Configuring and Deploying Low Latency DOCSIS Networks

A Technical Paper prepared for SCTE by

Greg White
Distinguished Technologist
CableLabs
g.white@cablelabs.com

Karthik Sundaresan
Distinguished Technologist
CableLabs
k.sundaresan@cablelabs.com

CableLabs
858 Coal Creek Circle, Louisville, CO,80027
303-661-9100

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

List of Tables

Title **Page Number**

1. Introduction

Now that Low Latency DOCSIS (LLD) gear is certified and available, operators are beginning lab testing and field trials, and are working out how to deploy and configure the equipment to provide the best performance for their users. This paper provides guidance and best practices for network operators in the configuration and management of Low Latency DOCSIS services. It is essentially a cheat-sheet on the different provisioning modes available in LLD as well as a run-down of the various control knobs available to the operator - what they do, how they interact with one another, and what effect they have on the service characteristics. It introduces each of the features that make up LLD and lays out a template of recommended configuration settings, the reasons behind those choices, and how those choices might evolve over time. It provides MSOs with a step-by-step approach to turning on each of the LLD features to ultimately ensure that customers can enjoy the benefits of the technology. In addition, this paper discusses some of the end-to-end aspects of delivering on the low latency promise, such as DSCP and ECN traffic marking, and performance monitoring.

2. Brief overview of Pre-LLD latency management features

Applications today that use a transport protocol like TCP seek out as much bandwidth as possible and use a “congestion control” algorithm (most commonly “Cubic”) 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 can vary rapidly as the activity of all the devices in the household varies. The behavior of the Cubic congestion control algorithm is that it increases its sending rate (or, more precisely, the number of bytes in flight) until it experiences packet loss (generally due to the bottleneck link buffer overrunning), then it pauses sending for a short time (allowing the bottleneck link buffer to drain a bit) before ramping up its sending rate again. The result typically is that the bottleneck link buffer is kept nearly full whenever a TCP file transfer is occurring. This in turn results in latency (and latency variation) for all of the applications that are sharing that bottleneck link.

There are two features available in DOCSIS equipment that pre-date LLD, and that can help manage the latency caused by congestion controlled traffic. For reference, these two features are briefly described here.

2.1. Buffer Control

Past implementations of cable modems had upstream buffer sizes that were found to be quite large, much larger than they needed to be to ensure good TCP throughput. CMTSs similarly were built with over-sized downstream buffers. In many cases, these buffers could hold multiple seconds worth of packets. This phenomenon was referred to as “bufferbloat”, and was not unique to DOCSIS equipment. The result was that applications would see highly variable latency, with huge latency spikes occurring whenever a sufficiently large TCP file transfer occurred.

DOCSIS 3.0/3.1/4.0 specifications allow a cable operator to tune the transmit buffer size for cable modems and CMTSs in their networks. The feature controls upstream buffering in the cable modem, and downstream buffering in the CMTS for each Service Flow.

The Buffer Control parameters limit the maximum queue depth of a Service Flow. The Service Flow buffer holds the packets that are enqueued for transmission and this parameter sets an upper limit on the amount of data that can be enqueued for transmission. By providing the ability to control per-Service Flow buffers, the Buffer Control parameters provide a means of balancing throughput and latency in a standardized and configurable manner.

In order to accommodate implementation differences (e.g., varying amounts of memory available for buffering) and to allow an optimized partitioning of buffering memory based on the number of active Service Flows, the main Buffer Control parameter for a Service Flow is referred to as Target Buffer (TLV [24/25].35.2), which the operator can use to set the desired buffer size in bytes. The implementation (CM or CMTS) is required to set the actual buffer as close to the target value as is reasonably possible.

In cases where more precise control of the buffer size is desired, there are two additional parameters: Minimum Buffer (TLV [24/25].35.1) and Maximum Buffer (TLV [24/25].35.3), both also specified in bytes. The implementation is required to reject the configuration if it cannot comply with the buffer size limits configured by these two parameters.

Alternatively, an operator can configure a Default Upstream Target Buffer Configuration (TLV 68) parameter that applies to all upstream Service Flows in the absence of the Buffer Control setting for the Service Flow. This parameter is only applicable to the upstream buffers in the CM. It is specified in milliseconds, and the CM is required to size each relevant Service Flow buffer by calculating the appropriate buffer size based on the Service Flow’s Maximum Sustained Traffic Rate value.

CableLabs previously published guidelines on configuration of the Buffer Control feature [Buffer Control]. Here we provide some updated recommendations on setting buffer sizes.

In order to ensure that a single TCP connection can fully utilize the SF rate, the SF buffer size needs to be greater than or equal to the base RTT (i.e. the round-trip-time in absence of queuing delay) of the TCP connection, yet to minimize the latency and latency variation caused by TCP traffic, the buffer should be kept as small as possible. Thus setting the buffer size involves a tradeoff between ensuring good TCP throughput for long distance file transfers, and good quality of experience for latency sensitive applications.

Since many TCP connections are between a client and a local CDN node (perhaps 10-20 ms base RTT), and in many areas RTTs greater than 50 ms are fairly rare, it is recommended that operators set the buffer size to 50 ms, as a balance between good TCP performance for most file transfers and good QoE for latency sensitive applications. But, this value is region dependent. In regions where typical RTTs are significantly shorter (or longer) than 50 ms, a smaller (or larger) value can be used.

Table 1 – Recommended Buffer Control Parameters

Option 1: Per SF configuration		
Parameter	**Upstream Settings**	**Downstream Settings**
Buffer Control: Target Buffer	TLV 24.35.2 = 50 ms * MSR / 8 *MSR = Maximum Sustained Rate for the Service Flow*	TLV 25.35.2 = 50 ms * MSR / 8
Option 2: Default per CM or CMTS		
Parameter	**Upstream Settings**	**Downstream Settings**
Default Target Buffer Configuration	TLV 68 = 50 ms	CMTS vendor proprietary = 50 ms

2.1. AQM

While the Buffer Control feature allows the operator to set the buffer size to a more appropriate level, Active Queue Management (AQM) can work to reduce the buffering latency even further without sacrificing TCP throughput. AQM algorithms monitor the queue depth (or queue delay) in the buffer, and automatically make intelligent decisions to drop packets at appropriate intervals in order to send a congestion signal to the traffic senders (e.g. TCP senders) that results in the queuing delay being kept relatively low, while allowing the link to be fully utilized. Thus, a network device can support enough buffering so that it can absorb bursts of packets on the ingress but doesn't let a standing queue build up.

The DOCSIS 3.1 specification requires cable modems to implement a particular AQM algorithm called DOCSIS-PIE that is described in [RFC8034]. The DOCSIS 3.1 specification also requires CMTS equipment to implement an AQM algorithm, although it leaves the choice of that algorithm up to the implementer. Active Queue Management in DOCSIS is further discussed in [DOCSIS AQM].

By default, AQM is turned on for all Service Flows. It is a best practice to ensure that the CM Upstream AQM disable (TLV 76) in the configuration file is set to Enable, or absent (the default is enable). The Service Flows also have AQM encodings within the Service Flow set of parameters. It is also recommended to ensure that the SF AQM Disable (TLV [24/25].40.1) in the configuration file is set to Enable, or absent (the default is enable) for both upstream and downstream Service Flows. Also it is recommended to leave the Classic AQM Latency Target (TLV [24/25].40.2) at its default value of 10ms, unless experimentation is done to validate another setting.

3. Brief Overview of LLD Features

The LLD functionalities that were recently added to the DOCSIS 3.1 specifications recognize that even AQM isn't enough to provide a consistent low latency experience for latency sensitive applications like multiplayer online gaming and cloud gaming. For these applications, the bursts of packets that AQM occasionally allows to queue up in the bottleneck link buffer, are disruptive to the Quality of Experience for the user.

The key insights that led to the development of LLD were: a) while the majority of application traffic uses traditional TCP and thus causes latency variation and packet loss, other applications (in particular many latency and loss sensitive ones) aren't causing these degradations, but nonetheless suffer as a result of them; and b) there are TCP congestion control algorithms that don't cause these degradations, but they are nearly impossible to deploy in the internet because they don't work well with existing bottleneck link technology.

LLD seeks to address these two insights by allowing applications that don't cause latency variation & loss to avoid being subjected to those degradations, and by introducing support for a new class of congestion control algorithm that can adjust to the available capacity of the bottleneck link without causing those degradations in the first place.

Further details can be found in [SCTE LLD], but the core of LLD consists of support for two queues in each direction, a "Classic" queue for traditional congestion-controlled traffic, and a "Low Latency" queue for traffic that doesn't cause latency, latency variation and loss. The Classic queue has a comparatively deep buffer (along with AQM) that allows traditional congestion controllers to achieve high throughput while keeping latency reasonably under control. The Low Latency queue has a very shallow buffer along with some other features that enable "well behaved" applications to achieve ultra-low delay.

The key to LLD is that it doesn't differentiate between these two categories of traffic based on any kind of subjective judgement as to the importance of one application vs. another, or even the relative latency/loss sensitivity of one vs. another. Instead, the distinction is made based on the application sender's behavior: does it send data in a smooth and/or sparse manner that doesn't materially contribute to queuing delay and packet loss, or does it send data in such a way that it over-drives the link, causing a queue to build up, and backs off only when it senses severe congestion. A related aspect of LLD is that it doesn't prefer one type of traffic over the other. Instead the goal is to allow both types to coexist and share the link capacity in a fair manner, and to allow the application to make the decision as to how it wishes to behave, and thus which queue it should utilize.

DOCSIS equipment, going all the way back to DOCSIS 1.1 gear, supports the concept of Service Flows, which are uni-directional logical pipes set up on the DOCSIS link between the CM and CMTS. Each Service Flow is described by Quality of Service parameters that govern aspects like data rate, and packet classifiers can be set up to direct packets into the appropriate Service Flows. Low Latency DOCSIS uses this existing Service Flow (SF) functionality to create a logical pipe (SF) for the Classic queue and another for the Low Latency queue, and uses the existing classifier mechanism to direct packets into the appropriate SF. LLD then adds some new functionality that ties this pair of Service Flows together, referred to as a Low Latency Aggregate Service Flow (LL ASF), as well as some additional features that enable the LL ASF and its constituent LL SF and Classic SF to provide great performance for all of the different applications that make use of it.

One of those additional features is support for a new congestion control mechanism that allows applications to dynamically adjust their sending rate in order to fully utilize the available capacity in a fair manner, but without causing queuing delay or loss. This mechanism is referred to as the "Low Latency, Low Loss, Scalable Throughput" (L4S) architecture, and it involves the bottleneck link providing real-time congestion signals to applications via a single bit in the header of each IP packet. This "Explicit Congestion Notification" mechanism allows the sender to react to the initial onset of congestion, without triggering packet drops, and thus send at a data rate that keeps the congestion level (i.e. queuing delay) to an absolute minimum.

4. Low Latency Service Configuration

Low Latency DOCSIS service requires configuration by the operator in order for the feature to be enabled. Some of the service parameters require direct configuration by the operator, and others are specified to have useful default values, and may only require modification in certain exceptional conditions. This section describes the service parameters (configuration file TLVs) and the effect that those parameters have on the low latency service offering.

4.1. Service Rate Configuration

In traditional (single upstream/downstream SF) DOCSIS configurations, the tier of service is defined via three Service Flow rate shaping parameters (TLVs) in each direction: Maximum Sustained Traffic Rate (TLV [24/25].8), Maximum Traffic Burst (TLV [24/25].9), and Peak Traffic Rate (TLV [24/25].27). With an LLD configuration, the same three rate shaping parameters are used, however they are configured on the ASF (TLV [70/71]) rather than the Service Flows underneath the ASF. The CMTS scheduler will rate shape the aggregate of traffic in both the LL SF and the Classic SF to meet the limits set by those TLVs.

The individual Service Flow parameters for rate shaping are not expected to be configured in an LLD configuration, and equipment may or may not support configurations in which those TLVs are set.

4.2. Low Latency Classifiers

4.2.1. Packet Classifiers in LLD ASFs

Classification of traffic into the two SFs under the LL ASF is configurable by the operator using traditional DOCSIS classifiers. The DOCSIS spec does not allow definition of classifiers that direct traffic to an ASF, but rather only to a SF.

In cases where an operator would have traditionally defined a single SF in each direction, classifiers were not needed since all traffic would by default use the single (primary) SF. When that configuration is updated to LLD, Classifiers will be needed to direct a subset of traffic to the LL SF, with the Classic SF serving as the primary SF.

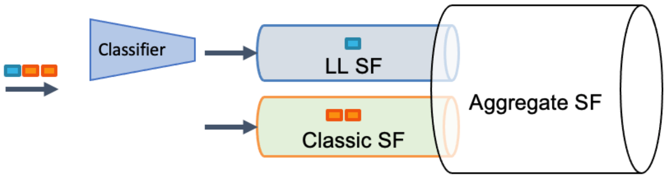


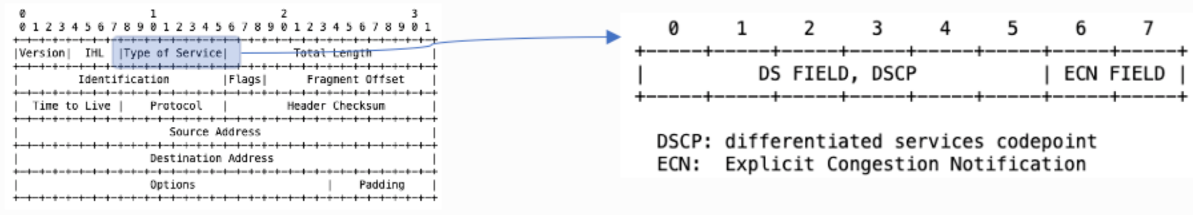
Figure 1 – Classifier Setup for US and DS Service Flows

In cases where an operator would have traditionally defined multiple SFs in each direction to carry traffic for different services (e.g. Community WiFi or voice signaling traffic), when that configuration is updated to support LLD (by replacing one or more of those SFs with ASFs), classifiers need to be defined - with appropriate Classifier Rule Priorities - such that the correct subset of traffic will be directed to the appropriate LL and Classic SFs.

The LLD-capable CMTS provides a feature referred to as “Classifier Merge” that can assist in the creation of these classifiers in certain cases. This feature is described in Section 5.5 of this paper.

