ms used previously. In certain configurations, a 1 ms MAP interval may introduce tradeoffs such as upstream and/or downstream inefficiency that will need to be weighed against the latency improvement.

4.2. Proactive Grant Service

DOCSIS scheduling services are designed to customize the behavior of the request-grant process for particular traffic types. LLD introduces a new scheduling service called Proactive Grant Service (PGS), which can eliminate the request-grant loop entirely (see Figure 8 in [LLD-white-paper]).

In PGS, a CMTS proactively schedules a stream of grants to a Service Flow at a rate that is intended to match or exceed the instantaneous demand. In doing so, the vast majority of packets carried by the Service Flow can be transmitted without being delayed by the Request-Grant process. During periods when the CMTS estimates no demand for bandwidth for a particular PGS Service Flow, it can conserve bandwidth by providing periodic unicast request opportunities rather than a stream of grants.

The service parameters that are specific to PGS are Guaranteed Grant Interval (GGI), Guaranteed Grant Rate (GGR), and Guaranteed Request Interval (GRI). In addition, the traditional rate-shaping parameters, such as Maximum Sustained Traffic Rate and Peak Rate, serve as an upper bound on the grants that can be provided to a PGS Service Flow.

PGS can eliminate the delay caused by the Request-Grant loop, but it comes at the price of efficiency. Inevitably, the CMTS will not be able to exactly predict the instantaneous demand for the Service Flow, so it may overestimate the capacity needed. When the shared channel is fully utilized, this could reduce the capacity available to other Service Flows.

The PGS scheduling type may appear at first to be similar to an existing DOCSIS upstream scheduling type "UGS/AD." The main differences with PGS are that it sets a minimum floor on the level of granting (minimum grant spacing and minimum granted bandwidth) rather than setting a fixed grant pattern (fixed grant size and precise grant spacing), it supports the "Continuous Concatenation and Fragmentation" method of filling grants (where a contiguous sequence of bytes are dequeued to fill the grant, regardless of packet boundaries) rather than only carrying a single packet in each grant, and the CM is expected to continue to send Requests to the CMTS to inform it of packets that might be waiting in the queue.

5. Low Latency DOCSIS Performance

CableLabs has developed a simulator using the NS3 platform (<https://www.nsnam.org>) in order to evaluate the performance of different aspects of LLD. The simulator models a DOCSIS 3.1 link

(OFDM/A channel types) between the CM and the CMTS and can be configured to enable or disable various components of the technology.

Because the latency performance of the service depends on the mix of applications in use by the customer, we have developed a set of 10 traffic mix scenarios that represent what we believe to be common busy-hour behaviors for a cable customer. All traffic mixes include two bidirectional UDP sessions that are modeled after online games, but they could also represent VoIP or video conferencing/chatting applications. One of the sessions has its packets marked as NQB and the other does not, allowing us to see the benefit that the low-latency queue provides.

In addition, each traffic mix has a set of other applications that create background load, as summarized in Table 2 (see Appendix B for details on the traffic types). All of this background load traffic utilizes the classic queue.

Some of these traffic mixes represent behaviors that may be very common for broadband users during busy hour, whereas others represent more extreme behaviors that users may occasionally engage in. When generating an overall view of the performance across all of the traffic mixes, we model the fact that they may not all be equally likely to occur by giving the more common mixes (1, 2, and 8) ten times the weight that we give to each of the other less common mixes.

TABLE2. BACKGROUND TRAFFIC MIXES

Traffic Mix 1	1 web user
Traffic Mix 2	1 web user, 1 video streaming user
Traffic Mix 3	1 web user, 1 FTP upstream
Traffic Mix 4	1 web user, 1 FTP downstream
Traffic Mix 5	1 web user, 1 FTP upstream and 1 FTP downstream
Traffic Mix 6	1 web user, 5 FTP upstream and 5 FTP downstream
Traffic Mix 7	1 web user, 5 FTP up, 5 FTP down, and 2 video streaming users
Traffic Mix 8	5 web users
Traffic Mix 9	16 TCP down (speedtest)
Traffic Mix 10	8 TCP up (speedtest)

Table 2

Table 3 summarizes the 99th percentile per-packet latency for the NQB-marked game traffic across all ten traffic mixes, as well as the weighted overall performance, for four different systems:

1. a legacy DOCSIS 3.1 system with AQM disabled, 2 ms MAP interval;

2. a legacy DOCSIS 3.1 system with AQM enabled, 2 ms MAP interval;

3. a Low Latency DOCSIS 3.1 system without PGS, 1 ms MAP interval; and

4. a Low Latency DOCSIS 3.1 system with PGS configured for 5 Mbps GGR, 1 ms MAP interval.