4.2.2. Marking of Non-Queue-Building and L4S Traffic

There are two categories of traffic that are compatible with the LL SF: sparse “Non-Queue-Building” traffic (such as VoIP and traditional online games), and capacity-seeking (i.e. high data rate) L4S traffic (such as cloud gaming and video conferencing). These two categories use two different fields in the IP header to identify their traffic. Non-Queue-Building (NQB) traffic uses the 6-bit Diffserv field, and L4S traffic uses the 2-bit Explicit Congestion Notification (ECN) field. These two fields together form an octet that was formerly referred to as the Type of Service (ToS) octet, with the DS field being the upper 6 bits, and the ECN field being the lower 2 bits.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
  0     1     2     3     4     5     6     7
+-----+-----+-----+-----+-----+-----+-----+-----+
|          DS FIELD, DSCP           | ECN FIELD |
+-----+-----+-----+-----+-----+-----+-----+-----+

  DSCP: differentiated services codepoint
  ECN:  Explicit Congestion Notification
```

Figure 2 – The differentiated Services and ECN fields in the IP Header

4.2.2.1. NQB and DSCP

The DS field encodes a value between 0 and 63, where each specific value is known as a Diffserv Code Point (DSCP). There are several DSCPs that are used by applications today to mark sparse traffic that is generally compatible with the LL SF. It is expected that operators will wish to classify all such traffic into the upstream LL SF.

While multiple DSCPs are in use by applications in the home network, there is no requirement or expectation that an operator will carry these DSCPs into their core network. Rather it is expected that operators will select a single DSCP to be used as the NQB codepoint within their core network. Thus the configuration in the upstream is likely to include classification on multiple DSCPs and use of the DOCSIS ToS Overwrite feature to ensure that all LL traffic gets a consistent DSCP in the core network, just as the operator likely will want to configure DOCSIS ToS Overwrite on the Classic SF to ensure that all classic traffic gets a consistent default DSCP (e.g. 0) in the core network.

In terms of DSCPs that are compatible with LL, the Expedited Forwarding (EF) DSCP (46) is defined by the IETF to denote voice traffic, and is used by default by some VoIP applications, as well as for the audio stream of some video conferencing applications. Further, Windows 10 allows applications to mark "AudioVideo" or "ExcellentEffort" traffic as CS5 (40). Several multiplayer online games utilize this value for their game state update packets. Windows 10 also allows "Voice" or "Control" traffic to be marked as CS7 (56). (Note: other operating systems do not limit the choices of DSCP that applications can choose). The IETF is in the process of finalizing the assignment of an "NQB" DSCP (45) specifically for applications to denote Non-Queue-Building behavior (see [IETF NQB]). Additionally, the IETF recommends that network operators re-mark upstream NQB packets to the DSCP value 5 prior to an interconnection with another network, and that they re-mark downstream NQB packets to DSCP 45 prior to a user's home network. The rationale for this recommendation can be found in [IETF NQB].

As a result, it is recommended that upstream traffic marked by the source as CS7(56), EF(46), NQB(45), or CS5(40) be classified into the LL SF. It is also recommended that the LL SF be configured (via DOCSIS ToS Overwrite) to change the LL DSCP to the value 5, and the Classic SF be configured to change the DSCP on classic traffic to the value 0. The four recommended DSCPs for upstream classification (56,46,45,40) can be selected using two DOCSIS ToS Range and Mask classifiers, one that matches values 45 & 46 (0xB4 B8 FC), and one that matches 40 & 56 (0x28 28 2F). These classifier encodings are shown in the example in Section 4.2.3.

For downstream traffic, it is recommended that operators classify the NQB (5) DSCP into the LL SF, and use the DOCSIS ToS Overwrite feature to change it to 45. Note that the value 5 above is only a recommendation, an operator could choose a different value for use in their core network (even the value 45 if that works well). If a different value is chosen, it would be necessary to either re-mark NQB to 5 at interconnection, or negotiate with the interconnection partner on the use of the different value for NQB traffic.

4.2.2.2. L4S and ECN

L4S compliant senders mark their packets with the value ECT1 (0b01) in the ECN field. Additionally, it is possible that ECN traffic gets re-marked to CE (0b11) by a network element. As a result, it is recommended that operators classify both of these ECN values to the LL SF. These can be selected via a single DOCSIS ToS Range and Mask Classifier (0x01 01 01).

4.2.3. Recommendations for LL classifiers

Table 2 – Low Latency Classifiers

Classifier	Upstream	Downstream
IPv4 Classifier for DSCP 45 & 46	22.9.1 = 0xB4 B8 FC	23.9.1 = 0xB4 B8 FC
IPv4 Classifier for DSCP 40 & 56	22.9.1 = 0x28 28 2F	23.9.1 = 0x28 28 2F
IPv4 Classifier for ECN (ECT1 & CE)	22.9.1 = 0x01 01 01	23.9.1 = 0x01 01 01
IPv6 Classifier for DSCP 45 & 46	22.12.1 = 0xB4 B8 FC	23.12.1 = 0xB4 B8 FC
IPv6 Classifier for DSCP 40 & 56	22.12.1 = 0x28 28 2F	23.12.1 = 0x28 28 2F
IPv6 Classifier for ECN (ECT1 & CE)	22.12.1 = 0x01 01 01	23.12.1 = 0x01 01 01

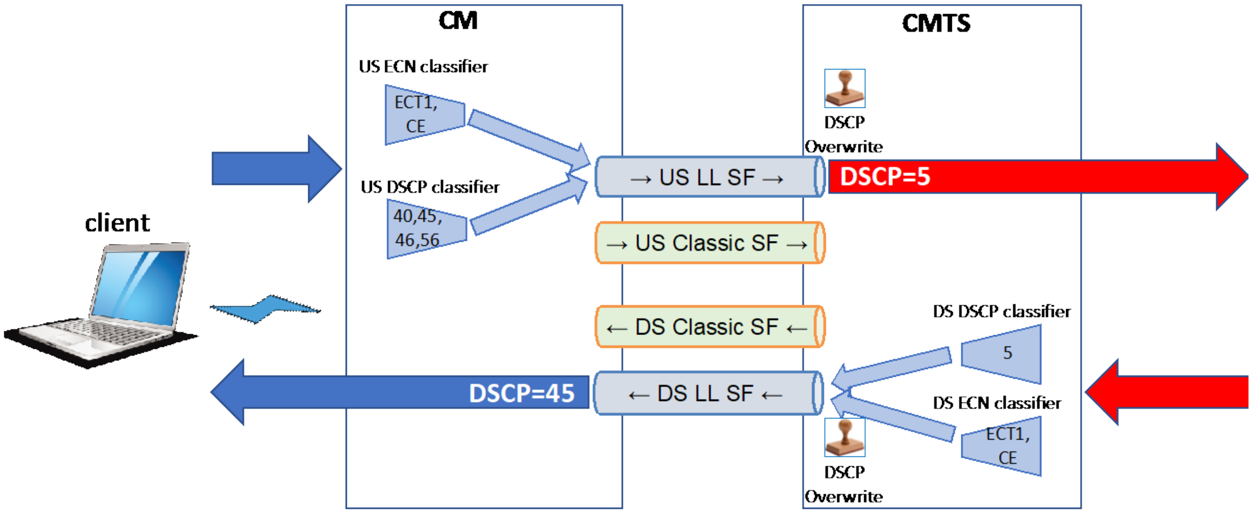


Figure 3 – Classifier Usage on CM & CMTS

4.2.4. DSCP Overwrite

Operators typically wish to manage the DSCP values that are utilized within the core network, and commonly mark all customer Internet traffic with a single DSCP (usually either CS0(0) or CS1(8)) so that it can be identified and handled appropriately in core network routers. This DSCP marking is usually set as packets ingress into the core network, both at interconnections (via router configuration), and at the access network (via the DOCSIS ToS Overwrite feature).

To support low latency end-to-end (including when one or both endpoints are outside the MSO's network) it is recommended that the operator use two DSCP values in their core network, one for default Internet traffic and the other for NQB Internet traffic. An operator can choose any pair of DSCPs that they wish, but as discussed in Section 4.2.2.1, the value 5 is recommended by IETF for use across interconnections. Here we will assume that the operator uses 0 for default Internet traffic and 5 for NQB Internet traffic in their core network.

For upstream SF configuration, it is recommended to use the ToS Overwrite feature to mark all traffic on the LL SF as NQB(5) and to mark all traffic on the Classic SF as CS0(0). Note that this will mark L4S traffic as NQB(5) as well, which is unintended. An alternative, in the case that the operator controls the home gateway software, is to selectively re-mark the four DSCPs 40,45,46,56 to 5 in the home gateway, and then just use DSCP 5 and ECN classifiers for the upstream LL SF in DOCSIS.

Current Wi-Fi gear does not yet support the same low latency features as DOCSIS (i.e. L4S, NQB isolation, Queue Protection). However, low latency traffic can be given a separate queue from classic traffic via the use of the WMM QoS Video Access Category. By default, most consumer Wi-Fi gear will map DSCPs in the range 32-47 to the Video Access Category. For upstream traffic, three of the four DSCP values that are recommended for low latency treatment (40,45,46) already get mapped into the Video Access Category by default, and the fourth DSCP (56) gets mapped into the Voice Access Category. It is recommended that operators mark downstream NQB traffic with the NQB(45) DSCP so that it is mapped to the Video Access Category by default in current Wi-Fi gear, and so that it can get full NQB treatment in future Wi-Fi gear that supports all of the features defined for the NQB PHB. Note that this will mark L4S traffic as NQB(45) as well, which is unintended. An alternative is to use selective re-marking in a router within the MSO core network (possibly even the CMTS router functionality) to change DSCP 5 to 45.

To summarize, the following DOCSIS ToS Overwrite settings are recommended.

Table 3 – DSCP TOS Overwrite Settings

Service Flow	Overwrite Setting	TLV
Upstream LL SF	set DSCP= 5 & don't modify ECN	24.23 = 0x03 14
Upstream Classic SF	set DSCP= 0 & don't modify ECN	24.23 = 0x03 00
Downstream LL SF	set DSCP= 45 & don't modify ECN	25.23 = 0x03 B4
Downstream Classic SF	set DSCP= 0 & don't modify ECN	25.23 = 0x03 00

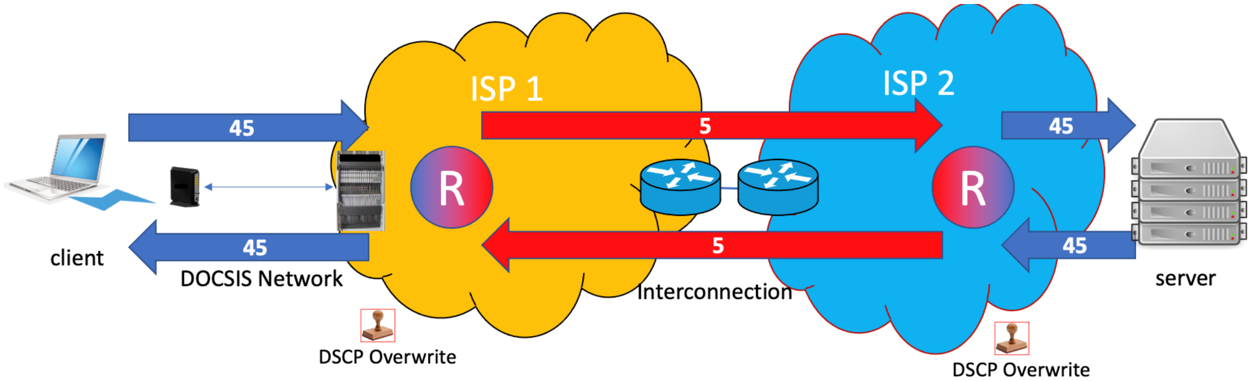


Figure 4 – NQB DSCP changes through the network

In the cases where the DOCSIS ToS Overwrite feature is used, it is noted above that this feature will overwrite the DSCP on L4S traffic as well. The implication of this is that in current Wi-Fi gear, the L4S traffic will be sent in the Video Access Category, which will give it higher priority across the Wi-Fi link than classic traffic by default. This may or may not be desirable. The IETF recommends that WiFi gear be configured such that the Video Access Category is given equal priority to the Best Effort access category, by adjusting the "EDCA" parameters for AC_VI to match those for AC_BE.

4.3. Queue Protection

LLD supports a Queue Protection (QP) function that monitors traffic classified into the LL SF in order to ensure that those traffic flows are compatible with the low latency queue. The QP function identifies flows that are misbehaving (i.e. causing queue build-up) and redirects the packets of those flows into the Classic SF.

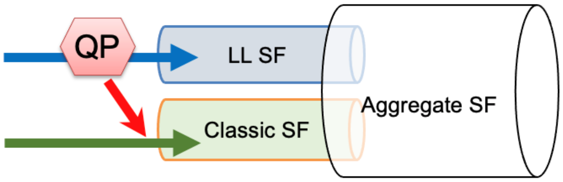


Figure 5 – Queue Protection

4.3.1. Algorithm Details

The three core concepts to understand about the queue protection function are:

1. What is a "microflow" in the eyes of the QP function?
2. What is the "congestion rate" of a microflow?
3. How does the QP decide whether to re-direct (sanction) packets?

The QP function defines a "microflow" as a stream of packets that share a common "flow ID" composed of elements of the packet header. The details of this functionality are described in Annex P.3 of [DOCSISv3.1 MULPI], but it can be summarized by the following. Implementations utilize the "5-tuple" of source IP address, destination IP address, IP Protocol (e.g. udp or tcp), and then the source port and destination port (in the case of TCP/UDP/UDP-Lite/SCTP/DCCP) or the Security Parameters Index (in the case of IPsec). In cases of un-encrypted tunnels (e.g. v4-in-v4, v4-in-v6, v6-in-v4, v6-in-v6, and GRE) implementations will parse into the inner header to find the appropriate fields. For protocols that lack a layer-4 header with meaningful flow identifiers (e.g. ICMP or IGMP), implementations will just use the 3-tuple of source-IP, dest-ip & protocol. Note that an **encrypted tunnel** (e.g. a VPN), obscures the microflows inside the tunnel, and so appears to the QP function as a single microflow.

As each packet enters the QP function, the algorithm calculates the packet's flow ID, updates the "queuing score" for that microflow, and then decides whether to put the packet in the low latency queue or the classic queue.

The queuing score for a microflow originates from a concept referred to as the "congestion rate" of the microflow. As a packet enters the QP function, the algorithm estimates how much queuing delay this packet would experience if it were to be enqueued in the low latency queue. That queue delay estimate is then used to select a "weight" for that packet between 0 and 1, using the ramp function shown in Figure 6 below.