We include LLD with and without PGS because some network operators may wish to deploy LLD without the overhead that comes with PGS scheduling.

TABLE 3. 99TH PERCENTILE ROUND-TRIP LATENCY FOR NQB-MARKED TRAFFIC BETWEEN THE CM AND CMTS

	Legacy DOCSIS 3.1 with no AQM	Legacy DOCSIS 3.1 with AQM	Low Latency DOCSIS with no PGS	Low Latency DOCSIS with PGS
Traffic Mix 1	7.7 ms	7.7 ms	4.7 ms	0.9 ms
Traffic Mix 2	7.7 ms	7.7 ms	4.8 ms	0.9 ms
Traffic Mix 3	159.5 ms	36.6 ms	4.7 ms	0.9 ms
Traffic Mix 4	7.8 ms	7.9 ms	4.7 ms	0.9 ms
Traffic Mix 5	159.6 ms	57.4 ms	4.7 ms	0.9 ms
Traffic Mix 6	253.7 ms	96.7 ms	4.7 ms	0.9 ms
Traffic Mix 7	253.9 ms	74.7 ms	4.7 ms	0.9 ms
Traffic Mix 8	7.7 ms	7.7 ms	4.7 ms	0.9 ms
Traffic Mix 9	259.3 ms	52.1 ms	4.8 ms	0.9 ms
Traffic Mix 10	254.0 ms	34.1 ms	4.8 ms	0.9 ms
Weighted Overall P99	250.5 ms	32.4 ms	4.7 ms	0.9 ms

Table 3

As can be seen in this table, there are several traffic mixes (notably 1, 2, 4, and 8) for which the relatively light traffic load doesn't create the conditions for TCP to cause significant queuing delay, so even the "Legacy DOCSIS 3.1 with no AQM" system results in fairly low latency. However, in the heavier traffic mixes, the benefit of AQM can be seen and the benefit of the dual-queue mechanism in LLD becomes very apparent. By separating the NQB-marked traffic from the queue-building traffic, the NQB-marked traffic is isolated from the delay created by the TCP flows entirely, and very reliable low latency is achieved. The right-most system, which additionally implements PGS, can eliminate the request-grant delay for the NQB traffic and thereby drive the round-trip latency below 1 ms at 99th percentile.

Figure 9 in [LLD-white-paper] illustrates the weighted overall latency performance across all ten traffic mixes. The plot is a log-log complementary cumulative distribution function, with the y-axis labeled with the equivalent quantile values.

Focusing, for instance, on the horizontal through the 99th percentile (P99), it can be seen that LLD with PGS holds delay below 0.9 ms for 99% of packets. In contrast, a DOCSIS 3.1 network without AQM can only hold delay below 250 ms for 99% of packets. So, P99 delay is more than 250 times better with LLD. We therefore see that LLD will bring a consistent, low-latency, responsive quality to cable broadband performance and user experiences for NBQ traffic.

6. Deployment Considerations

6.1. Device Support

Deploying LLD in the MSO network can be accomplished via software-only upgrades to the existing DOCSIS 3.1 CMs and CMTSs. Table 4 shows which LLD features need implementation on the CM side, the CMTS side, or both. The Dual Queue feature in the upstream requires an upgrade to the CM as well as to the CMTS. The other features (Dual Queue in Downstream, Upstream Scheduling improvements) only require upgrades on the CMTS, so they can be deployed to CMs that don't support LLD (including DOCSIS 3.0 modems).

TABLE 4. DEVICE DEPENDENCIES FOR LLD FEATURES

LLD Feature	Downstream Latency Improvements - CMTS upgrade?	Downstream Latency Improvements - CM upgrade?	Upstream Latency Improvements - CMTS upgrade?	Upstream Latency Improvements - CM upgrade?
Dual Queue (ASF, Coupled AQM, QP)	Required	Not required	Required	Required
Upstream Scheduling (Faster Req-Grant Loop, PGS)	Not applicable	Not applicable	Required	Not required

Table 4

6.2. Packet Marking

The design of LLD takes the approach that applications are in the best position to determine which flows or which packets are non-queue-building. Thus, applications such as online games will be able to tag their packets with the NQB DiffServ value to indicate that they behave in a non-queue-building way, so that LLD will be able to classify them into the Low Latency Service Flow.