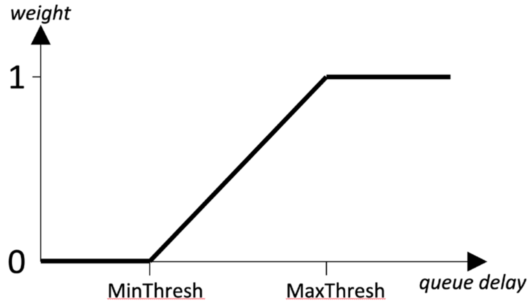


Figure 6 – Queue Protection Congestion Weight Function

This packet's size in bytes, times this weight value, is referred to as the "congestion bytes" for this packet. For example, if the packet's queue delay estimate is greater than MaxThresh, then all of the bytes of this packet are considered congestion bytes. Or, if the queue delay estimate is less than MinThresh, then this packet has zero congestion bytes. Finally, if the queue delay estimate is somewhere in between MinThresh and MaxThresh, then some fraction of the bytes of the packet are considered congestion bytes. The "congestion rate" for a microflow is then a measure of how quickly its congestion bytes are arriving. This is clearly a function of how bursty the microflow is (i.e. how much queue delay is it causing on its own) and any queue delay caused by other microflows.

The QP function mathematically 'filters' the arriving congestion bytes for a microflow through something that resembles a token bucket filter in order to generate the queuing score for that specific packet. In other words, each microflow is allowed to send at a certain congestion rate (default 4 Mbps), and any congestion bytes that arrive in excess of the allowed congestion rate 'queue up' virtually and form the basis for the queuing score. In fact the queuing score is simply the calculated time it will take for any queued-up congestion bytes (including the newly arrived congestion bytes for this packet) to drain out of this virtual queue.

The decision as to whether to re-direct the packet to the classic SF is made based on the current queuing score and the current queue delay estimate taken together. If the product of these two values (both are in units of time) exceeds the product of the two QP parameters: QPLatencyThreshold (default is equal to MaxThresh) and QPQueuingScoreThreshold (default 4 ms), and the current queue delay is greater than the QPLatencyThreshold, the packet will be re-directed. The significance of these two criteria are as follows. The 'product' criteria allows each flow to "burst" above the allowed congestion rate a little bit, but, if the queue delay grows, the amount that a flow is allowed to burst diminishes, thus causing the algorithm to more strictly enforce that allowed congestion rate. The queue delay criteria ensures that a packet arriving to a nearly empty queue won't be re-directed, even if it has a high queuing score.

This all may seem like a lot of state for the algorithm to track for the dozens or even hundreds of microflows that could exist at any point in time. But, the algorithm actually only needs to hold on to state for the microflows that currently have a queuing score. As soon as that queuing score expires (drains out), the algorithm forgets about the microflow.

Of note is the fact that the weight value used to calculate the congestion bytes for a packet is identical to the CE-marking probability used in the Immediate AQM algorithm described in the next section. This was not done simply for convenience or to reduce processing load. The result of using the same ramp function for both purposes is that the rate of CE-marked data arriving at the microflow receiver is a close approximation to the congestion rate calculated internally by the Queue Protection algorithm. As the next section describes, a responsive microflow will monitor the rate of CE-marked data arriving, and automatically adjust its sending rate in order to (in effect) keep the congestion rate low, thus ensuring that its packets remain in the low latency SF and are not re-directed to the classic SF.

4.3.2. Configuration Parameters

The behavior of the QP function is affected by six configurable parameters:

- MinThresh – the queue delay threshold below which a packet's bytes are not considered to be congestion bytes. This threshold is identical to the MinThresh used for IAQM explicit congestion notification marking. See the IAQM section of this paper for a discussion of how this value is configured and its default value.
- MaxThresh (TLV [24/25].40.3) – the queue delay threshold above which all of a packet's bytes are considered to be congestion bytes. This threshold is identical to the MaxThresh used for IAQM explicit congestion notification marking. See the IAQM section of this paper for a discussion of the default value.
- Queue Protection Enable (TLV [70/71].42.7) – a Boolean value that can be used to disable the Queue Protection function for an ASF. The default is true (enabled).
- Drain Rate Exponent (TLV [70/71].42.10) – the "drain rate" (aka "allowed congestion rate") sets the congestion rate allowance for microflows. Microflows that maintain a congestion rate that is less than this value will never have packets sanctioned. Microflows that exceed this congestion rate can experience sanctioning. This parameter is expressed as a power-of-two exponent, which allows the divide operation to be implemented as a simple bit-shift. The default value is currently 19, which corresponds to 2^{19} Bytes per second, or 4.2 Mbps. The reason for this choice of default value is that, in steady-state operation, L4S flows are expected to aim for a maximum of 2 CE marks every 15ms. Assuming 1500 byte MTU-sized packets, this equates to a maximum congestion rate of 1.6 Mbps. The exponent value 18 (2.1 Mbps) may be sufficient to allow well behaved L4S flows to avoid sanctioning, but the value 19 gives a bit more cushion. As L4S congestion control designs evolve, it may be worthwhile to experiment with the value of this parameter. Additionally, for ASF configurations where the Aggregate Maximum Sustained Traffic Rate (AMSR) value is less than 4.2 Mbps / scheduling_weight (4.66 Mbps using the default scheduling weight – scheduling weight is discussed in Section 4.6), this default value will be too high to trigger any packet sanctioning. In these scenarios, it will be necessary to configure the Drain Rate Exponent to a lower value if Queue Protection functionality is desired.
- QPLatencyThreshold (TLV [70/71].42.8) – the queue delay threshold below which QP sanctioning is suppressed, regardless of the queue score for the microflow. This value is also multiplied with the QPQueueingScoreThreshold to form the threshold for packet sanctioning described above. The default value of QPLatencyThreshold is equal to MaxThresh. Setting this to a lower value may be beneficial to better protect low latency traffic from an unresponsive microflow flooding the LL SF.
- QPQueuingScoreThreshold (TLV [70/71].42.9) – the nominal queuing score threshold for packet sanctioning. This parameter provides a congestion burst allowance for each microflow. Setting a larger value will allow microflows to cause larger bursts of queuing delay without being sanctioned, and setting a lower value will enforce the allowed congestion rate more strictly. As described above, the queue delay of a packet is multiplied by the microflow's queuing score, and this product is compared against the product of QPLatencyThreshold and QPQueuingScoreThreshold. The default value is 4ms.

4.3.3. Implications for applications

So, what does this all mean to an application or microflow?

Since the congestion rate of a microflow is always less than or equal to its total data rate, any microflow that maintains an **instantaneous** data rate (i.e. packet size divided by packet interarrival time) that is less than the Drain Rate will **never** have its packets re-directed to the classic queue. Conversely, any microflow that bursts (or continuously sends) at a rate greater than the Drain Rate, could be subject to having its packets re-directed, depending on how much queue delay it and the other flows sharing the low latency queue are creating.

Any microflow that supports L4S ECN congestion signaling can (and should be), in effect, monitoring its congestion rate and using closed-loop feedback to adjust its sending rate to avoid QP re-direction.

4.4. IAQM & Coupled AQM

The Low Latency SF supports an Active Queue Management function which performs Explicit Congestion Notification marking of packets in accordance with the L4S Architecture. Packets that pass the QP function (and thus are enqueued into the LL SF queue) and are marked by the sender as L4S ECN Capable Transport (ECT1) can potentially be marked with a Congestion Experienced (CE) mark by the LL AQM function. The LL AQM function takes no action on packets that are not marked by the sender as ECT1.

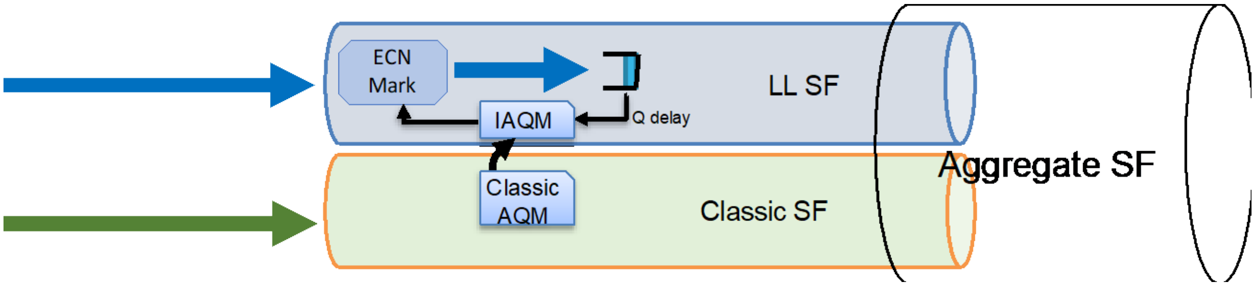


Figure 7 – IAQM in Low Latency Service Flow

4.4.1. *Algorithm Details*

The decision to CE mark a packet is driven by a coupling between two independent AQM functions, the “Immediate AQM” (IAQM) function and the Classic queue AQM function. The IAQM function uses the estimated queue delay of the packet to calculate a marking probability (probNative), the value of which is identical to the weight value used in QP.

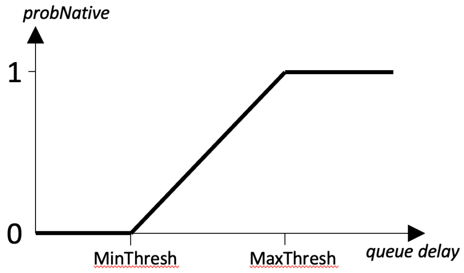


Figure 8 – IAQM Calculation of probNative

The two thresholds, MinThresh and MaxThresh, are actually configured by setting MaxThresh and the "range" of the ramp function (i.e. the difference between MaxThresh and MinThresh), where the range of the ramp function is represented by the base-2 log of the range in nanoseconds. This allows implementations to calculate probNative (and QP weight) using a simple bit-shift instead of a divide.

The Classic AQM function generates a separate probability value (drop_prob) as described in the next section. The LL AQM function calculates a CE-marking probability (probL) for the arrived packet using these two probabilities and a configurable coupling factor.

$$probL = \max\left(\text{probNative}, \min\left(1, \text{coupling\_factor} * \text{sqrt}\left(\text{drop\_prob}\right)\right)\right)$$

The purpose of this calculation, and of the LL AQM algorithm in general, is to send congestion signals to L4S capable transport protocols so that they can modulate their sending rates and bytes-in-flight to maintain both low queuing delay and a fair allocation of the link bandwidth.

When the drop_prob value is zero (i.e. there is very little traffic or queue build-up in the classic queue), probL is simply equal to probNative, and thus the LL AQM function sends congestion signals based on the instantaneous queue delay in the LL queue. By using low thresholds for the probNative ramp function, the AQM enables L4S senders to maintain low queuing delay.

When a queue forms in the Classic SF (e.g. due to classic congestion controlled traffic such as TCP cubic) the Classic AQM function will calculate a drop_prob that is appropriate for classic flows, and this value then dominates the probL equation and drives the CE-marking decisions. The result is that congestion in the Classic SF queue induces higher CE-marking in the LL SF queue, causing the L4S flows to slow down and yield capacity. The goal of this function is to balance the congestion signals for Classic flows and L4S flows such that all flows achieve approximately equal throughput.

4.4.2. Configuration Parameters

The LL AQM function is controlled by four configurable parameters:

- MaxThresh (TLV [24/25].40.4) – the queue delay above which the LL AQM would always trigger a CE-mark. The default is 1 ms.
- Range Exponent of Ramp Function (TLV [24/25].40.5) – the queue delay difference between the MinThresh value (below which the IAQM component of the LL AQM would not trigger a CE-mark) and the MaxThresh value, expressed as the base-2 log of the range in nanoseconds. The default value is 19, which equates to 524 µs.
- Coupling_factor (TLV [70/71].42.5) – the weight used to couple congestion signals from the Classic SF to the LL SF, expressed in units of tenths. Setting this to a higher value will result in classic congestion controlled flows getting greater throughput than L4S flows. Setting this to a lower value will result in L4S flows getting greater throughput than classic flows. Setting the value to 0 disables coupling. The default value is 2.0, which aims to achieve approximate fairness between classic and L4S flows in a range of conditions (see the Scheduling Weight section for further details).
- SF AQM Disable (TLV [24/25].40.1) – when set to True, disables CE-marking of packets by the LL AQM function. The default is false.

When the IAQM MinThresh value derived from the configurable parameters above equates to a value that is less than 4000 bytes divided by the AMSR, implementations will adjust the MinThresh and MaxThresh (leaving the range as configured) to ensure that MinThresh is equal to 4000 bytes divided by AMSR.

When using the default values of MaxThresh and Range Exponent, this occurs whenever the AMSR is less than ~67 Mbps.

4.5. Classic AQM

The Classic SF supports an Active Queue Management (AQM) algorithm that sends congestion signals to flows that utilize the classic SF. For cable modems, the required algorithm is DOCSIS-PIE [RFC8034], for CMTS/CCAP equipment, the choice of Classic AQM algorithm is left to the vendor. In DOCSIS-PIE, packet drops are used as congestion signals, regardless of whether the sender marks its packets as ECN capable.

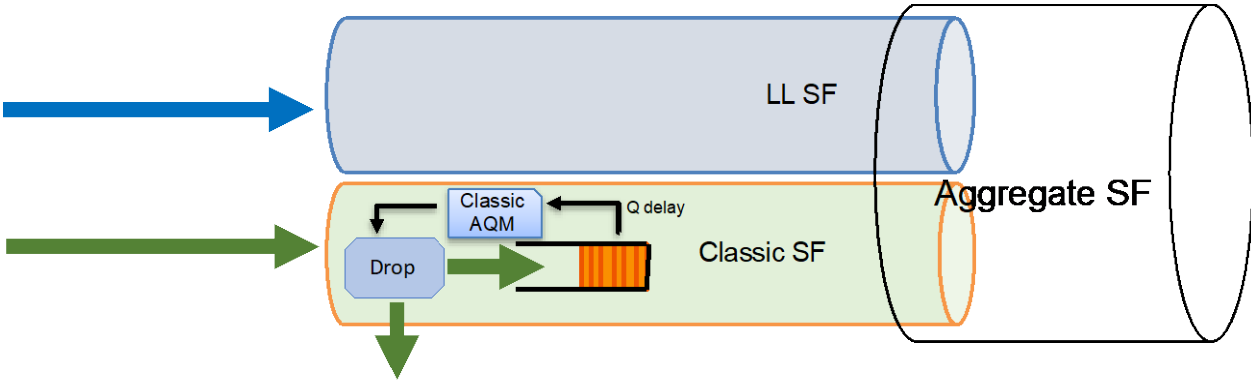


Figure 9 – DOCSIS PIE AQM in Classic Service Flow

The Classic AQM algorithm is controlled by two configurable parameters:

- Classic AQM Latency Target (TLV [24/25].40.2) – the average queue delay that the AQM algorithm is targeting. When the actual queue delay is less than the target, the AQM algorithm will reduce its drop probability, thus allowing classic congestion-controlled senders to increase their sending rate or bytes-in-flight. When the actual queue delay is greater than the target, the AQM algorithm will increase its drop probability, triggering classic senders to reduce their sending rate or bytes-in-flight, thus reducing queue delay. This value should be set to approximately one half of the expected base RTT (i.e. RTT in the absence of queue delay) that the majority of classic flows experience. The default for cable modems is 10ms. The default for CMTS/CCAP equipment is vendor-defined.
- SF AQM Disable (TLV [24/25].40.1) – when set to True, disables the Classic AQM function. The default is false.

4.6. Scheduling Weight

The two constituent SFs in the LL ASF compete for access to the bandwidth provided by the ASF rate shaper implemented in the CMTS. The CMTS scheduler is responsible for enforcing a configurable scheduling weight between the two SFs, both for downstream transmissions and for grants provided in the upstream. The scheduling weight is configured (TLV [70/71].42.6) as a value W between 1 and 255, where the LL SF is given W/256 share, and the Classic SF is given (1-W)/256 share of the bandwidth. The default value for W is 230, which equates to a scheduling weight of ~90% for the LL SF and ~10% for the Classic SF.

At first, it may seem that this parameter sets a hard partition in the capacity available to the two constituent SFs, and that the default value of (90/10) for LL vs Classic would result in LL traffic getting significantly greater throughput than Classic traffic (nine times as much!). Also, it may seem like a 50/50 ratio would be a more fair allocation. But this is not the case. The scheduling weight favors the LL SF, but is counter-balanced by the coupling mechanism described in Section 4.4 to enable per-flow fairness for all of the flows, both Classic and L4S. More detail on the coupled AQM mechanism and weighted scheduling can be found in [IETF dual-queue].

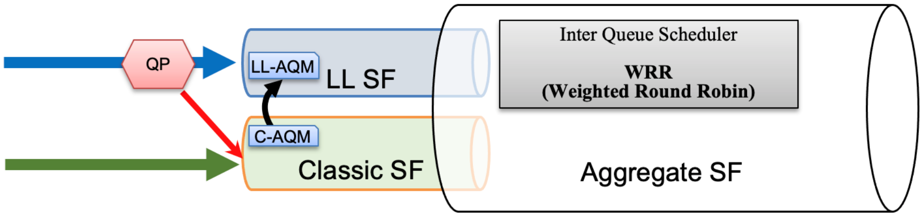


Figure 10 – WRR Scheduler

Without the coupling mechanism (e.g. if the operator disables coupling by setting Coupling Factor to 0), an operator wishing to provide "fair" allocation of bandwidth to flows utilizing the two SFs would need to predict the number of flows of each type that would likely be present, and set the weight accordingly. In essence, they would be forced to decide what fraction of the AMSR should be reserved for the LL SF vs the Classic SF.

So, when there is a mix of traffic utilizing both SFs, the default scheduling weight allows LL SF flows to consume up to 90% of the channel capacity, only as long as the Classic SF flows aren't pushing back via the coupled AQM. Thus per-flow fairness is provided as long as the ratio of L4S flows to Classic flows remains below 90/10. If the ratio of flows exceeds this value, then the L4S flows will be forced to share 90% of the capacity, and the much smaller number of Classic flows will share the remaining 10%, thus giving a throughput advantage to the Classic flows. It is expected that the ratio of L4S to Classic flows will grow over time as more operating systems and applications adopt an L4S congestion control algorithm, and thus there may be a point in time when a value for W greater than 230 would become more appropriate.

The above discussion assumes that the traffic sharing the ASF is responsive (i.e. it responds to congestion signals). However, one might wonder what would happen if unresponsive traffic were sent into one or the other of the SFs. If the unresponsive traffic is sent only to the Classic SF, it would consume whatever bandwidth that it consumes, and the remaining responsive flows (both the Classic and LL) would share the remaining bandwidth approximately equally. If the total Classic SF traffic (unresponsive and responsive) exceeds 10% of the AMSR (assuming the default scheduling weight), this would result in the Classic AQM generating a non-zero drop probability, which means that the unresponsive flow would experience a small amount of packet loss. On the other hand, if the unresponsive traffic is sent only to the LL SF, it would similarly consume whatever bandwidth that it consumes, and the remaining responsive flows (both Classic and LL) would share the remaining bandwidth approximately equally to a point. Due to the weighted scheduling, the total LL SF traffic cannot consume more than 90% of the AMSR (assuming the default scheduling weight) whenever there is capacity-seeking traffic in the Classic SF. So, if the unresponsive flow's traffic rate increases, it will eventually first squeeze the responsive LL SF traffic to zero throughput. If the unresponsive flow continues to increase its rate beyond 90% of AMSR, it will begin to have its packets sanctioned to the Classic SF, and only then begin to squeeze the responsive Classic SF traffic to zero throughput. So, in comparison, responsive traffic in the Classic SF has slightly more protection than LL SF traffic does from an unresponsive flow that occupies the LL SF. If two unresponsive flows are sent, one to the Classic SF and one to the LL SF, and together they are sending at a rate greater than AMSR, the LL SF flow would be able to consume up to 90% of the AMSR, and the Classic SF would be constrained to 10%. If the LL SF flow's rate exceeds 90% of AMSR, it would have its packets sanctioned to the Classic SF, where it would compete with the other unresponsive flow for access to the 10% of AMSR allocated to the Classic SF. Overall, this situation isn't tremendously different from the traditional single-queue situation, where an unresponsive traffic flow can squeeze the responsive traffic down to zero throughput if it wishes.

4.7. Buffer Sizing

The configuration of buffer sizes for both the LL SF and the Classic SF is done via the Buffer Control TLVs. By default, LL SFs have a buffer size that equates to 10 ms at the configured AMSR (i.e. the buffer size in bytes is equal to AMSR in bps, times 10 ms divided by 8 bits/byte), whereas Classic SFs have a default buffer size equal to at least 50ms at the configured AMSR. Note that when both queues are fully utilized, the effective tail drop limit for the LL SF would be 10ms / scheduling_weight (i.e. 11ms), and for the classic SF would be 50ms / (1 – scheduling_weight) (i.e. 500 ms).

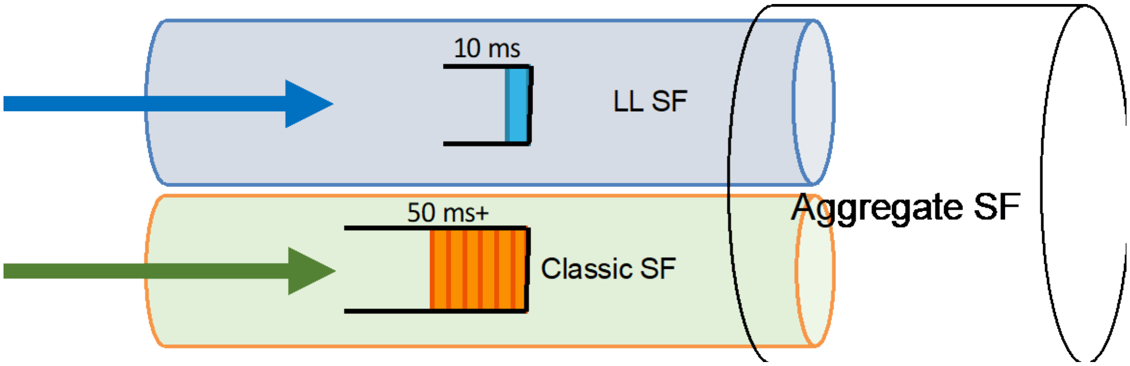


Figure 11 – Buffer sizing

For most deployment conditions, these default buffer sizes are expected to be sufficient. In many cases, the AQM and QP functions will prevent the buffers from reaching this tail drop limit. In particular, the QP function will generally keep the LL SF queue from significantly exceeding the MaxThresh value (i.e. 1 ms in the default configuration). For the Classic SF, the AQM algorithm will adjust drop probability in order to keep the steady-state queue delay at the target value (10ms).

4.8. Proactive Grant Service

The Proactive Grant Service (PGS) is designed to provide low latency for US Service Flows that may carry variable size data packets with random packet arrivals. With PGS a CMTS proactively schedules a stream of grants to the Service Flow. The intention is that the stream of grants is scheduled at a constantly adjusted rate that attempts to match or exceed the instantaneous demand. In doing so, the system ensures that the vast majority of packets carried by the Service Flow can be transmitted without being delayed by the Request-Grant process. If the traffic arrival rate exceeds the rate at which the CMTS is proactively providing grants, the CM will automatically piggyback requests for additional grants.

The bare minimum implementation of a PGS scheduler as described in the specification is the following.

There are three parameters for an upstream PGS Service Flow:

- Guaranteed Grant Interval (GGI) specifies the maximum interval (microseconds) between successive data transmission opportunities.
- Guaranteed Grant Rate (GGR) specifies the minimum granting rate, in bits/sec.
- Guaranteed Request Interval (GRI) specifies the maximum interval (microseconds) between successive request opportunities (unicast and piggyback).

If traffic activity is detected on a PGS Service Flow, the CMTS provides unsolicited data transmission opportunities at a minimum rate of GGR with an interval less than or equal to the GGI. The algorithm for detecting traffic activity on a PGS Service Flow is CMTS vendor specific.

The efficiency of a PGS service is highly dependent on the CMTS with its ability to detect inactivity and reduce the bandwidth usage to only the requests and shutting down any grants. There will always be a bit overhead in bandwidth lost, as an CMTS implementation may not be able to perfectly size the PGS grants to the varied type of traffic seen on the upstream.

Beyond this basic implementation, it is expected that CMTS schedulers will implement more sophisticated algorithms that can adjust the PGS grant rate automatically, in real-time, as demand for capacity fluctuates. Thus, the GGR value becomes a lower limit for the PGS granting, and when activity is present the CMTS grants at or above this level.