For these packet markings to be useful for the LLD classifiers, they will need to survive the journey from the application source to the CM or CMTS. In some cases, operators today clear the DiffServ Field in packets entering their network from an interconnecting network, which would prevent the markings making their way to the CMTS. This practice is presumably driven by the view that DiffServ Field usage is defined by each operator for use within its network, in which case preserving another network's markings has no value. As was described in Section 3.2, it is proposed that a single globally standard value be chosen to indicate NQB so that operators that intend to support LLD can ensure that this specific value traverses their inbound interconnects and their network and then arrives at the CMTS intact.

Although application marking is preferable, some network operators might want to provide immediate benefits to applications that behave in a non-queue-building way, in advance of application developers introducing support for NQB tagging. It might be possible to

repurpose the queue protection function to identify NQB behavior even if the packets are not tagged as NQB, e.g., by assuming that all non-TCP traffic is likely to be NQB and relying on queue protection to redirect the QB flows. This is currently an area of active research.

Further, it is possible that intermediary software or devices (either installed by the user or provided by the operator) could identify flows that are expected to be NQB and mark the packets on behalf of the application.

6.3. Provisioning Mechanisms

The LLD specifications include provisioning mechanisms to allow an MSO to deploy low-latency features with minimal operational impact. Figure 10 in [LLD-white-paper] shows all the pieces needed to build a low-latency service in the upstream and downstream direction. Although it is possible to define a Low Latency ASF, its constituent Classic and Low Latency SFs, and the associated classifiers explicitly in the CM's configuration file, a new feature known as the Aggregate QoS Profile can make this configuration automatic in many cases. Default classifiers will be created and default parameters for AQM and queue protection will be used, or any of these can be overridden by the operator as needed.

6.3.1. Aggregate QoS Profiles

Similar to Service Class Names that are expanded by the CMTS into a set of QoS parameters for a Service Flow during the registration process, an operator can create an Aggregate QoS Profile (AQP) on the CMTS to describe the parameters of an Aggregate Service Flow, its constituent Service Flows, and the classifiers used to identify NQB traffic.

Just like with Service Class Names, the operator can also provide explicit values in the configuration file for any ASF or SF parameters that they wish to "override".

6.3.2. Migration Using Existing Configuration File and Service Class Name

One very straightforward way to migrate to LLD configurations may not involve any changes to the CM configuration file. This method involves the automatic expansion of a Service Flow definition to a Low Latency ASF via the use of a Service Class Name and matching AQP definition.

When the CMTS sees a Service Class Name in a Service Flow definition from the CM's config file, if the CM indicates support for LLD, then

the CMTS will first use the Service Class Name as an AQP Name and look for a matching entry in the AQP Table. If it finds a matching entry, it will automatically expand the Service Flow into an ASF and two Service Flows.

This mechanism allows the operator to deploy LLD by simply updating the CMTS to support the feature and configuring AQP entries that match the Service Class Names in use in CM config files. Then, as CMs are updated over time to include support for LLD, they will automatically start being configured with a Low Latency ASF.

6.3.3. Explicit Definition of ASF in the Configuration File

An operator can also encode a Low Latency ASF in a CM configuration file directly using an Aggregate Service Flow TLV (70 or 71). The ASF TLV could have an AQP Name that is used by the CMTS to look up a definition of the ASF in its AQP Table. It could also have ASF parameters that would explicitly define the ASF or would override the AQP parameters. A configuration could also have explicit individual Service Flow TLVs (24 or 25) that are linked to the ASF via the Aggregate Service Flow Reference TLV.

6.4. Latency Histogram Reporting

As part of the AQM operation, CMs and CMTSs generate estimates of the queuing latency for the upstream and downstream Service Flows, respectively. The latency histogram reporting function exposes these estimates to the operator to provide information that can be utilized to characterize network performance, optimize configurations, or troubleshoot problems in the field.

This latency histogram reporting can be enabled via a configuration file setting or can be initiated by setting a MIB object on the device. The operator configures the bins of the histogram, and the CM or the CMTS logs the number of packets with recorded latencies into each of the bins. The CM implements histograms for upstream Service Flows, and the CMTS implements histograms for downstream Service Flows. (This function can be enabled even for Service Flows for which AQM is disabled.) The latency estimates from the AQM are represented in the form of a histogram as well as a maximum latency value. See Figure 11 in [LLD-white-paper].

7. Conclusion

LLD enables a huge leap in latency performance and will improve the Internet experience overall. With LLD, online gaming will become more responsive and video chats will cease to be "choppy." This technology will enable a range of new applications that require real-

time interface between the cyber and physical worlds, such as vehicular communications and remote health care services.

To realize the benefits of LLD, a number of parties need to take action. DOCSIS equipment manufacturers will need to develop and integrate the LLD features into software updates for CMTSs and CMs. Cable operators need to plan the roll-out of software updates and configurations to DOCSIS equipment and set up the network to support those services (e.g., carrying DiffServ/ECN markings through the network). Application and operating system vendors will need to adopt packet marking for NQB traffic and/or adopt the L4S congestion controller. Each element of the Internet ecosystem will make these decisions independently; the faster that all take the necessary steps, the more quickly the user experience will improve.

The cable industry has provisioned its network with substantial bandwidth and is poised to take another leap forward with its 10G networks. But more bandwidth is only part of the broadband performance story. Latency is becoming crucial to the evolution of broadband. That is why LLD is a cornerstone of cable's 10G future.

8. Acknowledgements

CableLabs would like to thank the participants of the Low Latency DOCSIS Working Group, representing ARRIS, Broadcom, Casa, Charter, Cisco, Comcast, Cox Communications, Huawei, Intel, Liberty Global, Nokia, Rogers, Shaw, Videotron

9. IANA Considerations

None

10. Security Considerations

TBD

11. Informative References

[DOCSIS-CCAP-OSSIv3.1]
Cable Television Laboratories, Inc., "DOCSIS 3.1 CCAP Operations Support System Interface Specification, CM-SP-CCAP-OSSIv3.1-I14-190121", January 21, 2019, <https://specification-search.cablelabs.com/CM-SP-CCAP-OSSIv3.1>.

[DOCSIS-CM-OSSIv3.1]
Cable Television Laboratories, Inc., "DOCSIS 3.1 Cable Modem Operations Support System Interface Specification, CM-SP-CM-OSSIv3.1-I14-190121", January 21, 2019, <https://specification-search.cablelabs.com/CM-SP-CM-OSSIv3.1>.

[DOCSIS-MULPIv3.1]
Cable Television Laboratories, Inc., "MAC and Upper Layer Protocols Interface Specification, CM-SP-MULPIv3.1-I17-190121", January 21, 2019, <https://specification-search.cablelabs.com/CM-SP-MULPIv3.1>.

[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-08 (work in progress), November 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-05 (work in progress), November 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.white-tsvwg-nqb]
White, G., "Identifying and Handling Non Queue Building Flows in a Bottleneck Link", draft-white-tsvwg-nqb-00 (work in progress), October 2018.

[LLD-white-paper]
White, G., Sundaresan, K., and B. Briscoe, "Low Latency DOCSIS: Technology Overview", February 2019, <https://cablela.bs/low-latency-docsis-technology-overview-february-2019>.

[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>.

[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>.

[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>.

[web-user-model] 3GPP, "3GPP2-TSGC5, HTTP, FTP and TCP models for 1xEV-DV simulations", 2001.

Appendix A. Low Latency and High Bandwidth: L4S

How can LLD support applications that want maximum speed, and low latency too? CableLabs is working with the Internet Engineering Task Force to make this a reality through a new technology called L4S: Low Latency Low Loss Scalable throughput [I-D.ietf-tsvwg-l4s-arch].

L4S improves many of today's applications (e.g., video chat, everything on the web), but it will also enable future applications that will need both high bandwidth and low delay, such as HD video conferencing, cloud-rendered interactive video, cloud-rendered virtual reality, augmented reality, remote presence with remote control, interactive light field experiences, and others yet to be invented.

L4S involves incremental changes to the congestion controller on the sender and to the AQM at the bottleneck. The key is to indicate congestion by marking packets using Explicit Congestion Notification (ECN) rather than discarding packets. L4S uses the 2-bit ECN field in the IP header (v4 or v6) and defines each marked packet to represent a lower strength of congestion signal [RFC8311] than the original ECN standard. All the benefits of L4S follow from that.

- Low Latency: The sender's L4S congestion controller makes small but frequent rate adjustments dependent on the proportion of ECN marked packets, and the L4S AQM starts applying ECN-marks to packets at a very shallow buffer threshold. This means an L4S queue can ripple at the very bottom of the buffer with sub-millisecond queuing delay but still fully utilize the link. Small, frequent adjustments could not even be considered if packet discards were used instead of ECN - they would induce a prohibitively high loss level. Further, AQMs could not consider a

very shallow threshold if small adjustments were not used, as severe link under-utilization would result.