It is recommended that all Low Latency SFs be configured with PGS scheduling. For systems where the upstream capacity is constrained, it might be appropriate to set GGR to 0 (or a very low value), and allow the CMTS to dynamically adjust the actual grant rate without restriction. For systems which have sufficient upstream capacity, (and especially if the CMTS scheduler implementation is a basic one) an operator may choose to allocate some amount of bandwidth to the PGS Service Flows. GGR settings of 1-2 Mbps shows a lot of effectiveness in reducing they request-grant delay seen by the small packets that are typical of upstream voice services and online games.

Keeping in mind that the request-grant delay for an LLD-capable system might be in the range of 2-6 ms, setting the GGI value to something toward the upper end of that range may not provide much benefit. As a result, it is recommended that the GGI be set to a value less than 2 ms.

A PGS configuration of GGR = 2 Mbps and GGI = 1 ms will provide a 250 byte grant every millisecond, and thus can forward small packets (less than 250 bytes) with at most 1 ms of media access delay, and slightly larger packets (250 – 500 bytes) with at most 2 ms of media access delay. If a large packet (say 1500 bytes) arrives, the first 250 bytes will be sent in the first grant, along with a piggybacked request for 1250 bytes. It will take a full request-grant delay for the additional data grant to arrive, and in the meantime, multiple proactive grants may have been provided, which the CM uses to send further fragments of the original packet. As a result, the total media access delay for isolated packet arrivals is never greater than the request-grant delay.

The GRI value can be set by the operator to enable unicast polling (i.e. request opportunities that are dedicated to this Service Flow) that continue even during inactivity. This can be used to enable a SF to quickly restart PGS grants after a period of inactivity, even in the presence of heavy service group congestion. A value of ‘0’ for GRI disables this polling feature.

4.9. CMTS MAP interval

LLD lowers the request-grant delay by requiring support for a shorter MAP Interval and a shorter MAP processing time. The MAP interval is the amount of upstream time that each MAP message contains grants for and 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. Thus, for every millisecond that the MAP interval is reduced, the request-grant delay improves by 2 milliseconds.

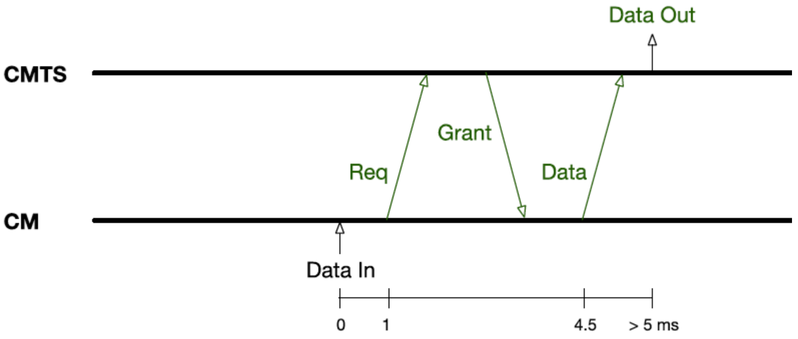


Figure 12 – CMTS Req-Grant Delay

DOCSIS MAP intervals have been in the 2 ms (typically) to 4 ms range for a long time. With LLD, CMTSs support a MAP Interval of 1 millisecond or less. Choosing a lower MAP interval for each upstream channel will reduce the latency experienced by all traffic, and is an important consideration for an operator.

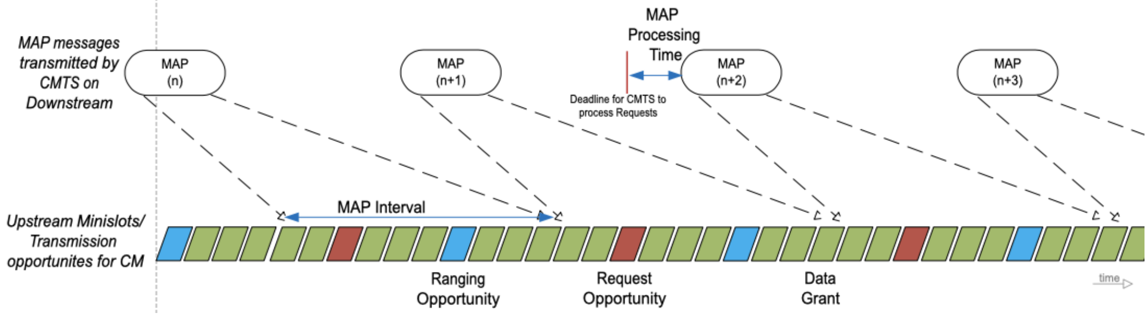

MAP messages transmitted by CMTS on Downstream
MAP (n)
MAP (n+1)
MAP Processing Time
Deadline for CMTS to process Requests
MAP (n+2)
MAP (n+3)
MAP Interval
Upstream Minislots/ Transmission opportunites for CM
Ranging Opportunity
Request Opportunity
Data Grant
time

Figure 13 – CMTS MAP Interval

Decreasing the MAP Interval does increase the amount of downstream MAP message traffic, but in many systems the latency benefit outweighs the small impact on available downstream capacity.

5. LLD Configuration Mechansims

Low latency service is configured in the upstream and/or downstream direction by enabling an Aggregate Service Flow with two underlying individual Service Flows, a Low Latency Service Flow (LL SF) and a Classic Service Flow. There are a few different ways an operator can configure Low Latency DOCSIS services on a DOCSIS 3.1 or 4.0 CM.

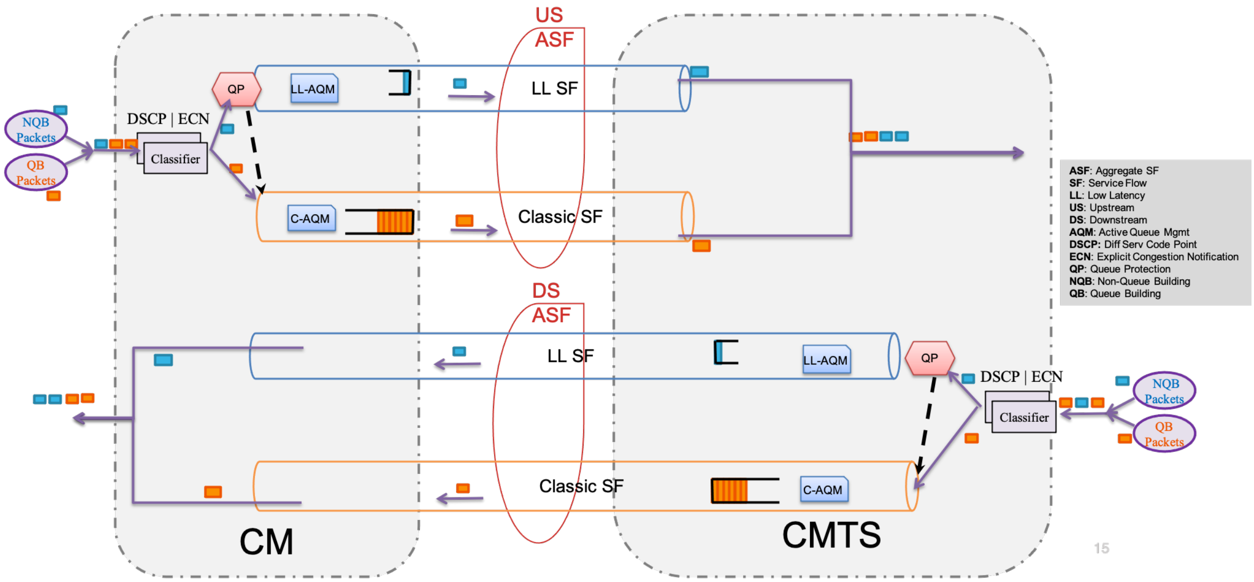


Figure 14 – Low Latency Provisioining : CM Service Flows

Enabling Low Latency services on a DOCSIS CM and CMTS involves the following components:

- Creating an ASF: An Aggregate Service Flow can be created by using explicit configuration file TLVs or by defining a template entry in the Aggregate QoS Profile (AQP) table that is then referenced in the configuration file. A separate ASF is needed for both the upstream and the downstream.
- Creating the individual SFs: Low Latency Service Flows and Classic Service Flows can be configured via explicit config file TLVs or by defining template entries in the Service Class table and then referencing those entries either in the configuration file or in an Aggregate QoS Profile (AQP) table entry. The Low Latency Service Flows and Classic Service Flows can be configured with different Quality of Service parameter sets. For example, in the upstream, the LL SF may use a scheduling type of PGS while the classic may use a scheduling type of Best Effort.

The following section describes the approaches an operator could take to configure these components.

5.1. AQP table with SF/SCN in configuration file

The primary goal with approach is to make use of existing config files and try to minimize changes to those provisioning process. This is possible in cases where the configuration file specifies a Service Class Name (SCN) for each Service Flow that the operator wishes to enable low latency services for. In this case, the current CM config file can be used as-is and the needed Low Latency DOCSIS feature configurations are made on the CMTS via the AQP table, using the Service Class Name from the config file as the AQP Name.

The provisioning system would need no changes and all CMs will boot up with the same config file as they did before the Low Latency DOCSIS features were turned on. For cable modems that support Low Latency DOCSIS, the CMTS will look up the Service Class Name in the AQP table first and expand the single Service Flow from the config file into an Aggregate Service Flow and the component Low Latency Service Flow and Classic Service Flow. For CMs that do not support Low Latency DOCSIS the CMTS will look up the Service Class Name in the Service Class Table and provision the appropriate single Service Flow.

Here is an example of the Service Flow parameters in a config file for the modem. In an actual configuration file, many other parameters will be present, this paper only shows the TLVs relevant to the LLD configuration.

Table 4 – CM Configuration File with TLV24/25 SCN

Config file parameter	TLV/Sub-TLV	Name /Value
Upstream Service Flow Encoding (TLV 24)	(Type 24.1) (Type 24.6) (Type 24.4)	Service Flow Reference = 1 QoS Parameter Set. = 07 Service Class Name = USBronze
Downstream Service Flow Encoding (TLV 25)	(Type 25.1) (Type 25.6) (Type 25.4)	Service Flow Reference= 2 QoS Parameter Set= 07 Service Class Name= DSBronze

Below are examples of an AQP & SCN definition on the CMTS, again we are focusing on the relevant LLD parameters.

Table 5 – AQP Definition on the CMTS

MIB Attributes docsQosAqpTable	US Profile	DS Profile
AQPName	USBronze	DSBronze
Direction	Upstream	Downstream
MaxAggregateTrafficRate	50 (Mbps)	100 (Mbps)
PeakTrafficRate	55 (Mbps)	110 (Mbps)
MaxTrafficBurst	50,000 (Bytes)	100,000 (Bytes)
DataRateUnitSetting	mbps	mbps
LowLatencyAsf	true	true
ClassicSfScn	USClassicSF	DSClassicSF
LatencySfScn	USLLSF	DSLLSF
AqmCouplingFactor	20 (value of 2)	20 (value of 2)
SchedulingWeight	230	230
QpEnable	0x01	0x01
QpLatencyThreshold	1000	1000
QpQueuingScoreThreshold	2000	2000
QpDrainRateExponent	19	19
LowLatencyClassifierList	0x16 0x07 0x09 0x05 0x01 0x03 0xB4 0xB8 0xFC 0x16 0x07 0x0C 0x05 0x01 0x03 0xB4 0xB8 0xFC 0x16 0x07 0x09 0x05 0x01 0x03 0x28 0x28 0x2F 0x16 0x07 0x0C 0x05 0x01 0x03 0x28 0x28 0x2F 0x16 0x07 0x09 0x05 0x01 0x03 0x01 0x01 0x01 0x16 0x07 0x0C 0x05 0x01 0x03 0x01 0x01 0x01	0x17 0x07 0x09 0x05 0x01 0x03 0xB4 0xB8 0xFC 0x17 0x07 0x0C 0x05 0x01 0x03 0xB4 0xB8 0xFC 0x17 0x07 0x09 0x05 0x01 0x03 0x28 0x28 0x2F 0x17 0x07 0x0C 0x05 0x01 0x03 0x28 0x28 0x2F 0x17 0x07 0x09 0x05 0x01 0x03 0x01 0x01 0x01 0x17 0x07 0x0C 0x05 0x01 0x03 0x01 0x01 0x01

Table 6 – SCN Definition on the CMTS

MIB Attributes docsQosScnTable	US SCN1	US SCN1	DS SCN 2	DS SCN 2
Service Class Name	USClassicSF	USLLSF	DSClassicSF	DSLLSF
TrafficPriority	0	0	0	0
MaxTrafficRate	0	0	0	0
MaxTrafficBurst	50,000	50,000	100,000	100,000
MinReservedRate	0	0	0	0
MinReservedPktSize	500	500	500	500
GuaranteedGrantRate	0	0	0	0
GuaranteedGrantInterval		1000		
GuaranteedRequestInterval		1000		
SchedulingType	bestEffort (2)	proactive GrantService (7)	bestEffort (2)	bestEffort (2)
Direction	upstream	upstream	downstream	downstream
AqmDisabled	False	False	False	False
ClassicAqmLatencyTarget	10		10	
AqmAlgorithm	docsisPIE (1)	Immediate Aqm (2)	docsisPIE (1)	Immediate Aqm (2)
ImmedAqmMaxThreshold		1000		1000
ImmedAqmRangeExponent RampFunc		19		19
DataRateUnitSetting	bps (0)	bps (0)	bps (0)	bps (0)

5.2. AQP table with ASF/AQP Name in configuration file

The goal with approach is to make use of the new AQP definitions as defined in the Low Latency DOCSIS technology. The idea here is to align the changes with the new provisioning TLVs. The idea is to use the current CM config files with the new AQP definitions which now will be used only by the CMs which support Low Latency DOCSIS. The operator will also make the needed the Low Latency DOCSIS feature configurations on the CMTS side. An operator will use the new as the profile names for the AQP definition on the CMTS. The config file will point to the same profile name as defined on the CMTS.

The provisioning system would need to now accommodate these new configuration files and CMs with LLD support will boot up with the new config file. This makes things explicit for the operator and in the network. For cable modems that support Low Latency DOCSIS the CMTS will look up the AQP table first and provision the Aggregate Service Flow and the component Low Latency Service Flow and Classic Service Flow. For CMs that do not support Low Latency DOCSIS, would receive a different configuration file and the CMTS will look up the service class name table and provision the appropriate Service Flow.

Here is an example of the ASF parameters in a new config file for the modem that reference an AQP definition on the CMTS.

Table 7 – CM Configuration File with TLV70/71 AQPName

Config file parameter	TLV/Sub-TLV	Name /Value
Upstream Aggregate Service Flow Encoding (TLV 70)	(Type 70.1) (Type 70.4)	Aggregate Service Flow Reference= 1 ASF QoS Profile (AQP) Name= USBronze
Downstream Aggregate Service Flow Encoding (TLV 71)	(Type 71.1) (Type 71.4)	Service Flow Reference= 2 ASF QoS Profile (AQP) Name = DSBronze

The same parameters in the QP and SCN table can be configured as per previous section.

5.3. Explicit TLVs in CM Configuration file

This approach requires no configuration on the CMTS, rather it includes all low latency configuration including the ASF and the constituent SFs directly in the CM config file. These config files will only be usable by CMs that support Low Latency DOCSIS, so the provisioning system will need to ensure that such a config file is only given to CMs that report support for LLD in their CM Capabilities encoding in the DHCP Discover.

Here is an example of a new config file for the LLD CM.

Table 8 – CM Configuration File with TLV24/25 SCN

Config file parameter	TLV/Sub-TLV	Name / Value
Upstream Parameters		
Upstream Aggregate Service Flow Encoding (TLV 70)	(Type 70.1) (Type 70.8) (Type 70.9) (Type 70.42.1)	Aggregate Service Flow Reference= 1 Maximum Sustained Rate= 50,000,000 Maximum Traffic Burst= 50000 Low Latency SF Reference= 3
Upstream Service Flow Encoding (LL SF) (TLV 24)	(Type 24.1) (Type 24.36) (Type 24.6) (Type 24.15) (Type 24.40.1) (Type 24.45) (Type 24.44) (Type 24.46)	Service Flow Reference= 2 Aggregate Service Flow Reference= 1 QoS Parameter Set= 07 Service Flow Scheduling Type= 07 (PGS) SF AQM Disable = 0 (enabled) Guaranteed Grant Rate = 0 Guaranteed Grant Interval = 1000 Guaranteed Request Interval = 1000