- o Low Loss: By definition, using ECN eliminates packet discard. In turn, that eliminates retransmission delays, which particularly impact the responsiveness of short web-like exchanges of data. Using ECN eliminates both the round-trip delay repairing a loss and the delay while detecting a loss. In addition, an L4S AQM can immediately signal queue growth using ECN, catching queue growth early. In contrast, classic AQMs hold back from discarding a packet for 100-200 ms because if a burst subsides of its own accord, a loss in itself could cause more harm than the good it would do as a signal to slow down. Furthermore, eliminating packet discard eliminates the collateral damage caused to flows that were not significantly contributing to congestion.

- o Scalable Throughput: Existing congestion control algorithms don't scale, so applications need to open many simultaneous connections to fully utilize today's broadband connections. An L4S congestion controller can rapidly ramp up its sending rate to match any link capacity. This is because L4S uses a "scalable congestion controller" that maintains the same frequency of control signals (2 ECN marks per round trip on average) regardless of flow rate. With classic congestion controllers, the faster they try to go, the longer they run blind without any control signals.

The technology behind L4S isn't new; it is based on a scalable congestion control called Data Center TCP (DCTCP) that is currently used in data centers to get very high throughputs with ultra-low delay and loss. What is new is the development of a way that scalable traffic can coexist with the existing TCP and QUIC traffic on the Internet - the key that unlocks a transition to L4S. Until now, DCTCP has been confined to data centers because it would starve any classic flows sharing a link.

Separation into two queues serves two purposes: (1) it isolates L4S flows from the queuing of classic TCP and QUIC and (2) it sends each type of traffic appropriately scaled congestion signals. This results in any number of application flows (of either type) all getting roughly equal bandwidth each, as if there were just one aggregate pool of bandwidth, with no division between the Service Flows.

The approach couples the levels of ECN and drop signaling, as shown in Figure 12 in [LLD-white-paper]. The packet rate of today's classic congestion controls conforms to the well-known square-root rule (on the left of the figure). So, the classic AQM applies a drop level to Classic traffic that is coupled to the square of the ECN

marking level being applied to Low Latency traffic. The squaring in the network counterbalances the square root at the sender, so the packet rates of the two types of flow turn out roughly the same.

Supporting L4S in LLD is relatively straightforward. All that is needed is to classify L4S flows into the Low Latency SF and support the logic in the Low Latency SF to perform immediate ECN marking of packets (see Section 3.2).

Appendix B. Simulation Details

For the results reported in this paper, we set up the following network with 5 types of client devices behind the CM and a set of servers north of the CMTS. See Figure 13 in [LLD-white-paper]. The link delays shown are 1-way values. The DOCSIS link is configured in the most latency-efficient manner (short interleavers, small OFDMA frame sizes) and models a plant distance of 8 km. The service is configured with a Maximum Sustained Traffic Rate (rate limit) of 50 Mbps in the upstream direction and 200 Mbps in the downstream direction.

The upstream game traffic model involves normally distributed packet interarrival times (mu=33 ms, sigma=3 ms) and normally distributed packet sizes (mu=110 bytes, sigma=20 bytes) constrained to discard draws of packet size <32 bytes or >188 bytes. The downstream game traffic model involves normally distributed packet interarrival times (mu=33 ms, sigma=5 ms) and normally distributed packet sizes (mu=432 bytes, sigma=20 bytes) constrained to discard draws of packet size <32 bytes or >832 bytes.

The background load traffic is configured as follows. The web user is based on the 3GPP standardized web user model [web-user-model]. The video streaming model is an abstracted model of a Dynamic Adaptive Streaming over HTTP (DASH) streaming video user where the video stream is 6 Mbps and is implemented as a 3.75 MB file download every 5 seconds. Each FTP session involves the sender selecting a file size using a log-normal random variable (mu=14.8, sigma=2.0, leading to a median file size of 2.7 MB), opening a TCP connection, sending the file, closing the TCP connection, then pausing for 100 ms before repeating the process. Although we refer to this model as an FTP model, the intention is that it models TCP usage across all applications other than web browsing and video streaming.

Authors' Addresses

Greg White
CableLabs
858 Coal Creek Circle
Louisville, CO 80027
US

Email: g.white@cablelabs.com

Karthik Sundaresan
CableLabs
858 Coal Creek Circle
Louisville, CO 80027
US

Email: k.sundaresan@cablelabs.com

Bob Briscoe
CableLabs
UK

Email: b.briscoe-contractor@cablelabs.com