Upstream Service Flow Encoding (Classic SF) (TLV 24)	(Type 24.1) (Type 24.36) (Type 24.6) (Type 24.15) (Type 24.40.1)	Service Flow Reference= 3 Aggregate Service Flow Reference= 1 QoS Parameter Set= 07 Service Flow Scheduling Type= 02 SF AQM Disable = 0 (enabled)
Upstream Pkt Classifier Encoding (TLV 22)	(TLV 22.1) (TLV 22.3) (TLV 22.9.1)	Classifier Reference = 1 Service Flow Reference = 2 IPv4 ToS Range and Mask= tos-low=0xb4, tos-high=0xb8, tos-mask=0xfc

Config file parameter	TLV/Sub-TLV	Name / Value
Upstream Pkt Classifier Encoding (TLV 22)	(TLV 22.1) (TLV 22.3) (TLV 22.9.1)	Classifier Reference = 2 Service Flow Reference = 2 IPv4 ToS Range and Mask= tos-low=0x28, tos-high=0x28, tos-mask=0x2f
Upstream Pkt Classifier Encoding (TLV 22)	(TLV 22.1) (TLV 22.3) (TLV 22.9.1)	Classifier Reference = 3 Service Flow Reference = 2 IPv4 ToS Range and Mask= tos-low=0x01, tos-high=0x01, tos-mask=0x01
Upstream Pkt Classifier Encoding (TLV 22)	(TLV 22.1) (TLV 22.3) (TLV 22.12.1)	Classifier Reference = 4 Service Flow Reference = 2 IPv6 TC Range and Mask= tos-low=0xb4, tos-high=0xb8, tos-mask=0xfc
Upstream Pkt Classifier Encoding (TLV 22)	(TLV 22.1) (TLV 22.3) (TLV 22.12.1)	Classifier Reference = 5 Service Flow Reference = 2 IPv6 TC Range and Mask= tos-low=0x28, tos-high=0x28, tos-mask=0x2f
Upstream Pkt Classifier Encoding (TLV 22)	(TLV 22.1) (TLV 22.3) (TLV 22.12.1)	Classifier Reference = 6 Service Flow Reference = 2 IPv6 TC Range and Mask= tos-low=0x01, tos-high=0x01, tos-mask=0x01
Downstream Parameters		
Downstream Aggregate Service Flow Encoding (TLV 71)	(Type 71.1) (Type 71.8) (Type 71.9) (Type 71.42.1)	Aggregate Service Flow Reference= 11 Maximum Sustained Rate= 200000000 Maximum Traffic Burst= 200000 Low Latency SF Reference= 12
Downstream Service Flow Encoding (LL SF) (TLV 25)	(Type 25.1) (Type 25.36) (Type 25.6) (Type 25.40.1)	Service Flow Reference= 12 Aggregate Service Flow Reference= 11 QoS Parameter Set= 07 SF AQM Disable = 0 (enabled)
Downstream Service Flow Encoding (Classic SF) (TLV 25)	(Type 25.1) (Type 25.36) (Type 25.6) (Type 25.40.1)	Service Flow Reference= 13 Aggregate Service Flow Reference= 11 (Type 25.36) QoS Parameter Set= 07 SF AQM Disable = 0 (enabled)
Downstream Pkt Classifier Encoding (TLV 23)	(TLV 23.1) (TLV 23.3) (TLV 23.9.1)	Classifier Reference = 11 Service Flow Reference = 12 IPv4 ToS Range and Mask= tos-low=0xb4, tos-high=0xb8, tos-mask=0xfc

Config file parameter	TLV/Sub-TLV	Name / Value
Downstream Pkt Classifier Encoding (TLV 23)	(TLV 23.1) (TLV 23.3) (TLV 23.9.1)	Classifier Reference = 12 Service Flow Reference = 12 IPv4 ToS Range and Mask= tos-low=0x28, tos-high=0x28, tos-mask=0x2f
Downstream Pkt Classifier Encoding (TLV 23)	(TLV 23.1) (TLV 23.3) (TLV 23.9.1)	Classifier Reference = 13 Service Flow Reference = 12 IPv4 ToS Range and Mask= tos-low=0x01, tos-high=0x01, tos-mask=0x01
Downstream Pkt Classifier Encoding (TLV 23)	(TLV 23.1) (TLV 23.3) (TLV 23.12.1)	Classifier Reference = 14 Service Flow Reference = 12 IPv6 TC Range and Mask= tos-low=0xb4, tos-high=0xb8, tos-mask=0xfc
Downstream Pkt Classifier Encoding (TLV 23)	(TLV 23.1) (TLV 23.3) (TLV 23.12.1)	Classifier Reference = 15 Service Flow Reference = 12 IPv6 TC Range and Mask= tos-low=0x28, tos-high=0x28, tos-mask=0x2f
Downstream Pkt Classifier Encoding (TLV 23)	(TLV 23.1) (TLV 23.3) (TLV 23.12.1)	Classifier Reference = 16 Service Flow Reference = 12 IPv6 TC Range and Mask= tos-low=0x01, tos-high=0x01, tos-mask=0x01

5.4. Parameter Overrides

When using an Aggregate QoS Profile encoding or a Service Class Encoding, it is possible to override the values of individual parameters configured on the CMTS by specifying the override values in the configuration file. When the CMTS finds a match for the SCN or AQP Name in the AQP Table or Service Class Table and there are overriding parameters in the CM Configuration file, the CMTS utilizes the parameter values from the config file in place of the values that were specified in the AQP table or Service Class Table.

In the case that the CMTS is expanding a Service Flow encoding into an ASF and two SFs as described in Section 5.1, the individual Service Flow parameters from the config file are distributed to the ASF and SFs as discussed in Section 7.7.4.2 of [DOCSISv3.1 MULPI].

5.5. Classifier Merge Operation

When using AQP expansion, as described in Section 5.1 and 5.2 above, the AQP Table includes an attribute that allows the operator to configure packet classifiers to select traffic that is directed to the low latency SF. When the associated Classic SF is the Primary SF, these classifiers may be sufficient for directing traffic into the LL SF, and the Classic SF itself would have no classifiers. In other cases, for example when the operator wishes to create two upstream Low Latency ASFs, it is necessary to craft classifiers that can appropriately select traffic for each of the three non-Primary SFs.

The CMTS supports a functionality called Classifier Merge that makes this process automatic and straightforward.

For each configuration file classifier that points to an ASF or points to a Service Flow that is expanded into an ASF via an AQP Name match (like discussed in Section 5.1), the CMTS will apply that classifier to the Classic SF under the ASF, and will merge in the classifier fields from the AQP table to build appropriate classifiers for the Low Latency SF.

As an example, consider the following upstream portion of a configuration file definition, which includes two upstream SFs and a classifier that directs traffic for the WAN subnet 11.12.13.0/24 into the secondary SF.

Table 9 – CM Configuration File with TLV24/25 SCN

Config file parameter	TLV/Sub-TLV	Name /Value
Upstream Service Flow Encoding 1 (TLV 24)	(Type 24.1) (Type 24.6) (Type 24.4)	Service Flow Reference = 1 QoS Parameter Set. = 07 Service Class Name = USBronze
Upstream Service Flow Encoding 2 (TLV 24)	(Type 24.1) (Type 24.6) (Type 24.4)	Service Flow Reference = 21 QoS Parameter Set. = 07 Service Class Name = USBronze
Upstream Pkt Classifier Encoding (TLV 22)	(TLV 22.1) (TLV 22.3) (TLV 22.9.5) (TLV 22.9.6)	Classifier Reference = 1 Service Flow Reference = 21 IPv4 Destination Address = 11.12.13.0 IPv4 Destination Mask = 255.255.255.0

In this configuration file, both SFs are configured with a Service Class Name that matches an AQP entry on the CMTS, so each of the two SFs will be expanded into an ASF and two SFs. Let's refer to these using reference numbers 1, 2, 3 for the ASF, Classic SF & Low Latency SF expanded from Service Flow Reference 1 (the primary SF), and reference numbers 21, 22, 23 for the ASF, Classic SF & Low Latency SF expanded from Service Flow Reference 21 (the secondary SF).

Referring to Table 5 in Section 5.1, the AQP entry for USBronze has six low latency classifiers defined (three IPv4 and three IPv6). The CMTS Classifier merge would result in the following set of ten classifiers:

Table 10 – Classifier Merge Example

Classifier for	Classifier reference	Service Flow Reference	Classifier TLVs
IPv4 DSCP 45 & 46	1	3 "Primary" Low Latency SF	IPv4 ToS Range and Mask= 0xb4, 0xb8, 0xfc
IPv6 DSCP 45 & 46	2	3 "Primary" Low Latency SF	IPv6 ToS Range and Mask= 0xb4, 0xb8, 0xfc
IPv4 DSCP 40 & 56	3	3 "Primary" Low Latency SF	IPv4 ToS Range and Mask= 0x28, 0x28, 0x2f
IPv6 DSCP 40 & 56	4	3	IPv6 ToS Range and Mask= 0x28, 0x28, 0x2f

		“Primary” Low Latency SF	
IPv4 ECN	5	3 “Primary” Low Latency SF	IPv4 ToS Range and Mask= 0x01,0x01, 0x01
IPv6 ECN	6	3 “Primary” Low Latency SF	IPv6 ToS Range and Mask= 0x01,0x01, 0x01
IPv4 DSCP 45 & 46 & WAN subnet 11.12.13.0/24	7	23 “Secondary” Low Latency SF	IPv4 ToS Range and Mask= 0xb4, 0xb8, 0xfc IPv4 Destination Address = 11.12.13.0 IPv4 Destination Mask = 255.255.255.0
IPv4 DSCP 40 & 56 & WAN subnet 11.12.13.0/24	8	23 “Secondary” Low Latency SF	IPv4 ToS Range and Mask= 0x28, 0x28, 0x2f IPv4 Destination Address = 11.12.13.0 IPv4 Destination Mask = 255.255.255.0
IPv4 ECN & WAN subnet 11.12.13.0/24	9	23 “Secondary” Low Latency SF	IPv4 ToS Range and Mask= 0x01, 0x01,0x01 IPv4 Destination Address = 11.12.13.0 IPv4 Destination Mask = 255.255.255.0
IPv4 WAN subnet 11.12.13.0/24	10	22 “Secondary” Classic SF	IPv4 Destination Address = 11.12.13.0 IPv4 Destination Mask = 255.255.255.0

Note that the classifier provided in the config file was an IPv4 classifier, so the CMTS does not merge it with the three IPv6 Low Latency classifiers from the AQP table, since this “merge conflict” would create invalid classifiers.

The classifier merge process is described in Section 7.7.4.3 and Annex Q of [DOCSISv3.1 MULPI].

5.6. Primary Service Flow

Independent of the method of Low Latency DOCSIS provisioning, the CM and CMTS continue to activate the Primary Service Flows at registration time. Low Latency DOCSIS introduced new rules that the CMTS uses to determine which upstream and downstream Service Flows are the primaries based on the contents of the CM configuration file, and it introduced a new Registration Response TLV that the CMTS sends to inform the CM of this selection. The rules that the CM & CMTS use to select the primary upstream and downstream SF are as follows.

When the CM sends its registration request, it is required to send the configuration file TLVs in the order that they appear in the config file. The CMTS then selects the first SF or ASF encoding (either TLV 24/25 or 70/71) for upstream and for downstream, from the registration request. If the first selected Service Flow is an individual Service Flow (TLV 24/25), then this becomes the Primary Service Flow. If the first selected Service Flow is an ASF (TLV 70/71), then the associated Classic SF is chosen to be the primary Service Flow (even if that Classic SF TLV encoding comes much later in the registration request). If the first selected Service Flow is a TLV 24/25 that gets expanded via an AQP expansion, then the associated Classic SF after AQP expansion is chosen to be the primary Service Flow. For a CM indicating LLD support, the CMTS sends the Primary Service Flow Indicator TLV to the CM in the registration response, to identify the primary upstream and downstream Service Flow.

5.7. Device Capabilities

Configuration of Aggregate Service Flows and individual Service Flows for low latency services happens during the Registration process or can be dynamically initiated by the CMTS post-registration.

The CMTS will support at least one upstream and one downstream Low Latency ASF instance per CM. The CMTS optionally can support more than one Low Latency ASF instance in each direction per CM. A CM will support at least two Low Latency ASFs in the upstream direction. This is conveyed by the CM during the registration process, via the CM Capabilities encoding (TLV 5), in the Low Latency Support (sub-TLV 5.76). A value of 0 indicates Low Latency features are not supported and a value of 1 or more indicates the number of ASFs supported by the CM.

5.8. Compatibility Features with CMs Lacking Low Latency Support

A subset of the Low Latency features can be utilized in cases where the CM does not indicate support for Low Latency in its Modem Capabilities encoding. For example, Downstream Low Latency ASFs (and their constituent Service Flows) can be instantiated in order to provide isolation between queue-building and non-queue-building traffic in the downstream direction.

In the upstream direction, it is not possible to configure a Low Latency ASF for a CM that lacks support for Low Latency. However, it is possible to configure separate upstream Service Flows along with classifiers to direct NQB traffic to one of the two Service Flows. Keep in mind, this configuration lacks all of the functionality associated with a Low Latency ASF, i.e. aggregate rate shaping, IAQM with ECN marking, Coupled AQM, default buffer sizing, and Queue Protection, and so should be used with caution.

It is possible to configure PGS scheduling for CMs that don't support LLD. PGS scheduling is only enforced by the CMTS. For CMs that do not indicate Low Latency Support in CM capabilities, or for which Low Latency is disabled (TLV 91), the CMTS replaces the PGS scheduling type with the BE scheduling type and removes the GGI, GGR and GRI parameters in Service Flow definitions in the Registration Response that it sends to the CM. Regardless of whether the CMTS communicates the PGS configuration to the CM, the CMTS is expected to enforce the GGI, GGR and GRI parameters as configured for the Service Flow. This enables configuration of proactive scheduling on CMs that do not indicate Low Latency Support in CM capabilities. Note that the CM will report the SF as having BE scheduling type.

5.9. IPDR Ramifications

Some operators utilize Internet Protocol Detail Records (IPDR) to track per-customer data utilization stats (byte counts) for usage based billing or monthly byte cap accounting. With a Low Latency configuration, each customer will have two SFs in each direction instead of one. As a result, operators will need to consider what updates are needed in their IPDR collection and data processing implementations in order to identify both SFs and sum their byte counts.

In some cases, IPDR post processing functions key off of the Service Class Name to identify the broadband service for a customer. If the operator is using the explicit configuration file approach described in Section 5.3 above, they can include the same Service Class Name for both the LL and Classic SF (so, effectively, all of the explicit config file parameters are overrides to a basic Service Class configuration), and it may then be that no changes are needed in the IPDR post processing.

Alternatively, if the operator is using one of the AQP expansion options described in Sections 5.1 & 5.2, they could still use the same SCN for both LL & Classic, as long as they are ok with BE scheduling on the LL SF (as opposed to PGS), and default values for all of the buffering and AQM parameters.

Another approach could be to use a special character in the SCN to delimit between the "service" name and the Service Flow role, e.g. a colon as in "MyServiceClassUpstream:L" and "MyServiceClassUpstream:C", and update the IPDR post processing function to look for a match up to the first instance of the delimiter.

6. Performance Monitoring

6.1. Service Flow Statistics

Low Latency DOCSIS technology also has added support for reporting statistics on Low Latency and Classic Service Flows. An operator may want to keep an eye on the IAQM Marking Rate, the number of packets undergoing sanctioning due to queue protection, AQM drop and tail drop statistics for the Service Flows when an operator has deployed LLD technology. These statistics are summarized in the table below and it would be good operational practice for an operator to track these statistics.

Table 11 – Service Flow statistics

Device	MIB Name	MIB Entries
CM & CMTS	DocsQosSfCongestionStatsEntry	docsQosSfCongestionSanctionedPkts docsQosSfCongestionTotalEct0Pkts docsQosSfCongestionTotalEct1Pkts docsQosSfCongestionCeMarkedEct1Pkts
CM & CMTS	DocsQosServiceFlowStatsEntry	docsQosServiceFlowPkts docsQosServiceFlowOctets docsQosServiceFlowTimeCreated docsQosServiceFlowTimeActive docsQosServiceFlowPHSUnknowns docsQosServiceFlowPolicedDropPkts docsQosServiceFlowPolicedDelayPkts docsQosServiceFlowAqmDroppedPkts

An operator using these statistics could be able to understand how Queue Protection is effectively working on the modems (upstream) or CMTS (downstream), and to help debug any issues that come up. Also once L4S traffic becomes more prevalent on the internet, the TotalEct1Pkts counter on the upstream/downstream will give the operator views into how much of the traffic is L4S compliant and the CeMarkedEct1Pkts counter will give the operators confidence that packets are being marked in the DOCSIS network and the network is supporting the L4S functionality appropriately.

6.2. Latency Histograms

Downstream Service Flows and Upstream Service Flows configured for BE or PGS scheduling support Active Queue Management (AQM) algorithms. As part of their operation, these AQMs generate estimates of the queuing latency for the Service Flow. The Latency Histogram Calculation function exposes these estimates to the operator in order to provide information that can be utilized to characterize network performance, optimize configurations, or troubleshoot problems in the field.

The 'Latency Histogram Encodings' parameter, when present, enables latency histogram calculation for the given Service Flow. The latency estimates from the AQM are represented in the form of a histogram as well as a maximum latency value. 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. While the latency histogram calculation function utilizes the latency estimation algorithm from AQM, the latency histogram calculation function can be enabled even for Service Flows for which the AQM algorithm is disabled.

6.2.1. Enabling Latency Histogram via CM Config file

The histograms can be enabled in the CM config file. The table below shows an example of a portion of a configuration file with the explicit histogram TLVs included within the SF definitions.

Table 12 – CM Configuration File Portion with Histograms Enabled

Config file parameter	TLV/Sub-TLV	Name / Value
	Upstream Parameters	
Upstream Aggregate Service Flow Encoding (TLV 70)	(Type 70.1) (Type 70.8) (Type 70.9) (Type 70.42.1)	Aggregate Service Flow Reference= 1 Maximum Sustained Rate= 50000000 Maximum Traffic Burst= 50000 Low Latency SF Reference= 3
Upstream Service Flow Encoding (LL SF) (TLV 24)	(Type 24.1) (Type 24.36) (Type 24.6) (Type 24.15) (Type 24.45) (Type 24.44) (Type 24.46) (Type 24.40.1) **(Type 24.40.6)**	Service Flow Reference= 2 Aggregate Service Flow Reference= 1 QoS Parameter Set= 07 Service Flow Scheduling Type= 07 Guaranteed Grant Rate = 0 Guaranteed Grant Interval = 1000 Guaranteed Request Interval = 1000 SF AQM Disable = 0 (enabled) **Latency Histogram Encoding** (in unit of 0.01ms) = 12,20,29,37,45,53,61,69, 100,200,400,800, 1000, 1200, 1500 (Enables 16 histogram bins from 0.12ms – 15ms)
Upstream Service Flow Encoding (Classic SF) (TLV 24)	(Type 24.1) (Type 24.36) (Type 24.6) (Type 24.15) (Type 24.40.1) **(Type 24.40.6)**	Service Flow Reference= 3 Aggregate Service Flow Reference= 1 QoS Parameter Set= 07 Service Flow Scheduling Type= 02 SF AQM Disable = 0 (enabled) **Latency Histogram Encoding** (in unit of 0.01ms) = 100, 300, 500, 600, 700, 800, 900, 1200, 1500, 1800, 2500, 5000, 7500, 10000, 12500 (Enables 16 histogram bins from 1 – 125ms,)
	Downstream Parameters	
Downstream Aggregate Service Flow Encoding (TLV 71)	(Type 71.1) (Type 71.8) (Type 71.9) (Type 71.42.1)	Aggregate Service Flow Reference= 11 Maximum Sustained Rate= 200000000 Maximum Traffic Burst= 200000 Low Latency SF Reference= 12

Config file parameter	TLV/Sub-TLV	Name / Value
Downstream Service Flow Encoding (LL SF) (TLV 25)	(Type 25.1) (Type 25.36) (Type 25.6) (Type 25.40.1) **(Type 25.40.6)**	Service Flow Reference= 12 Aggregate Service Flow Reference= 11 QoS Parameter Set= 07 SF AQM Disable = 0 (enabled) **Latency Histogram Encoding** (in unit of 0.01ms) = 50, 75, 100, 125, 150, 175, 200, 300, 400, 500, 600, 800, 900,1200, 1500 (Enables 16 histogram bins from 0.5ms – 15ms)
Downstream Service Flow Encoding (Classic SF) (TLV 25)	(Type 25.1) (Type 25.36) (Type 25.6) (Type 25.40.1) **(Type 25.40.6)**	Service Flow Reference= 13 Aggregate Service Flow Reference= 11 (Type 25.36) QoS Parameter Set= 07 SF AQM Disable = 0 (enabled) **Latency Histogram Encoding** (in unit of 0.01ms) = 100, 300, 500, 600, 700, 800, 900, 1200, 1500, 1800, 2500, 5000, 7500, 10000, 12500 (Enables 16 histogram bins from 1 – 125ms)

6.2.2. *Enabling Latency Histogram via SCN*

The histograms calculation can also be enabled using the service class name definition on the CMTS. the service class name MIB object (see [DOCS-QOS3-MIB]), can be configured with the appropriate settings. This service class name is then referenced from the config file either within the AQP or the SCN definitions.

Table 13 – SCN Latency Histogram definition on CMTS

CMTS MIB Name/entry	Value
docsQosServiceClassTable -- docsQosServiceClassLatencyHistBinEdges	The attribute is formatted as a string of unsigned 16-bit integers, each representing a histogram upper bin edge in units of 10 microseconds. (This matches the Config file encoding for the same)

6.2.3. *Enabling Latency Histogram via SNMP*

Once a cable modem is initialized the latency histogram calculations can also be enabled after registration. This is done by setting the histogram bin edges in the docsQosSfLatencyHistCfgTable, for a particular Service Flow on a particular modem. This is useful for testing on the fly where an operator would like to understand the latencies on a particular CM.

Table 14 – Enable Latency Histogram Calculation - Configuration

Device	MIB Name	MIB Entries	Example 1	Example 2
CM	docsQosSfLatencyHistCfgTable	DocsQosSfLatencyHistCfgEntry ::=		
		docsQosSfLatencySfLabel	myUSLLSN	US2LLSN

Device	MIB Name	MIB Entries	Example 1	Example 2
	(for Upstream Service Flows)	docsQosSfLatencyBin1UpperEdge docsQosSfLatencyBin2UpperEdge docsQosSfLatencyBin3UpperEdge docsQosSfLatencyBin4UpperEdge docsQosSfLatencyBin5UpperEdge docsQosSfLatencyBin6UpperEdge docsQosSfLatencyBin7UpperEdge docsQosSfLatencyBin8UpperEdge docsQosSfLatencyBin9UpperEdge docsQosSfLatencyBin10UpperEdge docsQosSfLatencyBin11UpperEdge docsQosSfLatencyBin12UpperEdge docsQosSfLatencyBin13UpperEdge docsQosSfLatencyBin14UpperEdge docsQosSfLatencyBin15UpperEdge docsQosSfLatencyBinEdgeNum	12, 20, 29, 37, 45, 53, 61, 69, 100, 200, 400, 800, 1000, 1200, 1500, 15	10, 50, 100, 300, 600, 800, 1000, 1500, -- -- -- -- -- -- -- 8

Note: This MIB table includes an SfLabel entry that the operator can set to a text string of their choosing. This string is used to identify the histogram data for a particular Service Flow in the histogram data files that are uploaded by the TFTP method described in Section 6.2.5. In the case that histogram calculation is turned on via the config file, the CM/CMTS will automatically populate the SfLabel field with the Service Class Name for that Service Flow (if one exists).

6.2.4. Querying Histogram Data via SNMP

Once the histogram calculations have been enabled either via the config file or via SNMP the operator can view the histogram of latencies via SNMP in the docsQosSfLatencyStatsTable. See [DOCS-QOS3-MIB] for the detailed definitions.

Table 15 – Latency Histogram Statistics

Device	MIB Name	MIB Entries
CM & CMTS	docsQosSfLatencyStatsTable for US & DS Service Flows	DocsQosSfLatencyStatsEntry ::= docsQosSfLatencyMaxLatency docsQosSfLatencyNumHistUpdates docsQosSfLatencyBin1Pkts docsQosSfLatencyBin2Pkts docsQosSfLatencyBin3Pkts docsQosSfLatencyBin4Pkts docsQosSfLatencyBin5Pkts docsQosSfLatencyBin6Pkts docsQosSfLatencyBin7Pkts docsQosSfLatencyBin8Pkts docsQosSfLatencyBin9Pkts docsQosSfLatencyBin10Pkts docsQosSfLatencyBin11Pkts docsQosSfLatencyBin12Pkts docsQosSfLatencyBin13Pkts docsQosSfLatencyBin14Pkts docsQosSfLatencyBin15Pkts

Device	MIB Name	MIB Entries
		docsQosSfLatencyBin16Pkts

6.2.5. TFTP Reporting of Histogram Data

The CMTS and CM features and capabilities can be leveraged to enable measurement and reporting of latency estimates through each of the Service Flows. With this information, operations personnel can monitor latency trends and adjust network configurations as appropriate. Latency statistics include histogram counts, maximum latencies, etc., for each enabled Service Flow.

An operator can view the instantaneous statistics via SNMP as described in the previous section or they can have the CM & CMTS store the historical latencies observed and upload that data as a file to an external TFTP server. The following objects show how to enable this style of bulk data uploads on both the CM and the CMTS.

See [CCAP OSSI] and [CM OSSI] for the detailed definitions.

Table 16 – TFTP report of Latency Histogram - Configuration

Device	MIB Name	MIB Entries	Example Values
CM	docsCmLatency RptCfgTable for Upstream Service Flows	docsCmLatencyRptCfgSnapshotDuration docsCmLatencyRptCfgNumSnapshots docsCmLatencyRptCfgNumFiles docsCmLatencyRptCfgMeasStatus docsCmLatencyRptCfgFileName	300 288 7 ready CMUpstreamLLHist
CMTS	docsCmtsLaten cyRptCfgTable for Downstream Service Flows	docsCmtsLatencyRptCfgCmMac docsCmtsLatencyRptCfgSnapshotDuration docsCmtsLatencyRptCfgNumSnapshots docsCmtsLatencyRptCfgNumFiles docsCmtsLatencyRptCfgMeasStatus docsCmtsLatencyRptCfgFileName	CM-MAC-Addr-xxx 600 144 7 ready CMDownstreamLLHist

The operator can configure the duration of a snapshot, the measurement duration of Service Flow latency estimates. One row of statistics per Service Flow is captured during this snapshot interval and stored in the file. The number of Snapshots thus batched into a file can also be configured. E.g. If the SnapshotDuration is set to 300 seconds (5 mins), and NumSnapshots is set to 288, when enabled, the CM returns the latency data samples recorded once per 5-minute interval over the next 24-hour period. The NumFiles attribute controls the number of such files the CM uploads (0 disables, 255 enables unlimited number of files and 1-254 are other valid values). A FileName prefix can call be configured to customize the name of the files.

Statistics files can be enabled for TFTP /bulk file upload via the bulk data transfer mechanisms defined in [CCAP OSSI] and [CM OSSI].

Table 17 – TFTP /bulk file upload Configuration

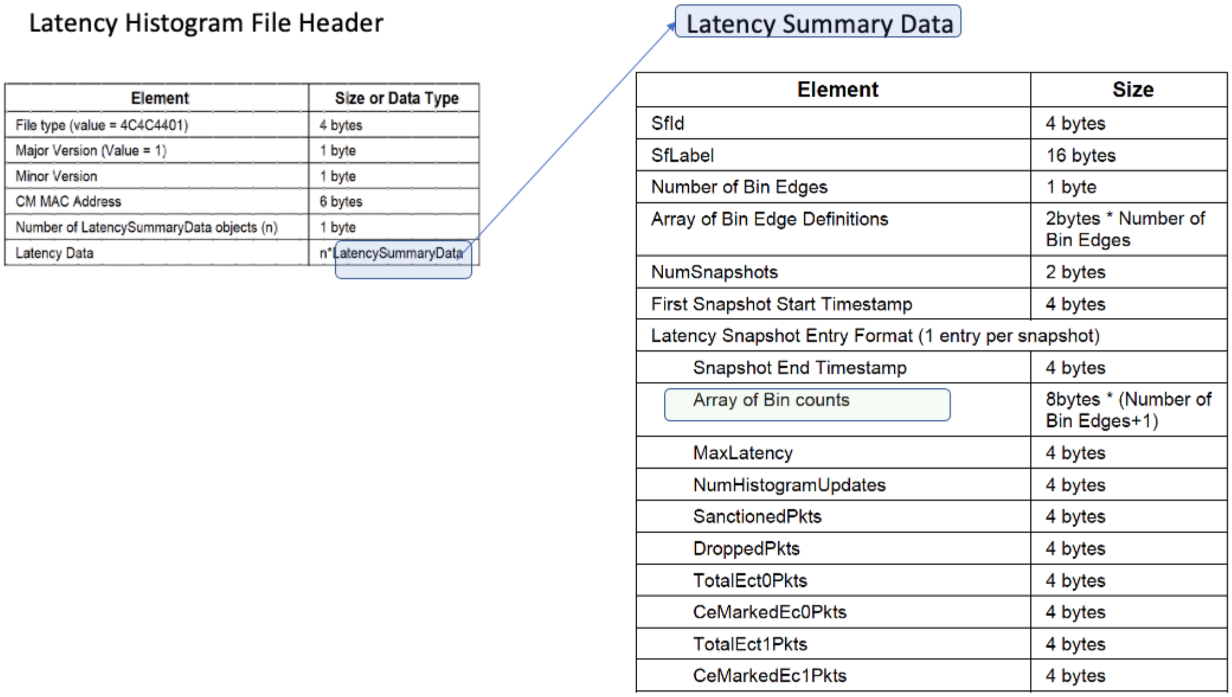
Device	MIB Name	MIB entry
CM	docsPnmBulkData docsPnmBulkDestIpAddrType docsPnmBulkDestIpAddr docsPnmBulkDestPath	

Device	MIB Name	MIB entry
	docsPnmBulkUploadControl	

CMTS	docsPnmCcapBulkDataControlTable
	DocsPnmCcapBulkDataControlEntry ::=
	docsPnmCcapBulkDataControlServerIndex
	docsPnmCcapBulkDataControlDestIpAddrType
	docsPnmCcapBulkDataControlDestIpAddr
	docsPnmCcapBulkDataControlDestPath
	docsPnmCcapBulkDataControlUploadControl
	docsPnmCcapBulkDataControlPnmTestSelector
		docsPnmCcapBulkDataControlPnmTestSelector {
		other(0),
		dsOfdmSymbolCapture (1),
		dsOfdmNoisePowerRatio(2),
		cmtsUsOfdmaActiveAndQuietProbe(3),
		usImpulseNoise(4),
		usOfdmaRxMerPerSubcarrier(5),
		upstreamHistogram(6),
		usOfdmaRxPower(7),
		usTriggeredSpectrumCapture(8),
		latencyRpt(9)
		}

6.2.6. *Using the Reported Histogram Data*

When commanded by the operator to upload histogram data the CM or the CMTS will upload a file to a TFTP server with the latency histogram metrics captured as per the snapshot configuration in the previous section.



Latency Histogram File Header

Element	Size or Data Type
File type (value = 4C4C4401)	4 bytes
Major Version (Value = 1)	1 byte
Minor Version	1 byte
CM MAC Address	6 bytes
Number of LatencySummaryData objects (n)	1 byte
Latency Data	n*LatencySummaryData

Latency Summary Data

Element	Size
SfId	4 bytes
SfLabel	16 bytes
Number of Bin Edges	1 byte
Array of Bin Edge Definitions	2bytes * Number of Bin Edges
NumSnapshots	2 bytes
First Snapshot Start Timestamp	4 bytes
Latency Snapshot Entry Format (1 entry per snapshot)	
Snapshot End Timestamp	4 bytes
Array of Bin counts	8bytes * (Number of Bin Edges+1)
MaxLatency	4 bytes
NumHistogramUpdates	4 bytes
SanctionedPkts	4 bytes
DroppedPkts	4 bytes
TotalEct0Pkts	4 bytes
CeMarkedEc0Pkts	4 bytes
TotalEct1Pkts	4 bytes
CeMarkedEc1Pkts	4 bytes

Figure 15 – Latency Histogram File format

Each latency histogram file from a CM or CMTS will have a standard file header as shown in the figure and will have a number of latency summary data entries. Each latency summary data entry will identify the Service Flow for which the latency histogram is being reported, the bin edge definitions and the number of snapshots with their timestamps, and for each snapshot will provide the count of packets in each of the bins and also some additional statistics such as the maximum latency, the number of sanctioned packets within that snapshot, etc. CableLabs has developed a tool to decode these LLD Histogram files and this is available at the CableLabs [C3 Repository].

The histogram bin counts and bin edge definitions can be used to visualize the latencies seen by the Service Flow as shown below. The two graphs below show the latency data for an upstream Aggregate Service Flow which includes a Low Latency Service Flow and a Classic Service Flow. We can see that the majority of the packets in the Low Latency Service Flow have a latency between 1~2 milliseconds. In the Classic Service Flow we can see that the majority of the packets have a latency between 9 ~ 12 milliseconds. Building histograms like this with well-crafted bin edges will help an operator understand the latency performance of each of the CMs/CMTSs. this data can then be aggregated across the network to develop a baseline of latency performance

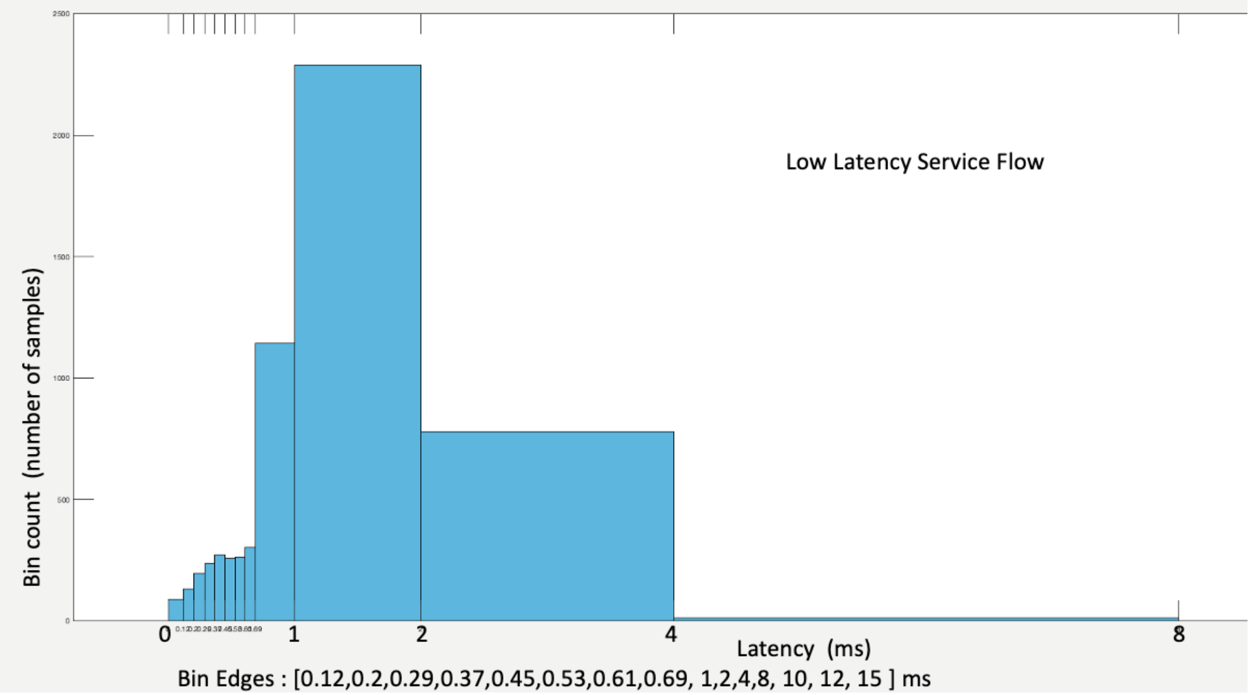


Figure 16 – Histogram plot Upstream Low Latency Service Flow

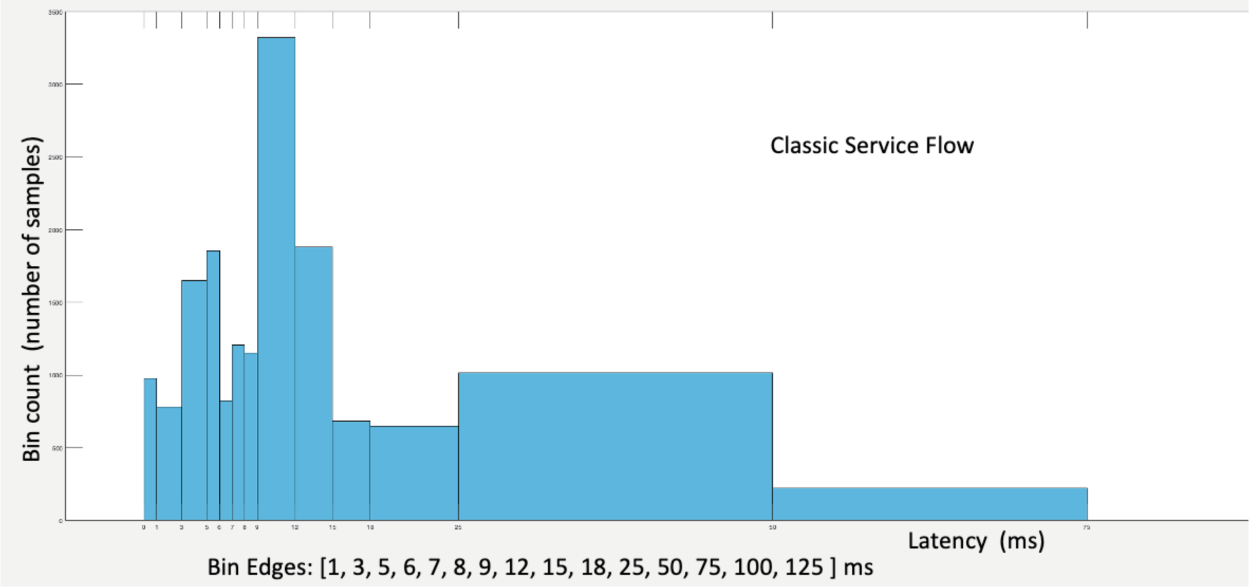


Figure 17 – Histogram plot Upstream Classic Service Flow

7. Conclusion

Low Latency DOCSIS technology (LLD) tackles the two main causes of latency in the network: queuing delay and media acquisition delay. In LLD, data traffic from applications that aren't causing latency can take a different logical path through the DOCSIS network without being stuck behind data from applications that are causing latency, as is the case in today’s Internet architectures. In addition, LLD improves the DOCSIS upstream media acquisition delay with a faster request-grant loop and a new proactive scheduling mechanism.

While the LLD parameters can be tweaked to achieve the behavior that an operator wants, the specification already chooses default values for each the parameters based on the combined judgement of the LLD working group.

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. It allows for multiple ways to provision and enable the low latency services onto the CM/ CMTS. This ranges from methods which minimize the config file changes to methods which minimize the CMTS side configuration, and an operator can choose one of the methods to initiate field trials and ultimately finalize the configuration when deploying LLD across the footprint.

In addition, it also introduces new tools to report statistics of latency performance to the operator, which can be useful to validate configuration and functionality of implementations, as well as to monitor performance over time.

Abbreviations

AC_BE	Access Category - Best Effort
AC_VI	Access Category - Video
aka	also known as
AQM	Active Queue Management
AQP	Aggregate QoS Profile
ASF	Aggregate Service Flow
B	Byte
bps	bits per second
CCAP	Converged Cable Access Platform
CDN	Content Distribution Network
CE	Congestion Experienced
CM	Cable Modem
CMTS	Cable Modem Termination System
CS	Class Selector
DCCP	Datagram Congestion Control Protocol
DHCP	Dynamic Host Configuration Protocol
DOCSIS	Data-Over-Cable Service Interface Specification
DS	Diffserv
DS	Downstream
DSCP	Diffserv Code Point
ECN	Explicit Congestion Notification
ECT	ECN Capable Transport
EDCA	Enhanced Distributed Channel Access
EF	Expedited Forwarding
GGI	Guaranteed Grant Interval
GGR	Guaranteed Grant Rate
GRE	Generic Routing Encapsulation
GRI	Guaranteed Request Interval
IAQM	Immediate Active Queue Management
ICMP	Internet Control Message Protocol
ID	Identifier
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPDR	Internet Protocol Detail Record
L4S	Low-Latency Low-Loss Scalable Throughput
LL	Low Latency
LLD	Low Latency DOCSIS
MAC	Medium Access Control
MAP	Map
max	Maximum
Mbps	Megabits per second
MIB	Management Information Base
min	Minimum
ms	millisecond
MSO	Multiple-System Operator

MSR	Maximum Sustained Traffic Rate
MTU	Maximum Transmission Unit
MULPI	MAC and Upper Layer Protocols Interface
NQB	Non-Queue-Building
OSSI	Operations Support System Interface
PGS	Proactive Grant Service
PHB	Per-Hop Behavior
PIE	Proportional Integral Enhanced
QB	Queue-Building
QoE	Quality of Experience
QoS	Quality of Service
QP	Queue Protection
RFC	Request For Comments
RTT	Round-Trip Time
SCN	Service Class Name
SCTE	Society of Cable Telecommunications Engineers
SCTP	Stream Control Transmission Protocol
SF	Service Flow
SNMP	Simple Network Management Protocol
sqrt	Square root
TCP	Transport Control Protocol
TFTP	Trivial File Transfer Protocol
TLV	type-length-value encoding
ToS	Type of Service
UDP	User Datagram Protocol
US	Upstream
VPN	Virtual Private Network
WRR	Scheduling weight
WAN	Wide Area Network

Bibliography & References

[IETF NQB] *A Non-Queue-Building Per-Hop Behavior (NQB PHB) for Differentiated Services*, Internet Engineering Task Force draft-ietf-tsvwg-nqb-07, Work In Progress, July 2021.

[DOCS-QOS3-MIB] DOCSIS Quality of Service MIB Module, http://mibs.cablelabs.com/MIBs/DOCSIS/DOCS-QOS3-MIB-2021-06-24.txt, CableLabs

[DOCS-PNM-MIB] DOCSIS PNM MIB Module, http://mibs.cablelabs.com/MIBs/DOCSIS/DOCS-PNM-MIB-2021-06-17.txt, CableLabs

[C3 Repository]. CableLabs Common Code Community, C3, https://code.cablelabs.com

[IETF L4S] *Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/draft-ietf-tsvwg-l4s-arch/, Work In Progress, July 2021.

[Buffer Control] DOCSIS® Best Practices and Guidelines: Cable Modem Buffer Control CM-GL-Buffer-V01-110915 https://www.cablelabs.com/specifications/cable-modem-buffer-control

[IETF dual-queue] *DualQ Coupled AQMs for Low Latency, Low Loss and Scalable Throughput*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/draft-ietf-tsvwg-aqm-dualq-coupled/, Work In Progress, July 2021.

[RFC8034] Active Queue Management (AQM) Based on Proportional Integral Controller Enhanced (PIE) for Data-Over-Cable Service Interface Specifications (DOCSIS) Cable Modems https://datatracker.ietf.org/doc/html/rfc8034

[DOCSISv3.1 MULPI] DOCSIS 3.1 MAC and Upper Layer Protocols Interface Specification https://www.cablelabs.com/specifications/CM-SP-MULPIv3.1

[DOCSIS AQM] Active Queue Management in DOCSIS 3.X Cable Modems, Greg White, Dan Rice, CableLabs, May 2014, https://www-res.cablelabs.com/wp-content/uploads/2019/02/28094021/DOCSIS-AQM_May2014.pdf

[SCTE LLD] Low Latency DOCSIS: Overview And Performance Characteristics, SCTE 2019 https://www.nctatechnicalpapers.com/Paper/2019/2019-low-latency-docsis

[CCAP OSSI] DOCSIS 3.1 CCAP Operations Support System Interface Specification, https://www.cablelabs.com/specifications/CM-SP-CCAP-OSSIv3.1

[CM OSSI] DOCSIS 3.1 CM Operations Support System Interface Specification, http://www.cablelabs.com/specifications/CM-SP-CM-OSSIv3.1